aws

第 2 版使用者指南

AWS Command Line Interface



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

AWS Command Line Interface: 第 2 版使用者指南

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能隸屬於 Amazon，或與 Amazon 有合作關係，或由 Amazon 贊助。

Table of Contents

..... xv
關於 AWS CLI 1
關於 AWS CLI 第 2 版 1
開發套件主要版本的維護與支援 2
Amazon Web Services 簡介 2
關於範例 2
其他文件與資源 3
AWS CLI 文件和資源 3
AWS SDKs和工具 3
開始使用 5
先決條件 6
建立 IAM 或 IAM Identity Center 管理員帳戶 6
後續步驟 7
安裝/更新 7
AWS CLI 安裝和更新指示 7
針對 AWS CLI 安裝和解除安裝錯誤進行故障診斷 25
後續步驟 25
過去版本 25
針對 AWS CLI 安裝和解除安裝錯誤進行故障診斷 39
後續步驟 39
從來源建置和安裝 39
為什麼要從來源建置？ 40
快速步驟 40
步驟 1：設定所有要求 42
步驟 2：設定 AWS CLI 來源安裝 46
步驟 3：建置 AWS CLI 52
步驟 4：安裝 AWS CLI 53
步驟 5：驗證 AWS CLI 安裝 55
工作流程範例 55
針對 AWS CLI 安裝和解除安裝錯誤進行故障診斷 58
後續步驟 58
Amazon ECR Public/Docker 58
先決條件 59
在 Amazon ECR Public 和 Docker Hub 之間進行選擇 59

執行官方映像 59
關於官方映像的介面和向後相容性的說明 60
使用特定版本和標籤 61
更新至最新的官方映像 61
分享主機檔案、憑證、環境變數和組態 62
縮短 docker run 命令 68
設定 71
收集您的憑證資訊以進行程式設計存取 71
設定新的組態和憑證 72
使用現有組態和憑證檔案 80
設定 AWS CLI 81
組態和憑證優先順序 81
本區段的其他主題 82
組態設定 82
組態和憑證檔案的格式 83
組態設定存放在何處？ 90
使用命名設定檔 91
使用命令設定和檢視組態設定 92
設定新的組態和憑證命令範例 95
支援的 config 檔案設定 98
環境變數 114
如何設定環境變數 115
AWS CLI 支援的環境變數 116
命令列選項 125
如何使用命令列選項 125
AWS CLI 支援的全域命令列選項 125
命令列選項的常見用途 129
命令完成 130
運作方式 130
在 Linux 或 macOS 上設定命令完成 131
在 Windows 上設定命令完成 134
重試 136
可用的重試模式 136
設定重試模式 138
檢視重試嘗試的記錄 139
HTTP 代理 140

使用範例 140
驗證代理 141
在 Amazon EC2 執行個體上使用代理 142
故障診斷 143
端點 143
設定單一命令的端點 143
設定所有 的全域端點 AWS 服務 143
設定為對所有 AWS 服務使用 FIP 端點 145
設定為所有 AWS 服務使用雙堆疊端點 146
設定服務特定端點 147
帳戶型端點 150
端點組態和設定優先順序 152
驗證與存取憑證 153
組態和憑證優先順序 154
本區段的其他主題 155
IAM Identity Center 驗證 155
先決條件 156
使用 aws configure sso 精靈設定您的設定檔 157
使用 aws configure sso-session 精靈只設定 sso-session 區段 160
使用 config 檔案手動設定 161
登入 IAM Identity Center 工作階段 163
使用 IAM Identity Center 設定檔執行命令 165
登出 IAM Identity Center 工作階段 165
故障診斷 165
相關資源 166
IAM Identity Center 概念 166
教學課程： AWS IAM Identity Center 和 Amazon S3 170
短期憑證 175
IAM 角色 176
先決條件 177
使用 IAM 角色概觀 177
設定和使用角色 178
使用 MFA 180
跨帳戶角色和外部 ID 181
指定角色工作階段名稱以更輕鬆地稽核 182
擔任具有 Web 身分的角色 182

清除快取的憑證 184
IAM 使用者 184
步驟 1：建立 IAM 使用者 185
步驟 2：取得您的存取金鑰 185
設定 AWS CLI 185
Amazon EC2 中繼資料 187
先決條件 188
設定用於 Amazon EC2 中繼資料的設定檔 188
外部憑證 189
使用 AWS CLI 192
取得協助 192
內建 AWS CLI 說明命令 193
AWS CLI 參考指南 197
API 文件 198
故障診斷錯誤 198
其他協助 198
命令結構 198
命令結構 199
等候命令 200
指定參數值 201
常用參數類型 202
具有字串的引號 206
檔案參數 210
產生 CLI 骨架範本 213
速記語法 227
自動提示 229
運作方式 230
自動提示功能 230
自動提示模式 233
設定自動提示 233
控制命令輸出 234
敏感輸出 234
伺服器端與用戶端輸出選項 235
輸出格式 235
分頁 244
篩選 輸出 249

傳回碼 272
精靈 273
運作方式 274
Aliases 275
先決條件 275
步驟 1：建立別名檔案 275
步驟 2：建立別名 276
步驟 3：呼叫別名 279
別名儲存庫範例 281
資源 282
代碼範例 283
引導式命令範例 283
DynamoDB 284
Amazon EC2 287
S3 Glacier 305
IAM 311
Amazon S3 315
Amazon SNS 333
命令範例 336
ACM 342
API Gateway 353
API Gateway HTTP 和 WebSocket API 417
API Gateway Management API 468
App Mesh 470
App Runner 514
AWS AppConfig 548
Application Auto Scaling 581
Application Discovery Service 598
AppRegistry 604
Athena 615
Auto Scaling 648
自動擴展計畫 715
AWS Backup 722
AWS Batch 728
AWS Budgets 743
Amazon Chime 753

雲端控制 API 796
AWS Cloud Map 802
AWS Cloud9 824
AWS CloudFormation 832
CloudFront 880
Amazon CloudSearch 984
CloudTrail 985
CloudWatch 1001
CloudWatch Logs 1035
CloudWatch 網路監控 1041
CloudWatch 可觀測性存取監視器 1054
CloudWatch 可觀測性管理員 1065
CloudWatch Synthetics 1071
CodeArtifact 1088
CodeBuild 1115
CodeCommit： 1178
CodeDeploy 1248
CodeGuru Reviewer 1288
CodePipeline 1306
AWS CodeStar 通知 1337
CodeConnections 1348
Amazon Cognito 身分 1355
Amazon Cognito 身分提供者 1361
Amazon Comprehend 1505
Amazon Comprehend Medical 1638
AWS Config 1672
Amazon Connect 1695
AWS Cost and Usage Report 1711
Cost Explorer 服務 1714
Firehose 1722
Amazon Data Lifecycle Manager 1725
AWS Data Pipeline 1731
DataSync 1740
DAX 1744
偵測 1762
Device Farm 1773

AWS Direct Connect 1778
AWS Directory Service 1828
AWS Directory Service 資料 1830
AWS DMS 1855
Amazon DocumentDB 1898
DynamoDB 1954
DynamoDB Streams 2048
Amazon EC2 2055
Amazon EC2 Instance Connect 2707
Amazon ECR 2708
Amazon ECR Public 2738
Amazon ECS 2766
Amazon EFS 2889
Amazon EKS 2897
Elastic Beanstalk 2992
Elastic Load Balancing - 第 1 版 3021
Elastic Load Balancing - 第 2 版 3048
Elastic Transcoder 3101
ElastiCache 3128
MediaStore 3232
Amazon EMR 3248
Amazon EMR on EKS 3296
EventBridge 3297
EventBridge Pipes 3303
Firewall Manager 3310
AWS FIS 3320
Amazon GameLift 伺服器 3338
Global Accelerator 3371
AWS Glue 3408
GuardDuty 3430
AWS Health 3448
HealthImaging 3455
HealthLake 3481
HealthOmics 3494
IAM 3560
IAM Access Analyzer 3695

映像建置器 3730
Incident Manager 3771
Incident Manager Contacts 3792
Amazon Inspector 3814
AWS IoT 3861
AWS IoT Analytics 4036
Device Advisor 4063
AWS IoT data 4077
AWS IoT Events 4079
AWS IoT Events-Data 4104
AWS IoT Greengrass 4129
AWS IoT Greengrass V2 4211
AWS IoT Jobs SDK release 4236
AWS IoT SiteWise 4239
AWS IoT Things Graph 4287
AWS IoT Wireless 4313
Amazon IVS 4348
Amazon IVS 聊天功能 4391
Amazon IVS 即時串流 4404
Amazon Kendra 4453
Kinesis 4462
AWS KMS 4481
Lake Formation 4548
Lambda 4598
License Manager 4638
Lightsail 4651
Macie 4774
Amazon Managed Grafana 4779
MediaConnect 4781
MediaConvert 4796
MediaLive 4820
MediaPackage 4826
MediaPackage VOD 4840
MediaStore 資料平面 4852
MediaTailor 4857
MemoryDB 4862

Amazon MSK 4899
網路流量監控 4907
Network Manager 4923
OpenSearch Service 4960
OpsWorks 4973
AWS OpsWorks CM 5028
組織 5043
AWS Outposts 5079
AWS Payment Cryptography 5083
AWS Payment Cryptography 資料平面 5103
Amazon Pinpoint 5112
Amazon Polly 5135
AWS 價格表 5141
AWS Private CA 5145
AWS Proton 5153
QLDB 5165
Amazon RDS 5187
Amazon RDS 資料服務 5379
Amazon RDS Performance Insights 5383
Amazon Redshift 5397
Amazon Rekognition 5473
AWS RAM 5549
Resource Explorer 5572
資源群組 5593
Resource Groups 標記 API 5606
AWS RoboMaker 5609
Route 53 5646
Route 53 網域註冊 5659
Route 53 設定檔 5685
Route 53 Resolver 5696
Amazon S3 5739
Amazon S3 控制 5828
S3 Glacier 5843
Secrets Manager 5864
安全中樞 5891
Security Lake 5966

AWS Serverless Application Repository 5999
Service Catalog 6001
Service Quotas 6032
Amazon SES 6042
Shield 6054
Signer 6069
Snowball Edge 6079
Amazon SNS 6080
Amazon SQS 6101
Storage Gateway 6121
AWS STS 6124
支援 6133
Amazon SWF 6146
Systems Manager 6162
Amazon Textract 6333
Amazon Transcribe 6344
Amazon Translate 6385
Trusted Advisor 6386
Verified Permissions 6406
VPC Lattice 6430
AWS WAF Classic 6465
AWS WAF Classic 區域性 6470
AWS WAFV2 6475
WorkDocs 6519
Amazon WorkMail 6552
Amazon WorkMail Message Flow 6575
WorkSpaces 6576
X-Ray 6591
Bash 指令碼範例 6609
DynamoDB 6609
Amazon EC2 6707
HealthImaging 6814
IAM 6823
AWS KMS 6887
Amazon S3 6889
AWS STS 6913

安全 6916
資料保護 6916
資料加密 6917
身分和存取權管理 6918
目標對象 6918
使用身分驗證 6918
使用政策管理存取權 6921
AWS 服務 如何使用 IAM 6923
對 AWS 身分和存取進行故障診斷 6923
合規驗證 6925
恢復能力 6926
基礎設施安全性 6926
強制執行最低 TLS 版本 6927
故障診斷錯誤 6928
先嘗試的一般故障診斷 6928
檢查您的 AWS CLI 命令格式 6928
檢查您的 AWS 區域AWS CLI 命令是否正在使用 6929
確認您執行的是最新版本的 AWS CLI 6929
設定 --debug 選項 6930
啟用和檢閱 AWS CLI 命令歷史記錄日誌 6936
確認您的 AWS CLI 已設定 6936
命令未找到錯誤 6936
「aws --version」命令傳回的版本與您安裝的版本不同 6939
解除安裝 後，「aws --version」命令會傳回版本 AWS CLI 6940
已 AWS CLI 處理參數名稱不完整的命令 6941
存取遭拒錯誤 6942
無效的憑證和金鑰錯誤 6943
簽章不相符錯誤 6944
SSL 憑證錯誤 6946
JSON 無效錯誤 6947
其他資源 6949
遷移指南 6950
新功能和變更 6950
AWS CLI 第 2 版新功能 6950
中斷第 1 AWS CLI 版和第 2 AWS CLI 版之間的變更 6952
遷移指示 6958

將版本 1 替換為版本 2 6959
並排安裝 6959
解除安裝 6961
針對 AWS CLI 安裝和解除安裝錯誤進行故障診斷 6964
文件歷史記錄 6965

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。

什麼是 AWS Command Line Interface？

AWS Command Line Interface (AWS CLI) 是一種開放原始碼工具，可讓您使用命令列 Shell 中的命令與 AWS 服務互動。透過最少的 AWS CLI 組態， 可讓您從終端機程式中的命令提示字元，開始執行實作相當於瀏覽器型 所提供功能的 AWS Management Console 命令：

- Linux Shell – 使用常見的 Shell 程式 (例 `bash`、`zsh` 和 `tcsh`)，在 Linux 或 macOS 中執行命令。
- Windows 命令列 – 在 Windows 上，於 Windows 命令提示或 PowerShell 中執行命令。
- 遠端 – 透過遠端終端程式 (例如 PuTTY 或 SSH) 或使用 AWS Systems Manager，在 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上執行命令。

API 和 AWS Management Console 提供 中所有 IaaS （基礎設施即服務） AWS 管理、管理和存取函數 AWS AWS CLI。New AWS IaaS 功能和服務會在啟動時或在啟動後 180 天內透過 API 和 CLI 提供完整 AWS Management Console 功能。

AWS CLI 可讓您直接存取 AWS 服務的公APIs。您可以使用 探索服務的功能 AWS CLI，並開發 shell 指令碼來管理您的 資源。除了低階 API 同等命令之外，數個 AWS 服務還提供 的自訂功能 AWS CLI。自訂功能可能包括較高階的命令，可簡化具有複雜 API 的服務使用。

關於 AWS CLI 第 2 版

AWS CLI 第 2 版是 的最新主要版本， AWS CLI 並支援所有最新的功能。在 第 2 版中推出的某些功能無法回溯至第 1 版，因此您必須升級才能存取這些功能。第 1 版有一些「重大」變更，可能需要變更您的指令碼。如需第 2 版重大變更的清單，請參閱 第 2 AWS CLI 版的遷移指南。

第 2 AWS CLI 版只能作為綁定安裝程式進行安裝。雖然您可能會在套件管理員中找到它，但這些是不受支援且非正式的套件，這些套件並非由 產生或管理 AWS。我們建議您僅 AWS CLI 從官方 AWS 分發點安裝 ，如本指南所述。

若要安裝第 2 AWS CLI 版，請參閱 the section called “安裝/更新”。

若要檢查目前已安裝的版本，請使用下列命令：

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

如需版本歷程記錄，請參閱 GitHub 上的 AWS CLI 第 2 版變更記錄。

開發套件主要版本的維護與支援

如需開發套件主要版本及其基礎相依性之維護與支援的相關資訊，請參閱《AWS 開發套件及工具參考指南》中的以下內容：

- AWS SDKs和工具維護政策
- AWS SDKs和工具版本支援矩陣

Amazon Web Services 簡介

Amazon Web Services (AWS) 集合了各項數位基礎設施服務，讓開發人員可在開發應用程式時加以利用。服務包括運算、儲存、資料庫和應用程式同步 （ 訊息和佇列 ）。 AWS 使用pay-as-you-go服務模型。您只需為您 (或您應用程式) 使用的服務付費。此外，為了讓 成為 AWS 更接近的原型設計和實驗平台， AWS 提供免費用量方案。在此方案中，特定用量層級以下的服務皆為免費。如需 AWS 成本和免費方案的詳細資訊，請參閱 AWS 免費方案。若要取得 AWS 帳戶，請開啟AWS 首頁，然後選擇建立 AWS 帳戶。

關於 AWS CLI 使用者指南中的範例

本指南中的 AWS Command Line Interface (AWS CLI) 範例使用下列慣例進行格式化：

- 提示 – 命令提示使用 Linux 提示字元，並顯示為 ($)。對於 Windows 特有的命令，`C:\>` 將其用作提示。鍵入命令時，請不要包含該提示。
- 目錄 – 命令必須從特定的目錄執行時，該目錄名稱會顯示在提示符號的前方。
- 使用者輸入 – 於命令列輸入的命令文字採用 **`user input`** 格式。
- 可取代文字 – 可變文字，包括您選擇的資源名稱，或您必須包含在命令中的 AWS 服務產生的 IDs，格式為`#####`。在多行命令或需要特定鍵盤輸入的命令中，鍵盤命令也可以顯示為可取代文字。
- 輸出 – AWS 服務傳回的輸出會顯示在使用者輸入下，格式為 `computer output`。

下列 **`aws configure`** 命令範例示範了使用者輸入、可取代文字以及輸出：

1. 請在命令列中輸入 **`aws configure`**，再按下 Enter。
2. AWS CLI 輸出一行文字，提示您輸入其他資訊。
3. 請依次輸入各個存取金鑰，並按下 Enter。
4. 然後，以顯示的格式輸入 AWS 區域名稱，按 Enter，然後按 Enter 以略過輸出格式設定。

5. 最終 Enter 命令會顯示為可取代文字，因為該行沒有使用者輸入。

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

下面的例子顯示輸出中有一個簡單的命令。若要使用此範例，請輸入命令的完整文字 (提示後方反白顯示的文字)，並按下 Enter。安全性群組的名稱 *my-sg* 可用您想要的安全群組名稱取代。JSON 文件 (包含大括弧) 為輸出。如果您將 CLI 設為以文字格式或表格格式輸出，則輸出的格式將會有所差異。JSON 為預設輸出格式。

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

的其他文件和資源 AWS CLI

AWS CLI 文件和資源

除了本使用者指南之外，當您使用 時，以下是寶貴的線上資源 AWS CLI。

- AWS CLI 第 2 版參考指南
- AWS CLI Bash 指令碼程式碼範例儲存庫。開放原始碼 bash 指令碼範例。Bash 指令碼範例託管於 GitHub 上的 AWS 程式碼範例儲存庫中。
- AWS CLI GitHub 儲存庫。您可以在 GitHub AWS CLI 上檢視和撥出 的原始程式碼。加入 GitHub 上的使用者社群，以提供意見回饋、請求功能，並提交您自己的貢獻。這包括檢視和提供 AWS CLI 文件的命令範例。
- AWS CLI 別名範例儲存庫 您可以在 GitHub AWS CLI 上檢視 和 分支別名範例。
- AWS CLI 第 2 版變更日誌

AWS SDKs和工具

根據您的使用案例，您可能想要選擇其中一個 AWS SDKs或工具，以更符合您的需求：

- AWS SDKs和工具參考指南
- 適用於 C++ 的 AWS SDK
- 適用於 Go 的 AWS SDK
- 適用於 Java 的 AWS SDK
- 適用於 JavaScript 的 AWS SDK
- 適用於 Kotlin 的 AWS SDK
- 適用於 .NET 的 AWS SDK
- AWS SDK for Python (Boto)
- 適用於 PHP 的 AWS SDK
- AWS Tools for PowerShell
- 適用於 Ruby 的 AWS SDK
- 適用於 Rust 的 AWS SDK
- 適用於 SAP ABAP 的 AWS SDK
- 適用於 Swift 的 AWS SDK
- AWS Amplify

開始使用 AWS CLI

本章提供開始使用第 2 版 AWS Command Line Interface (AWS CLI) 的步驟，並提供相關說明的連結。

1. 完成所有先決條件 - 若要使用 存取 AWS 服務 AWS CLI，您至少需要 AWS 帳戶 和 IAM 登入資料。為了提高 AWS 帳戶的安全，我們建議您不要使用根帳戶憑證。您應該建立一個具最低權限的使用者，以便為您將在 AWS中執行的任務提供存取憑證。
2. AWS CLI 使用下列其中一種方法安裝或存取 ：
   - （建議） the section called “安裝/更新”。
   - the section called “過去版本”。 如果您的團隊將其工具與特定版本對齊，則主要會使用安裝特定版本。
   - the section called “從來源建置和安裝”。 AWS CLI 從 GitHub 來源建置 是更深入的方法，主要供在不受預先建置安裝程式直接支援的平台上工作的客戶使用。
   - the section called “Amazon ECR Public/Docker”.
   - 使用 從瀏覽器存取 AWS 主控台中的第 2 AWS CLI 版 AWS CloudShell。如需詳細資訊，請參閱「AWS CloudShell 使用者指南」。
3. 存取 之後 AWS CLI，請使用您的 IAM 登入資料來設定您的 AWS CLI ，以供第一次使用。

> **對安裝程式進行故障診斷或設定錯誤**
>
> 如果您在安裝、解除安裝或設定 之後發生問題 AWS CLI，請參閱 故障診斷錯誤 以取得疑難排解步驟。

主題

- 使用第 2 AWS CLI 版的先決條件
- 安裝或更新至最新版本的 AWS CLI
- 安裝 AWS CLI 第 2 版的過去版本
- AWS CLI 從來源建置和安裝
- 執行 的官方 Amazon ECR Public 或 Docker 映像 AWS CLI
- 設定 AWS CLI

使用第 2 AWS CLI 版的先決條件

若要使用 存取 AWS 服務 AWS CLI，您需要 AWS 帳戶 和 IAM 登入資料。執行 AWS CLI 命令時，AWS CLI 需要能夠存取這些 AWS 登入資料。為了提高 AWS 帳戶的安全性，建議您不要使用您的根帳戶登入資料。您應該建立具有最低權限的使用者，以提供存取憑證給您將在其中執行的任務 AWS。

主題

- 建立 IAM 或 IAM Identity Center 管理員帳戶
- 後續步驟

建立 IAM 或 IAM Identity Center 管理員帳戶

您必須先建立 IAM 或 IAM Identity Center 帳戶 AWS CLI，才能設定 。

若要建立管理員使用者，請選擇下列其中一個選項。

選擇一種管理管理員的方式	到	根據	您也可以
在 IAM Identity Center (建議)	使用短期憑證存取 AWS。 這與安全性最佳實務一致。有關最佳實務的資訊，請參閱 IAM 使用者指南中的 IAM 安全最佳實務。	請遵循 AWS IAM Identity Center 使用者指南的入門中的說明。	在 AWS Command Line Interface 使用者指南中設定 AWS CLI 以使用 來設定 AWS IAM Identity Center程式設計存取。
在 IAM 中 (不建議使用)	使用長期憑證存取 AWS。	遵循《IAM 使用者指南》中建立緊急存取的 IAM 使用者的指示。	《IAM 使用者指南》中的透過管理 IAM 使用者的存取金鑰來設定程式設計存取。

後續步驟

建立 AWS 帳戶 和 IAM 登入資料後，若要使用 AWS CLI ，您可以執行下列其中一項操作：

- 在電腦上安裝最新版本的 第 2 AWS CLI 版。
- 在電腦上安裝過去版本的 第 2 AWS CLI 版。
- 使用 Docker 映像從您的電腦存取第 2 AWS CLI 版。 ???
- 使用 從瀏覽器存取 AWS 主控台中的第 2 AWS CLI 版 AWS CloudShell。如需詳細資訊，請參閱 AWS CloudShell 使用者指南。

安裝或更新至最新版本的 AWS CLI

本主題說明如何在支援的作業系統上安裝或更新最新版本的 AWS Command Line Interface (AWS CLI)。如需最新版本 的資訊 AWS CLI，請參閱 GitHub 上的AWS CLI 第 2 版變更日誌。

若要安裝過去版本的 AWS CLI，請參閱 the section called “過去版本”。如需解除安裝說明，請參閱 解除安裝。

> **Important**
>
> AWS CLI 第 1 版和第 2 版使用相同的aws命令名稱。如果您先前已安裝 AWS CLI 版本 1，請參閱 第 2 AWS CLI 版的遷移指南。

主題

- AWS CLI 安裝和更新指示
- 針對 AWS CLI 安裝和解除安裝錯誤進行故障診斷
- 後續步驟

AWS CLI 安裝和更新指示

如需安裝說明，請根據您的作業系統展開相應的區段。

Linux

安裝和更新需求

- 您必須能夠解壓縮或「解壓縮」下載的套件。如果您的作業系統沒有內建 `unzip` 命令，請使用對等的命令。
- AWS CLI 使用 `glibc`、 `groff`和 `less`。依預設，這些會包含在 Linux 的大多數主要發行版中。
- 我們支援 AWS CLI 64 位元版本的最新 CentOS、Fedora、Ubuntu、Amazon Linux 1、Amazon Linux 2、Amazon Linux 2023 和 Linux ARM 發行版本。
- 由於 AWS 不會維護 以外的第三方儲存庫`snap`，因此我們無法保證它們包含最新版本的 AWS CLI。

安裝或更新 AWS CLI

> **Warning**
>
> 如果這是您第一次在 Amazon Linux 上更新 ，若要安裝最新版本的 AWS CLI，您必須使用以下命令解除安裝預先安裝的`yum`版本：
>
> ```
> $ sudo yum remove awscli
> ```
>
> 移除 yum 安裝 AWS CLI 之後，請遵循下列 Linux 安裝指示。

您可以使用下列 AWS CLI 其中一種方法來安裝 ：

- 命令列安裝程式是版本控制的好選項，因為您可以指定要安裝的版本。此選項不會自動更新，而且您每次更新時都必須下載新的安裝程式，以覆寫先前的版本。
- 正式支援的**snap**套件是讓 一律擁有最新版本的良好選項， AWS CLI 因為快照套件會自動重新整理。不支援選取 的次要版本， AWS CLI 因此如果您的團隊需要鎖定版本，則不是最佳安裝方法。

Command line installer - Linux x86 (64-bit)

若要更新目前安裝的 AWS CLI，請在每次更新時下載新的安裝程式以覆寫先前的版本。請依照命令列中的下列步驟，在 Linux AWS CLI 上安裝 。

以下是提供基本安裝的單一複製和貼上群組中的快速安裝步驟。如需引導說明，請參閱以下步驟。

> **Note**
>
> （選用）下列命令區塊會下載並安裝 ，AWS CLI 而無需先驗證下載的完整性。若要驗證下載的完整性，請使用以下逐步說明。

若要安裝 AWS CLI，請執行以下命令。

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

若要更新 AWS CLI目前的安裝，新增現有的符號連結和安裝程式資訊,以使用 `--bin-dir`、`--install-dir` 和 `--update` 參數建構 `install` 命令。下列命令區塊使用 */usr/local/bin* 的範例符號連結和 */usr/local/aws-cli* 的範例安裝程式位置，為目前使用者在 AWS CLI 本機安裝 。

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

引導式安裝步驟

1. 使用下列其中一種方式下載安裝檔案：

   - 使用 **`curl`** 命令 – `-o` 選項會指定已下載套件寫入的檔案名稱。下列範例命令的選項會使下載的檔案以本機時間 `awscliv2.zip` 寫入目前的目錄。

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

   - 從 URL 下載 – 若要使用瀏覽器下載安裝程式，請使用下列 URL：https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip

2. (可選) 驗證所下載 zip 檔案的完整性

   如果您選擇`.zip`在上述步驟中手動下載 AWS CLI 安裝程式套件，則可以使用 GnuPG工具，使用下列步驟來驗證簽章。

AWS CLI 安裝程式套件.zip檔案使用 PGP 簽章進行密碼編譯簽署。如果檔案有任何損壞或更改，此驗證會失敗，且您不應該繼續進行安裝。

a. 使用您的套件管理工具下載並安裝 gpg 命令。如需有關 GnuPG 的詳細資訊，請參閱 GnuPG 網站。

b. 若要建立公開金鑰檔案，請建立文字檔案，然後貼上下列文字。

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

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

以下公開金鑰詳細資訊僅供參考。

```
Key ID:          A6310ACC4672475C
Type:            RSA
```

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

c. 使用下列命令匯入 AWS CLI 公有金鑰，以您建立之公有金鑰的檔案名稱取代 *public-key-file-name*。

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

d. 下載您下載之套件的 AWS CLI 簽章檔案。它會有與其對應的 `.zip` 檔案具有相同的路徑和名稱，但具有副檔名 `.sig`。在下列範例中，我們將它作為名稱 `awscliv2.sig` 的檔案儲存在目前的資料夾。

對於最新版本的 AWS CLI，請使用下列命令區塊：

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

e. 驗證簽章，並將下載的 `.sig` 和 `.zip` 檔案名稱以參數形式傳遞給 gpg 命令。

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

輸出應看起來如下列內容。

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

> **Important**
>
> 輸出中的警告是預期的，並不表示有問題。這是因為您的個人 PGP 金鑰（如果您有）和 AWS CLI PGP 金鑰之間沒有信任鏈。如需詳細資訊，請參閱「信任網路」。

3. 解壓縮安裝程式。如果您的 Linux 發行版沒有內建 unzip 命令，請使用類似命令解壓縮。下列範例命令會解壓縮套件，並在名為 aws 的目前目錄下建立目錄。

```
$ unzip awscliv2.zip
```

> **Note**
>
> 從舊版更新時，unzip 命令會提示覆寫現有檔案。若要略過這些提示 (例如使用指令碼自動化)，對於 unzip 請使用 -u 更新標記。這個標記會視需要自動更新現有檔案和建立新檔案。
>
> ```
> $ unzip -u awscliv2.zip
> ```

4. 執行安裝程式。安裝命令會使用新解壓縮的 aws 目錄中名為 install 的檔案。依預設，檔案會全部安裝到 /usr/local/aws-cli，並在 /usr/local/bin 中建立符號連結。命令包括 sudo，以授予寫入權限給這些資料夾。

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

如果指定您已具有寫入權限的資料夾，則可以在不使用 sudo 的情況下進行安裝。使用下列 install 命令指示來指定安裝位置：

- 請確定您提供給 -i 和 -b 參數的路徑不包含任何包含空格字元或其他空格字元的磁碟區名稱或目錄名稱。如果有空格，則安裝會失敗。
- --install-dir 或 -i – 此選項指定要將所有檔案複製到的目錄。

  預設值為 /usr/local/aws-cli。
- --bin-dir 或 -b – 此選項會指定安裝目錄中的 aws 主程式以符號方式連結至指定路徑的檔案 aws。您必須具有指定目錄的寫入許可。建立路徑中已存在目錄的符號連結，可讓您不需將安裝目錄新增到使用者的 $PATH 變數。

預設值為 /usr/local/bin。

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

> **Note**
>
> 若要更新目前安裝的 AWS CLI，請新增現有的符號連結和安裝程式資訊，以使用 --update 參數建構 install命令。
>
> ```
> $ sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/aws-cli --update
> ```
>
> 若要找到現有的符號連結和安裝目錄，請執行以下步驟：
>
> 1. 使用 which 命令來查找您的符號連結。這提供您與 --bin-dir 參數一起使用的路徑。
>
>    ```
>    $ which aws
>    /usr/local/bin/aws
>    ```
>
> 2. 使用 ls 命令來查找符號連結指向的目錄。這提供您與 --install-dir 參數一起使用的路徑。
>
>    ```
>    $ ls -l /usr/local/bin/aws
>    lrwxrwxrwx 1 ec2-user ec2-user 49 Oct 22 09:49 /usr/local/bin/aws -> /usr/local/aws-cli/v2/current/bin/aws
>    ```

5. 請使用以下命令來確認安裝。

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

如果找不到 aws 命令，您可能需要重新啟動終端機或遵循 故障診斷錯誤 中的故障診斷。

Command line - Linux ARM

若要更新目前安裝的 AWS CLI，請在每次更新時下載新的安裝程式以覆寫先前的版本。請依照命令列中的下列步驟，在 Linux AWS CLI 上安裝 。

以下是提供基本安裝的單一複製和貼上群組中的快速安裝步驟。如需引導說明，請參閱以下步驟。

> **Note**
>
> （選用）下列命令區塊會下載並安裝 ， AWS CLI 而無需先驗證下載的完整性。若要驗證下載的完整性，請使用以下逐步說明。

若要安裝 AWS CLI，請執行下列命令。

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

若要更新 AWS CLI目前的安裝，新增現有的符號連結和安裝程式資訊,以使用 `--bin-dir`、`--install-dir` 和 `--update` 參數建構 `install` 命令。下列命令區塊使用 */usr/local/bin* 的範例符號連結，以及 */usr/local/aws-cli* 的範例安裝程式位置。

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

引導式安裝步驟

1. 使用下列其中一種方式下載安裝檔案：

   - 使用 **`curl`** 命令 – `-o` 選項會指定已下載套件寫入的檔案名稱。下列範例命令的選項會使下載的檔案以本機時間 `awscliv2.zip` 寫入目前的目錄。

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

   - 從 URL 下載 – 若要使用瀏覽器下載安裝程式，請使用下列 URL：https://awscli.amazonaws.com/awscli-exe-linux-aarch64.zip

2. (可選) 驗證所下載 zip 檔案的完整性

   如果您選擇`.zip`在上述步驟中手動下載 AWS CLI 安裝程式套件，則可以使用 GnuPG工具，使用下列步驟來驗證簽章。

   AWS CLI 安裝程式套件`.zip`檔案使用 PGP 簽章進行密碼編譯簽署。如果檔案有任何損壞或更改，此驗證會失敗，且您不應該繼續進行安裝。

   a. 使用您的套件管理工具下載並安裝 gpg 命令。如需有關 GnuPG 的詳細資訊，請參閱 GnuPG 網站。

   b. 若要建立公開金鑰檔案，請建立文字檔案，然後貼上下列文字。

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

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

以下公開金鑰詳細資訊僅供參考。

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

c. 使用下列命令匯入 AWS CLI 公有金鑰，以您建立之公有金鑰的檔案名稱取代 *public-key-file-name*。

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

d. 下載您下載之套件的 AWS CLI 簽章檔案。它會有與其對應的 `.zip` 檔案具有相同的路徑和名稱，但具有副檔名 `.sig`。在下列範例中，我們將它作為名稱 `awscliv2.sig` 的檔案儲存在目前的資料夾。

對於最新版本的 AWS CLI，請使用下列命令區塊：

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

e. 驗證簽章，並將下載的 `.sig` 和 `.zip` 檔案名稱以參數形式傳遞給 gpg 命令。

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

輸出應看起來如下列內容。

```
gpg: Signature made Mon Nov  4 19:00:01 2019 PST
gpg:                using RSA key FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC
 4672 475C
gpg: Good signature from "AWS CLI Team <aws-cli@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
```

```
gpg:          There is no indication that the signature belongs to the
 owner.
Primary key fingerprint: FB5D B77F D5C1 18B8 0511  ADA8 A631 0ACC 4672 475C
```

> **Important**
>
> 輸出中的警告是預期的，並不表示有問題。這是因為您的個人 PGP 金鑰（如果您有）和 AWS CLI PGP 金鑰之間沒有信任鏈。如需詳細資訊，請參閱「信任網路」。

3. 解壓縮安裝程式。如果您的 Linux 發行版沒有內建 unzip 命令，請使用類似命令解壓縮。下列範例命令會解壓縮套件，並在名為 aws 的目前目錄下建立目錄。

```
$ unzip awscliv2.zip
```

> **Note**
>
> 從舊版更新時，unzip 命令會提示覆寫現有檔案。若要略過這些提示 (例如使用指令碼自動化)，對於 unzip 請使用 -u 更新標記。這個標記會視需要自動更新現有檔案和建立新檔案。
>
> ```
> $ unzip -u awscliv2.zip
> ```

4. 執行安裝程式。安裝命令會使用新解壓縮的 aws 目錄中名為 install 的檔案。依預設，檔案會全部安裝到 /usr/local/aws-cli，並在 /usr/local/bin 中建立符號連結。命令包括 sudo，以授予寫入權限給這些資料夾。

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

如果指定您已具有寫入權限的資料夾，則可以在不使用 sudo 的情況下進行安裝。使用下列 install 命令指示來指定安裝位置：

- 請確定您提供給 -i 和 -b 參數的路徑不包含任何包含空格字元或其他空格字元的磁碟區名稱或目錄名稱。如果有空格，則安裝會失敗。
- --install-dir 或 -i – 此選項指定要將所有檔案複製到的目錄。

  預設值為 /usr/local/aws-cli。

- `--bin-dir` 或 `-b` – 此選項會指定安裝目錄中的 aws 主程式以符號方式連結至指定路徑的檔案 aws。您必須具有指定目錄的寫入許可。建立路徑中已存在目錄的符號連結，可讓您不需將安裝目錄新增到使用者的 $PATH 變數。

  預設值為 /usr/local/bin。

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

> **Note**
>
> 若要更新目前安裝的 AWS CLI，請新增現有的符號連結和安裝程式資訊，以使用 `--update` 參數建構 `install`命令。
>
> ```
> $ sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/aws-cli --update
> ```
>
> 若要找到現有的符號連結和安裝目錄，請執行以下步驟：
>
> 1. 使用 `which` 命令來查找您的符號連結。這提供您與 `--bin-dir` 參數一起使用的路徑。
>
>    ```
>    $ which aws
>    /usr/local/bin/aws
>    ```
>
> 2. 使用 `ls` 命令來查找符號連結指向的目錄。這提供您與 `--install-dir` 參數一起使用的路徑。
>
>    ```
>    $ ls -l /usr/local/bin/aws
>    lrwxrwxrwx 1 ec2-user ec2-user 49 Oct 22 09:49 /usr/local/bin/aws -> /usr/local/aws-cli/v2/current/bin/aws
>    ```

5. 請使用以下命令來確認安裝。

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

如果找不到 aws 命令，您可能需要重新啟動終端機或遵循 故障診斷錯誤 中的故障診斷。

Snap package

我們在 AWS CLI 上提供官方 AWS 支援的 版本snap。如果您想要在您的系統上一律 AWS CLI 安裝最新版本的 ，則 Snap 套件會在自動更新時為您提供此功能。不支援選取 的次要版本 AWS CLI ，因此如果您的團隊需要鎖定版本，則它不是最佳安裝方法。如果您想要安裝特定次要版本的 AWS CLI，建議您使用命令列安裝程式。

1. 如果您的 Linux 平台尚未snap安裝 ，snap請在您的平台上安裝 。

   a. 如需安裝 的資訊snap，請參閱 Snap 文件中的安裝協助程式。

   b. 您可能需要重新啟動系統，才能正確更新PATH變數。如果您有安裝問題，請遵循 Snap 文件中的修正常見問題中的步驟。

   c. 若要驗證 snap 是否正確安裝，請執行下列命令。

   ```
   $ snap version
   ```

2. 針對 執行下列snap install命令 AWS CLI。

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

   根據您的許可，您可能需要將 sudo 新增至命令。

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

   > **Note**
   >
   > 若要檢視 的快照儲存庫 AWS CLI，包括其他snap指示，請參閱 Canonical Snapcraft 網站上的 aws-cli頁面。

3. 確認 已正確 AWS CLI 安裝。

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

   如果發生錯誤，請參閱 故障診斷 的錯誤 AWS CLI。

macOS

安裝和更新需求

- 我們在 macOS 第 11 版及更新版本 AWS CLI 上支援 。如需詳細資訊，請參閱AWS 開發人員工具部落格上的 AWS CLI v2 的 macOS 支援政策更新。
- 由於 AWS 不會維護第三方儲存庫，因此我們無法保證它們包含最新版本的 AWS CLI。

macOS 版本支援矩陣

AWS CLI 版本	支援的 macOS 版本
2.21.0 – 目前版本	11+
2.17.0 –2.20.0	10.15+
2.0.0 – 2.16.12	10.14 及更新版本

安裝或更新 AWS CLI

如果您要更新至最新版本，請使用您在目前版本中使用的相同安裝方法。您可以透過下列方式在 macOS AWS CLI 上安裝 。

GUI installer

下列步驟說明如何 AWS CLI 使用標準 macOS 使用者介面和瀏覽器安裝最新版本的 。

1. 在瀏覽器中下載 macOS pkg 檔案：https://awscli.amazonaws.com/AWSCLIV2.pkg
2. 請依照畫面上的指示，執行您下載的檔案。您可以選擇 AWS CLI 以下列方式安裝 ：
   - 對於電腦上的所有使用者 (需要 **sudo**)
     - 您可以安裝到任何資料夾，或選擇建議的 `/usr/local/aws-cli` 的預設資料夾。
     - 安裝程式會自動在 `/usr/local/bin/aws` 建立符號連結，其連結至您所選擇的安裝資料夾中的主程式。
   - 只對於目前的使用者 (不需要 **sudo**)
     - 您可以安裝到任何具有寫入權限的資料夾。

- 由於標準的使用者許可，在安裝程式完成之後，您的 $PATH 必須在命令提示字元中使用下列命令，手動建立指向 aws 和 aws_completer 程式的符號連結檔案。符號連結的預設位置為 /usr/local/bin/：

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

如果您沒有 資料夾的寫入許可，您可能需要在 命令sudo中使用 。下列範例使用 sudo搭配 中符號連結的預設位置/usr/local/bin/：

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

> **Note**
>
> 您可以在安裝程式中的任何位置按 Cmd+L 來檢視安裝的偵錯日誌。這會開啟一個日誌窗格，使您能夠篩選和儲存日誌。日誌檔也會自動儲存到 /var/log/install.log。

3. 若要確認 Shell 可以在您的 aws 中尋找並執行 $PATH 命令，請使用下列命令。

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

如果找不到 aws 命令，您可能需要重新啟動終端機或遵循 故障診斷錯誤 中的故障診斷。

Command line installer - All users

如果您具有 sudo 許可，則可以為電腦上的所有使用者安裝 AWS CLI 。我們提供一組易於複製和貼上的步驟。請參閱後續步驟中每一行的描述。

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

引導式安裝說明

1. 使用 curl 命令下載檔案。在此範例中，-o 選項會指定系統寫入下載套件的檔案名稱。在前述的範例中，檔案會寫入目前的資料夾中的 AWSCLIV2.pkg。

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

2. 執行標準 macOS installer 程式，將下載的 .pkg 檔案指定為來源。使用 -pkg 參數可指定要安裝的套件名稱，以及磁碟機要安裝套件的 -target / 參數。檔案會安裝到 /usr/local/aws-cli，並且會在 /usr/local/bin 中自動建立符號連結。您必須在命令上包含 sudo，才能將寫入權限授予這些資料夾。

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

安裝完成後，偵錯記錄檔會寫入 /var/log/install.log。

3. 若要確認 Shell 可以在您的 aws 中尋找並執行 $PATH 命令，請使用下列命令。

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

如果找不到 aws 命令，您可能需要重新啟動終端機或遵循 故障診斷錯誤 中的故障診斷。

Command line - Current user

1. 若要指定要安裝 AWS CLI 的資料夾，您必須建立具有任何檔案名稱的 XML 檔案。這個檔案是 XML 格式化的檔案，看起來如下列範例。如圖所示保留所有值，但您必須將第 9 行中的路徑 */Users/myusername* 替換為要安裝 AWS CLI 的資料夾路徑。資料夾必須已存在，否則命令會失敗。下列 XML 範例名為 *choices.xml*，指定安裝程式在資料夾 AWS CLI 中安裝 /Users/myusername，並在其中建立名為 的資料夾aws-cli。

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

```
      <string>customLocation</string>
      <key>attributeSetting</key>
      <string>/Users/myusername</string>
      <key>choiceIdentifier</key>
      <string>default</string>
    </dict>
  </array>
</plist>
```

2. 使用 curl 命令下載 pkg 安裝程式。在此範例中，-o 選項會指定系統寫入下載套件的檔案名稱。在前述的範例中，檔案會寫入目前的資料夾中的 AWSCLIV2.pkg。

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

3. 現在您可以使用下列選項來執行標準 macOS installer 程式：

   - 使用參數 -pkg 指定要安裝的套件的名稱。
   - 若要指定僅限目前使用者的安裝，您必須設定參數 -target 為 CurrentUserHomeDirectory。
   - 指定您在 -applyChoiceChangesXML 參數中建立的 XML 檔案的路徑 (相對於目前的資料夾) 和名稱。

   下列範例會在 資料夾 AWS CLI 中安裝 /Users/myusername/aws-cli。

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

4. 由於標準使用者許可通常不允許寫入 $PATH 中的資料夾，因此此模式下的安裝程式不會嘗試新增連至 aws 和 aws_completer 的符號連結。若要 AWS CLI 讓 正確執行，您必須在安裝程式完成後手動建立符號連結。如果您的 $PATH 包含可以寫入的資料夾，則如果將該資料夾指定為目標路徑，則可執行下列命令而無需 sudo。如果您的 $PATH 中沒有可寫入的資料夾，則必須使 sudo 以取得寫入指定目標資料夾的許可。符號連結的預設位置為 /usr/local/bin/。以您的 AWS CLI 安裝路徑替換 folder/installed。

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

安裝完成後，偵錯記錄檔會寫入 /var/log/install.log。

5. 若要確認 Shell 可以在您的 aws 中尋找並執行 $PATH 命令，請使用下列命令。

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

如果找不到 aws 命令，您可能需要重新啟動終端機或遵循 故障診斷錯誤 中的故障診斷。

Windows

安裝和更新需求

- 我們在 Microsoft 支援的 64 位元 Windows 版本 AWS CLI 上支援 。
- 安裝軟體的管理員權限

安裝或更新 AWS CLI

若要更新目前在 Windows AWS CLI 上安裝的 ，請在每次更新時下載新的安裝程式以覆寫先前的版本。 AWS CLI 會定期更新。若要檢視最新版本發佈的時間，請參閱 GitHub 上的 AWS CLI 第 2 版變更日誌。

1. 下載並執行適用於 Windows 的 AWS CLI MSI 安裝程式 (64 位元) ：

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

   您也可以執行 msiexec 命令來執行 MSI 安裝程式。

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

   如需可與 msiexec 一起使用的各種參數，請參閱 Microsoft Docs 網站上的 msiexec。例如，您可以使用 /qn 標記進行無訊息安裝。

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

2. 若要確認安裝，請開啟開始功能表、搜尋 cmd 以開啟命令提示字元視窗，然後在命令提示字元中使用 aws --version 命令。

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

如果 Windows 找不到程式，您可能需要關閉命令提示字元再重新開啟，以重新整理路徑，或遵循故障診斷錯誤 中的故障診斷。

針對 AWS CLI 安裝和解除安裝錯誤進行故障診斷

如果您在安裝或解除安裝 之後遇到問題 AWS CLI，請參閱 故障診斷錯誤 以取得疑難排解步驟。如需最相關的故障診斷步驟，請參閱 the section called “命令未找到錯誤”、the section called “「aws --version」命令傳回的版本與您安裝的版本不同” 和 the section called “解除安裝 後，「aws --version」命令會傳回版本 AWS CLI”。

後續步驟

成功安裝 後 AWS CLI，您可以安全地刪除下載的安裝程式檔案。完成 中的步驟the section called “先決條件”並安裝 後 AWS CLI，您應該執行 the section called “設定”。

安裝 AWS CLI 第 2 版的過去版本

本主題說明如何在支援的作業系統上安裝 AWS Command Line Interface 第 2 版 (AWS CLI) 的過去版本。如需 AWS CLI 第 2 版的資訊，請參閱 GitHub 上的第 AWS CLI 2 版變更日誌。

AWS CLI 第 2 版安裝說明：

Linux

安裝要求

- 您知道要安裝的 AWS CLI 版本 2。如需版本清單，請參閱 GitHub 上的 AWS CLI 第 2 版變更日誌。
- 您必須能夠解壓縮或「解壓縮」下載的套件。如果您的作業系統沒有內建 `unzip` 命令，請使用對等的命令。
- 第 2 AWS CLI 版使用 `glibc`、 `groff`和 `less`。依預設，這些會包含在 Linux 的大多數主要發行版中。
- 我們支援 CentOS、Fedora、Ubuntu、Amazon Linux AWS CLI 1、Amazon Linux 2 和 Linux ARM 最近發行版本的 64 位元版本 2。

- 由於 AWS 不會維護第三方儲存庫，因此我們無法保證它們包含最新版本的 AWS CLI。

安裝說明

請依照命令列中的下列步驟，在 Linux AWS CLI 上安裝 。

我們會根據您使用 64 位元 Linux 或 Linux ARM，在一個易於複製和貼上群組中提供步驟。請參閱後續步驟中每一行的描述。

Linux x86 (64-bit)

> **Note**
>
> （選用）下列命令區塊會下載並安裝 ， AWS CLI 而無需先驗證下載的完整性。若要驗證下載的完整性，請使用以下逐步說明。

如需版本清單，請參閱 GitHub 上的 AWS CLI 第 2 版變更日誌。

若要安裝 AWS CLI，請執行下列命令。

若要指定版本，請在檔案名稱中附加連字號和版本號碼。在這個範例中，版本 *2.0.30* 的檔案名稱為 `awscli-exe-linux-x86_64-2.0.30.zip`，因此使用以下命令：

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

若要更新 AWS CLI目前的安裝，新增現有的符號連結和安裝程式資訊,以使用 `--bin-dir`、`--install-dir` 和 `--update` 參數建構 `install` 命令。下列命令區塊使用 */usr/local/bin* 的範例符號連結，以及 */usr/local/aws-cli* 的範例安裝程式位置。

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

Linux ARM

> **Note**
>
> （選用）下列命令區塊會下載並安裝 ，AWS CLI 而無需先驗證下載的完整性。若要驗證下載的完整性，請使用以下逐步說明。

如需版本清單，請參閱 GitHub 上的 AWS CLI 第 2 版變更日誌。

若要安裝 AWS CLI，請執行下列命令。

若要指定版本，請在檔案名稱中附加連字號和版本號碼。在這個範例中，版本 *2.0.30* 的檔案名稱為 `awscli-exe-linux-aarch64-2.0.30.zip`，因此使用以下命令：

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

若要更新 AWS CLI目前的安裝，新增現有的符號連結和安裝程式資訊,以使用 `--bin-dir`、`--install-dir` 和 `--update` 參數建構 `install` 命令。下列命令區塊使用 */usr/local/bin* 的範例符號連結，以及 */usr/local/aws-cli* 的範例安裝程式位置。

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

1. 使用下列其中一種方式下載安裝檔案：

   Linux x86 (64-bit)

   - 使用 **`curl`** 命令 – `-o` 選項會指定已下載套件寫入的檔案名稱。下列範例命令的選項會使下載的檔案以本機時間 `awscliv2.zip` 寫入目前的目錄。

     若要指定版本，請在檔案名稱中附加連字號和版本號碼。在這個範例中，版本 *2.0.30* 的檔案名稱為 `awscli-exe-linux-x86_64-2.0.30.zip`，因此使用以下命令：

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

如需版本清單，請參閱 GitHub 上的 AWS CLI 第 2 版變更日誌。

- 從 URL 下載–

  在您的瀏覽器中，將連字號和版本號碼附加至檔案名稱， AWS CLI 以下載特定版本的 。

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

  在此範例中，版本 *2.0.30* 的檔案名為 awscli-exe-linux-x86_64-2.0.30.zip，因此使用以下連結：https://awscli.amazonaws.com/awscli-exe-linux-x86_64-2.0.30.zip

Linux ARM

- 使用 **curl** 命令 – -o 選項會指定已下載套件寫入的檔案名稱。下列範例命令的選項會使下載的檔案以本機時間 awscliv2.zip 寫入目前的目錄。

  若要指定版本，請在檔案名稱中附加連字號和版本號碼。在這個範例中，版本 *2.0.30* 的檔案名稱為 awscli-exe-linux-aarch64-2.0.30.zip，因此使用以下命令：

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

- 從 URL 下載–

  在您的瀏覽器中，將連字號和版本號碼附加至檔案名稱， AWS CLI 以下載特定版本的 。

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

  在這個範例中，版本 *2.0.30* 的檔案名稱為 awscli-exe-linux-aarch64-2.0.30.zip，因此使用以下連結：https://awscli.amazonaws.com/awscli-exe-linux-aarch64-2.0.30.zip

2. (可選) 驗證所下載 zip 檔案的完整性

如果您選擇`.zip`在上述步驟中手動下載 AWS CLI 安裝程式套件，則可以使用 GnuPG工具，使用下列步驟來驗證簽章。

AWS CLI 安裝程式套件`.zip`檔案使用 PGP 簽章進行密碼編譯簽署。如果檔案有任何損壞或更改，此驗證會失敗，且您不應該繼續進行安裝。

a. 使用您的套件管理工具下載並安裝 gpg 命令。如需有關 GnuPG 的詳細資訊，請參閱 GnuPG 網站。

b. 若要建立公開金鑰檔案，請建立文字檔案，然後貼上下列文字。

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

mQINBF2Cr7UBEADJZHcgusOJl7ENSyumXh85z0TRV0xJorM2B/JL0kHOyigQluUG
ZMLhENaG0bYatdrKP+3H91lvK050pXwnO/R7fB/FSTouki4ciIx5OuLlnJZIxSzx
PqGl0mkxImLNbGWoi6Lto0LYxqHN2iQtzlwTVmq9733zd3XfcXrZ3+LblHAgEt5G
TfNxEKJ8soPLyWmwDH6HWCnjZ/aIQRBTIQ05uVeEoYxSh6wOai7ss/KveoSNBbYz
gbdzoqI2Y8cgH2nbfgp3DSasaLZEdCSsIsK1u05CinE7k2qZ7KgKAUIcT/cR/grk
C6VwsnDU0OUCideXcQ8WeHutqvgZH1JgKDbznoIzeQHJD238GEu+eKhRHcz8/jeG
94zkcgJOz3KbZGYMiTh277Fvj9zzvZsbMBCedV1BTg3TqgvdX4bdkhf5cH+7NtWO
lrFj6UwAsGukBTAOxC0l/dnSmZhJ7Z1KmEWilro/gOrjtOxqRQutlIqG22TaqoPG
fYVN+en3Zwbt97kcgZDwqbuykNt64oZWc4XKCa3mprEGC3IbJTBFqglXmZ7l9ywG
EEUJYOlb2XrSuPWml39beWdKM8kzr1OjnlOm6+lpTRCBfo0wa9F8YZRhHPAkwKkX
XDeOGpWRj4ohOx0d2GWkyV5xyN14p2tQOCdOODmz80yUTgRpPVQUtOEhXQARAQAB
tCFBV1MgQ0xJIFRlYW0gPGF3cy1jbGlAYW1hem9uLmNvbT6JAlQEEwEIAD4CGwMF
CwkIBwIGFQoJCAsCBBYCAwECHgECF4AWIQT7Xbd/1cEYuAURraimMQrMRnJHXAUC
aGveYQUJDMpiLAAKCRCmMQrMRnJHXKBYD/9Ab0qQdGiO5hObchG8xh8Rpb4Mjyf6
0JrVo6m8GNjNj6BHkSc8fuTQJ/FaEhaQxj3pjZ3GXPrXjIIVChmICLlFuRXYzrXc
Pw0lniybypsZEVai5kO0tCNBCCFuMN9RsmmRG8mf71C4FSTbUDmxG/QlYK+0IV/l
uJkzxWa+rySkdpm0JdqumjegNRgObdXHAQDWlubWQHWyZyIQ2B4U7AxqSpcdJp6I
S4Zds4wVLd1WE5pquYQ8vS2cNlDm4QNg8wTj58e3lKN47hXHMIb6CHxRnb947oJa
pg189LLPR5koh+EorNkA1wu5mAJtJvy5YMsppy2y/kIjp3lyY6AmPT1posgGk70Z
CmToEZ5rbd7AREXtlh76A0cabMDFlEHDIK8RNUOSRr7L64+KxOUegKBfQHb9dADY
qqiKqpCbKgvtWlds909Ms74JBgr2KwZCSY1HaOxnIr4CY43QRqAq5YHOay/mU+6w
hhmdF18vpyK0vfkvvGresWtSXbag7Hkt3XjaEw76BzxQH21EBDqU8WJVjHgU6ru+
DJTs+SxgJbaT3hb/vyjlw0lK+hFfhWKRwgOXH8vqducF95NRSUxtS4fpqxWVaw3Q
V2OWSjbne99A5EPEySzryFTKbMGwaTlAwMCwYevt4YT6eb7NmFhTx0Fis4TalUs+
j+c7Kg92pDx2uQ==
=OBAt
-----END PGP PUBLIC KEY BLOCK-----
```

以下公開金鑰詳細資訊僅供參考。

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

c. 使用下列命令匯入 AWS CLI 公有金鑰，以您建立之公有金鑰的檔案名稱取代 *public-key-file-name*。

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

d. 下載您下載之套件的 AWS CLI 簽章檔案。它會有與其對應的 `.zip` 檔案具有相同的路徑和名稱，但具有副檔名 `.sig`。在下列範例中，我們將它作為名稱 `awscliv2.sig` 的檔案儲存在目前的資料夾。

Linux x86 (64-bit)

對於最新版本的 AWS CLI，請使用下列命令區塊：

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

對於特定版本的 AWS CLI，請將連字號和版本號碼附加至檔案名稱。在這個範例中，版本 *2.0.30* 的檔案名稱為 `awscli-exe-linux-x86_64-2.0.30.zip.sig`，因此使用以下命令：

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

如需版本清單，請參閱 GitHub 上的 AWS CLI 第 2 版變更日誌。

Linux ARM

對於最新版本的 AWS CLI，請使用下列命令區塊：

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

對於特定版本的 AWS CLI，請將連字號和版本號碼附加至檔案名稱。在這個範例中，版本 *2.0.30* 的檔案名稱為 awscli-exe-linux-aarch64-2.0.30.zip.sig，因此使用以下命令：

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

如需版本清單，請參閱 GitHub 上的 AWS CLI 第 2 版變更日誌。

e. 驗證簽章，並將下載的 .sig 和 .zip 檔案名稱以參數形式傳遞給 gpg 命令。

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

輸出應看起來如下列內容。

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

> **Important**
>
> 輸出中的警告是預期的，並不表示有問題。這是因為您的個人 PGP 金鑰（如果有的話）與 AWS CLI PGP 金鑰之間沒有信任鏈。如需詳細資訊，請參閱「信任網路」。

3. 解壓縮安裝程式。如果您的 Linux 發行版沒有內建 unzip 命令，請使用類似命令解壓縮。下列範例命令會解壓縮套件，並在名為 aws 的目前目錄下建立目錄。

```
$ unzip awscliv2.zip
```

4. 執行安裝程式。安裝命令會使用新解壓縮的 aws 目錄中名為 install 的檔案。依預設，檔案會全部安裝到 /usr/local/aws-cli，並在 /usr/local/bin 中建立符號連結。命令包括 sudo，以授予寫入權限給這些資料夾。

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

如果指定您已具有寫入權限的資料夾，則可以在不使用 sudo 的情況下進行安裝。使用下列 install 命令指示來指定安裝位置：

- 請確定您提供給 -i 和 -b 參數的路徑不包含任何包含空格字元或其他空格字元的磁碟區名稱或目錄名稱。如果有空格，則安裝會失敗。
- --install-dir 或 -i – 此選項指定要將所有檔案複製到的目錄。

  預設值為 /usr/local/aws-cli。
- --bin-dir 或 -b – 此選項會指定安裝目錄中的 aws 主程式以符號方式連結至指定路徑的檔案 aws。您必須具有指定目錄的寫入許可。建立路徑中已存在目錄的符號連結，可讓您不需將安裝目錄新增到使用者的 $PATH 變數。

  預設值為 /usr/local/bin。

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

> **Note**
>
> 若要將 AWS CLI 第 2 版的目前安裝更新至較新版本，請新增現有的符號連結和安裝程式資訊，以使用 --update 參數建構 install命令。
>
> ```
> $ sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/aws-cli --update
> ```
>
> 若要找到現有的符號連結和安裝目錄，請執行以下步驟：
>
> 1. 使用 which 命令來查找您的符號連結。這提供您與 --bin-dir 參數一起使用的路徑。
>
>    ```
>    $ which aws
>    /usr/local/bin/aws
>    ```

2. 使用 `ls` 命令來查找符號連結指向的目錄。這提供您與 `--install-dir` 參數一起使用的路徑。

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

5. 請使用以下命令來確認安裝。

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

如果找不到 `aws` 命令，您可能需要重新啟動終端機或遵循 故障診斷錯誤 中的故障診斷。

macOS

安裝要求

- 您知道要安裝的 AWS CLI 版本 2。如需版本清單，請參閱 GitHub 上的 AWS CLI 第 2 版變更日誌。
- 我們支援 Apple 支援的 AWS CLI 64 位元 macOS 版本 2。
- 由於 AWS 不會維護第三方儲存庫，因此我們無法保證它們包含最新版本的 AWS CLI。

安裝說明

您可以透過下列方式在 macOS 上安裝 第 2 AWS CLI 版。

GUI installer

以下步驟說明如何使用標準 macOS 使用者界面和瀏覽器來安裝或更新至 AWS CLI 第 2 版的最新版本。如果您要更新至最新版本，請使用您目前版本所使用的相同安裝方法。

1. 在您的瀏覽器中，將連字號和版本號碼附加至檔案名稱， AWS CLI 以下載特定版本的 。

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

在這個範例中，版本 *2.0.30* 的檔案名稱為 `AWSCLIV2-2.0.30.pkg`，因此使用以下連結：https://awscli.amazonaws.com/AWSCLIV2-2.0.30.pkg。

2. 請依照畫面上的指示，執行您下載的檔案。您可以選擇以下列方式安裝第 2 AWS CLI 版：

   - 對於電腦上的所有使用者 (需要 **sudo**)
     - 您可以安裝到任何資料夾，或選擇建議的 /usr/local/aws-cli 的預設資料夾。
     - 安裝程式會自動在 /usr/local/bin/aws 建立符號連結，其連結至您所選擇的安裝資料夾中的主程式。
   - 只對於目前的使用者 (不需要 **sudo**)
     - 您可以安裝到任何具有寫入權限的資料夾。
     - 由於標準的使用者許可，在安裝程式完成之後，您的 $PATH 必須在命令提示字元中使用下列命令，手動建立指向 aws 和 aws_completer 程式的符號連結檔案。如果您的 $PATH 包含可以寫入的資料夾，則如果將該資料夾指定為目標路徑，則可執行下列命令而無需 sudo。如果您在 $PATH 中沒有可寫入的資料夾，則必須在命令中使用 sudo，以取得寫入指定目標資料夾的許可。符號連結的預設位置為 /usr/local/bin/。

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

   > **Note**
   >
   > 您可以在安裝程式中的任何位置按 Cmd+L 來檢視安裝的偵錯日誌。這會開啟一個日誌窗格，使您能夠篩選和儲存日誌。日誌檔也會自動儲存到 /var/log/install.log。

3. 若要確認 Shell 可以在您的 aws 中尋找並執行 $PATH 命令，請使用下列命令。

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

   如果找不到 aws 命令，您可能需要重新啟動終端機或遵循 故障診斷錯誤 中的故障診斷。

Command line installer - All users

如果您有sudo許可，您可以為電腦上的所有使用者安裝 第 2 AWS CLI 版。我們提供一組易於複製和貼上的步驟。請參閱後續步驟中每一行的描述。

對於特定版本的 AWS CLI，請將連字號和版本號碼附加至檔案名稱。在這個範例中，版本 *2.0.30* 的檔案名稱為 AWSCLIV2-2.0.30.pkg，因此使用以下命令：

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

1. 使用 curl 命令下載檔案。在此範例中，-o 選項會指定系統寫入下載套件的檔案名稱。在前述的範例中，檔案會寫入目前的資料夾中的 AWSCLIV2.pkg。

   對於特定版本的 AWS CLI，請將連字號和版本號碼附加至檔案名稱。在這個範例中，版本 *2.0.30* 的檔案名稱為 AWSCLIV2-2.0.30.pkg，因此使用以下命令：

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

   如需版本清單，請參閱 GitHub 上的 AWS CLI 第 2 版變更日誌。

2. 執行標準 macOS installer 程式，將下載的 .pkg 檔案指定為來源。使用 -pkg 參數可指定要安裝的套件名稱，以及磁碟機要安裝套件的 -target / 參數。檔案會安裝到 /usr/local/aws-cli，並且會在 /usr/local/bin 中自動建立符號連結。您必須在命令上包含 sudo，才能將寫入權限授予這些資料夾。

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

   安裝完成後，偵錯記錄檔會寫入 /var/log/install.log。

3. 若要確認 Shell 可以在您的 aws 中尋找並執行 $PATH 命令，請使用下列命令。

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

   如果找不到 aws 命令，您可能需要重新啟動終端機或遵循 故障診斷錯誤 中的故障診斷。

Command line - Current user

1. 若要指定要安裝 AWS CLI 的資料夾，您必須建立 XML 檔案。這個檔案是 XML 格式化的檔案，看起來如下列範例。保留顯示的所有值，但您必須將第 9 行中的路徑 */Users/myusername* 取代為要安裝第 2 AWS CLI 版之資料夾的路徑。資料夾必須已存在，否則命令會失敗。此 XML 範例指定安裝程式在資料夾 AWS CLI 中安裝 /Users/myusername，並在其中建立名為 的資料夾aws-cli。

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

2. 使用 curl 命令下載 pkg 安裝程式。在此範例中，-o 選項會指定系統寫入下載套件的檔案名稱。在前述的範例中，檔案會寫入目前的資料夾中的 AWSCLIV2.pkg。

   對於特定版本的 AWS CLI，請將連字號和版本號碼附加至檔案名稱。在這個範例中，版本 *2.0.30* 的檔案名稱為 AWSCLIV2-2.0.30.pkg，因此使用以下命令：

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

   如需版本清單，請參閱 GitHub 上的 AWS CLI 第 2 版變更日誌。

3. 現在您可以使用下列選項來執行標準 macOS installer 程式：

   - 使用參數 -pkg 指定要安裝的套件的名稱。
   - 若要指定僅限目前使用者的安裝，您必須設定參數 -target 為 CurrentUserHomeDirectory。
   - 指定您在 -applyChoiceChangesXML 參數中建立的 XML 檔案的路徑 (相對於目前的資料夾) 和名稱。

下列範例會在 資料夾 AWS CLI 中安裝 /Users/myusername/aws-cli。

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

4. 由於標準使用者許可通常不允許寫入 $PATH 中的資料夾，因此此模式下的安裝程式不會嘗試新增連至 aws 和 aws_completer 的符號連結。若要 AWS CLI 讓 正確執行，您必須在安裝程式完成後手動建立符號連結。如果您的 $PATH 包含可以寫入的資料夾，則如果將該資料夾指定為目標路徑，則可執行下列命令而無需 sudo。如果您的 $PATH 中沒有可寫入的資料夾，則必須使 sudo 以取得寫入指定目標資料夾的許可。符號連結的預設位置為 /usr/local/bin/。

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

安裝完成後，偵錯記錄檔會寫入 /var/log/install.log。

5. 若要確認 Shell 可以在您的 aws 中尋找並執行 $PATH 命令，請使用下列命令。

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

如果找不到 aws 命令，您可能需要重新啟動終端機或遵循 故障診斷錯誤 中的故障診斷。

Windows

安裝要求

- 您知道要安裝的 AWS CLI 版本 2。如需版本清單，請參閱 GitHub 上的 AWS CLI 第 2 版變更日誌。
- 我們在 Microsoft 支援的 64 位元 Windows 版本 AWS CLI 上支援 。
- 安裝軟體的管理員權限

安裝說明

若要更新目前在 Windows 上安裝的第 2 AWS CLI 版，請在每次更新時下載新的安裝程式以覆寫先前的版本。 AWS CLI 會定期更新。若要檢視最新版本發佈的時間，請參閱 GitHub 上的 AWS CLI 第 2 版變更日誌。

1. 以下列其中一種方式下載並執行適用於 Windows 的 AWS CLI MSI 安裝程式 (64 位元）：

   - 下載並執行 MSI 安裝程式：若要為特定版本的 建立下載連結 AWS CLI，請將連字號和版本編號附加至檔案名稱。

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

     在這個範例中，版本 *2.0.30* 的檔案名稱為 `AWSCLIV2-2.0.30.msi`，因此使用以下連結：https://awscli.amazonaws.com/AWSCLIV2-2.0.30.msi。

   - 使用 Msiexec 命令：或者，您也可以使用 MSI 安裝程式，方法是將連結新增至 `msiexec` 命令。對於特定版本的 AWS CLI，請將連字號和版本號碼附加至檔案名稱。

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

     在這個範例中，版本 *2.0.30* 的檔案名稱為 `AWSCLIV2-2.0.30.msi`，因此使用以下連結：https://awscli.amazonaws.com/AWSCLIV2-2.0.30.msi。

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

     如需可與 `msiexec` 一起使用的各種參數，請參閱 Microsoft Docs 網站上的 msiexec。

   如需版本清單，請參閱 GitHub 上的 AWS CLI 第 2 版變更日誌。

2. 若要確認安裝，請開啟開始功能表、搜尋 cmd 以開啟命令提示字元視窗，然後在命令提示字元中使用 `aws --version` 命令。

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

   如果 Windows 找不到程式，您可能需要關閉命令提示字元再重新開啟，以重新整理路徑，或遵循故障診斷錯誤 中的故障診斷。

針對 AWS CLI 安裝和解除安裝錯誤進行故障診斷

如果您在安裝或解除安裝 之後遇到問題 AWS CLI，請參閱 故障診斷錯誤 以取得故障診斷步驟。如需最相關的故障診斷步驟，請參閱 the section called “命令未找到錯誤”、the section called “「aws --version」命令傳回的版本與您安裝的版本不同” 和 the section called “解除安裝 後，「aws --version」命令會傳回版本 AWS CLI”。

後續步驟

完成 the section called “先決條件” 中的步驟並安裝 AWS CLI後，則應執行 the section called “設定”。

AWS CLI 從來源建置和安裝

本主題將描述如何在支援的作業系統上從來源安裝或更新 AWS Command Line Interface (AWS CLI)。

如需最新版本 的資訊 AWS CLI，請參閱 GitHub 上的AWS CLI 第 2 版變更日誌。

> **Important**
>
> AWS CLI 第 1 版和第 2 版使用相同的aws命令名稱。如果您先前已安裝 AWS CLI 版本 1，請參閱 第 2 AWS CLI 版的遷移指南。

主題

- 為什麼要從來源建置？
- 快速步驟
- 步驟 1：設定所有要求
- 步驟 2：設定 AWS CLI 來源安裝
- 步驟 3：建置 AWS CLI
- 步驟 4：安裝 AWS CLI
- 步驟 5：驗證 AWS CLI 安裝
- 工作流程範例
- 針對 AWS CLI 安裝和解除安裝錯誤進行故障診斷
- 後續步驟

為什麼要從來源建置？

AWS CLI 適用於大多數平台和環境的預先建置安裝程式，以及 Docker 映像。

這些安裝程式通常涵蓋大多數的使用案例。從來源安裝的說明是為了幫助我們的安裝程式未涵蓋的使用案例。部分使用案例如下：

- 預先建置的安裝程式不支援您的環境。例如，預先建置的安裝程式不支援 ARM 32 位元。
- 預先建置的安裝程式具有您環境缺少的相依性。例如，Alpine Linux 使用 `musl`，但是現有的安裝程式需要 `glibc`，造成預先建置的安裝程式無法立即運作。
- 預先建置的安裝程式需要您的環境限制存取的資源。例如，強化安全性的系統可能不會授予共用記憶體的權限。這是凍結的 `aws` 安裝程式所需的。
- 預先建置的安裝程式通常是套件管理工具中維護程式的封鎖程式，通常建議程式碼和套件在建置過程中取得完全控制權限。從來源建置讓分發維護程式能以更簡化的流程持續更新 AWS CLI 。啟用維護程式可在客戶從 `brew`、`yum` 和 `apt` 等第三方套件管理工具安裝時，提供客戶更新的 AWS CLI 版本。
- 修補 AWS CLI 功能的客戶需要從來源建置和安裝 AWS CLI 。這對想要在對 AWS CLI GitHub 儲存庫做出變更之前測試對來源所做的變更的社群成員來說特別重要。

快速步驟

> **Note**
>
> 所有程式碼範例都是假設從來源目錄的根目錄執行。

若要 AWS CLI 從來源建置和安裝 ，請遵循本節中的步驟。 AWS CLI 利用 GNU 自動工具從來源安裝。在最簡單的情況下， AWS CLI 您可以從來源安裝 ，方法是從 AWS CLI GitHub 儲存庫的根目錄執行預設範例命令。

1. 為您的環境設定所有需求。這包括能執行 GNU 自動工具產生的檔案和安裝 Python 3.8 或更新版本。
2. 在終端機中，導覽至 AWS CLI 來源資料夾的最上層，然後執行 `./configure`命令。此命令會檢查系統中是否有所有必要的相依性，並根據偵測到和指定的組態產生 `Makefile` 用於建置和安裝 AWS CLI 。

Linux and macOS

下列./configure命令範例 AWS CLI 會使用預設設定來設定 的建置組態。

```
$ ./configure
```

Windows PowerShell

執行任何呼叫 MSYS2 的命令之前，您必須保留現有的工作目錄：

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

然後使用下列./configure命令範例， AWS CLI 使用 Python 可執行檔的本機路徑設定 的建置組態、安裝至 C：\Program Files\AWSCLI，以及下載所有相依性。

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

如需詳細資訊、可用的組態選項和預設設定資訊，請參閱 the section called “步驟 2：設定 AWS CLI 來源安裝 ” 一節。

3. 執行 make 命令。此命令 AWS CLI 會根據您的組態設定建置 。

   下列 make 命令範例會使用您現有的 ./configure 設定，以預設選項建置。

   Linux and macOS

   ```
   $ make
   ```

   Windows PowerShell

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

   如需詳細資訊和可用的建置選項，請參閱 the section called “步驟 3：建置 AWS CLI” 一節。

4. 執行 make install 命令。此命令會將 建置的 安裝 AWS CLI 到系統上設定的位置。

下列`make install`命令範例會安裝您建置的 ， AWS CLI 並使用預設命令設定在設定的位置中建立符號連結。

Linux and macOS

```
$ make install
```

Windows PowerShell

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

安裝之後， AWS CLI 請使用下列方式將路徑新增至 ：

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

如需詳細資訊和可用的安裝選項，請參閱 the section called “步驟 4：安裝 AWS CLI” 一節。

5. 使用以下命令確認 AWS CLI 已成功安裝 ：

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

如需安裝錯誤的故障診斷步驟，請參閱 the section called “針對 AWS CLI 安裝和解除安裝錯誤進行故障診斷” 一節。

步驟 1：設定所有要求

若要 AWS CLI 從來源建置 ，您需要事先完成下列項目：

> **Note**
>
> 所有程式碼範例都是假設從來源目錄的根目錄執行。

1. 透過 AWS CLI 偽造 AWS CLI GitHub 儲存庫或下載來源 tarball 來下載來源。說明如下：

- 從 GitHub 延伸和複製 AWS CLI 儲存庫。如需更多詳細資訊，請參閱 GitHub 文件中的《延伸儲存庫》。
- 在 下載最新的來源 tarball，使用以下命令https://awscli.amazonaws.com/awscli.tar.gz擷取內容：

```
$ curl -o "https://awscli.amazonaws.com/awscli.tar.gz" -o "awscliv2.zip"
$ tar -xzf awscli.tar.gz
```

> **Note**
>
> 要下載特定版本，請使用以下連結格式：https://awscli.amazonaws.com/awscli-*versionnumber*.tar.gz
> 例如，如需版本 2.10.0，則連結將如下：https://awscli.amazonaws.com/awscli-*2.10.0*.tar.gz
> 來源版本從 AWS CLI的 2.10.0 版開始提供。

(可選) 完成下列步驟，驗證所下載 zip 檔案的完整性：

1. 您可以依循下列步驟使用 GnuPG 工具來驗證簽章。

   AWS CLI 安裝程式套件.zip檔案使用 PGP 簽章進行密碼編譯簽署。如果檔案有任何損壞或更改，此驗證會失敗，且您不應該繼續進行安裝。

2. 使用您的套件管理工具下載並安裝 gpg 命令。如需有關 GnuPG 的詳細資訊，請參閱 GnuPG 網站。
3. 若要建立公開金鑰檔案，請建立文字檔案，然後貼上下列文字。

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

mQINBF2Cr7UBEADJZHcgusOJl7ENSyumXh85z0TRV0xJorM2B/JL0kHOyigQluUG
ZMLhENaG0bYatdrKP+3H91lvK050pXwnO/R7fB/FSTouki4ciIx5OuLlnJZIxSzx
PqGl0mkxImLNbGWoi6Lto0LYxqHN2iQtzlwTVmq9733zd3XfcXrZ3+LblHAgEt5G
TfNxEKJ8soPLyWmwDH6HWCnjZ/aIQRBTIQ05uVeEoYxSh6wOai7ss/KveoSNBbYz
gbdzoqI2Y8cgH2nbfgp3DSasaLZEdCSsIsK1u05CinE7k2qZ7KgKAUIcT/cR/grk
C6VwsnDU0OUCideXcQ8WeHutqvgZH1JgKDbznoIzeQHJD238GEu+eKhRHcz8/jeG
94zkcgJOz3KbZGYMiTh277Fvj9zzvZsbMBCedV1BTg3TqgvdX4bdkhf5cH+7NtWO
lrFj6UwAsGukBTAOxC0l/dnSmZhJ7Z1KmEWilro/gOrjtOxqRQutlIqG22TaqoPG
fYVN+en3Zwbt97kcgZDwqbuykNt64oZWc4XKCa3mprEGC3IbJTBFqglXmZ7l9ywG
```

```
EEUJYOlb2XrSuPWml39beWdKM8kzr1OjnlOm6+lpTRCBfo0wa9F8YZRhHPAkwKkX
XDeOGpWRj4ohOx0d2GWkyV5xyN14p2tQOCdOODmz80yUTgRpPVQUtOEhXQARAQAB
tCFBV1MgQ0xJIFRlYW0gPGF3cy1jbGlAYW1hem9uLmNvbT6JAlQEEwEIAD4CGwMF
CwkIBwIGFQoJCAsCBBYCAwECHgECF4AWIQT7Xbd/1cEYuAURraimMQrMRnJHXAUC
aGveYQUJDMpiLAAKCRCmMQrMRnJHXKBYD/9Ab0qQdGiO5hObchG8xh8Rpb4Mjyf6
0JrVo6m8GNjNj6BHkSc8fuTQJ/FaEhaQxj3pjZ3GXPrXjIIVChmICLlFuRXYzrXc
Pw0lniybypsZEVai5kO0tCNBCCFuMN9RsmmRG8mf7lC4FSTbUDmxG/QlYK+0IV/l
uJkzxWa+rySkdpm0JdqumjegNRgObdXHAQDWlubWQHWyZyIQ2B4U7AxqSpcdJp6I
S4Zds4wVLd1WE5pquYQ8vS2cNlDm4QNg8wTj58e3lKN47hXHMIb6CHxRnb947oJa
pg189LLPR5koh+EorNkA1wu5mAJtJvy5YMsppy2y/kIjp3lyY6AmPT1posgGk70Z
CmToEZ5rbd7ARExtlh76A0cabMDFlEHDIK8RNUOSRr7L64+KxOUegKBfQHb9dADY
qqiKqpCbKgvtWlds909Ms74JBgr2KwZCSY1HaOxnIr4CY43QRqAq5YHOay/mU+6w
hhmdF18vpyK0vfkvvGresWtSXbag7Hkt3XjaEw76BzxQH21EBDqU8WJVjHgU6ru+
DJTs+SxgJbaT3hb/vyjlw0lK+hFfhWKRwgOXH8vqducF95NRSUxtS4fpqxWVaw3Q
V2OWSjbne99A5EPEySzryFTKbMGwaTlAwMCwYevt4YT6eb7NmFhTx0Fis4TalUs+
j+c7Kg92pDx2uQ==
=OBAt
-----END PGP PUBLIC KEY BLOCK-----
```

以下公開金鑰詳細資訊僅供參考。

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

4. 使用下列命令匯入 AWS CLI 公有金鑰，以您建立之公有金鑰的檔案名稱取代 *public-key-file-name*。

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

5. 下載您在 https：//https://awscli.amazonaws.com/awscli.tar.gz.sig 下載之套件的 AWS CLI 簽章檔案。它會有與其對應的 tarball 檔案具有相同的路徑和名稱，但具有副檔名 `.sig`。將它儲存在與 tarball 檔案相同的路徑下。或使用下列命令區塊：

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

6. 驗證簽章，並將下載的 .sig 和 .zip 檔案名稱以參數形式傳遞給 gpg 命令。

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

輸出應看起來如下列內容。

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

> **Important**
>
> 輸出中的警告是預期的，並不表示有問題。這是因為您的個人 PGP 金鑰（如果您有）和 AWS CLI PGP 金鑰之間沒有信任鏈。如需詳細資訊，請參閱「信任網路」。

2. 您有一個環境可以執行 GNU 自動工具 產生的檔案，例如 configure 和 Makefile。這些檔案在 POSIX 平台上廣泛可攜。

   Linux and macOS

   如果您的環境尚未安裝 Autotools，或者您需要更新它們，請遵循 GNU 文件中我如何安裝 Autotools (以使用者身分)？或基本安裝中的安裝說明。

   Windows PowerShell

   > **Warning**
   >
   > 如果您處於 Windows 環境中，建議您使用預先建置的安裝程式。如需預先建置安裝程式的安裝說明，請參閱 the section called “安裝/更新”

由於 Windows 未隨附 POSIX 相容的 Shell，因此您需要安裝其他軟體，才能 AWS CLI 從來源安裝 。MSYS2 提供了一系列工具和程式庫來幫助建置和安裝 Windows 軟體，尤其是對於 Autotools 使用的 POSIX 架構指令碼。

1. 安裝 MSYS2。如需有關安裝和使用 MSYS2 的資訊，請參閱 MSYS2 文件中的安裝和使用說明。
2. 使用以下命令開啟 MSYS2 終端並安裝 Autotools。

```
$ pacman -S autotools
```

> **Note**
>
> 使用本指南中適用於 Windows 的設定、建置和安裝程式碼範例時，會假設使用預設的 MSYS2 安裝路徑 `C:\msys64\usr\bin\bash`。在 PowerShell 中呼叫 MSYS2 時，您將使用以下格式，並使用引號中的 bash 命令：
>
> ```
> PS C:\> C:\msys64\usr\bin\bash  -lc "command example"
> ```
>
> 下列命令範例會呼叫 `./configure` 命令。
>
> ```
> PS C:\> C:\msys64\usr\bin\bash  -lc "./configure"
> ```

3. 已安裝 Python 3.8 或更新版本的解譯器。所需的最低 Python 版本遵循與官方 Python 支援政策 AWS SDKs和工具相同的時間表。解譯器僅支援到終止支援日期之後的 6 個月。
4. (選擇性) 安裝 AWS CLI所有的建置和執行時間 Python 程式庫相依性。如果缺少任何相依性以及不知道如何安裝，`./configure` 命令能提供您資訊。

   您可以透過配置自動安裝和使用這些相依性，請參閱 the section called “下載相依性” 獲得更多資訊。

步驟 2：設定 AWS CLI 來源安裝

建置和安裝 的組態 AWS CLI 會使用 `configure`指令碼指定。如需所有組態選項的文件，請使用 `--help` 選項執行 `configure` 指令碼：

Linux and macOS

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

Windows PowerShell

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

最重要的選項如下：

- 安裝位置
- Python 解譯器
- 下載相依性
- 安裝類型

安裝位置

的來源安裝 AWS CLI 使用兩個可設定的目錄來安裝 AWS CLI：

- `libdir` - AWS CLI 將安裝 的父目錄。 AWS CLI 安裝的路徑為 *<libdir-value>*/aws-cli。Linux 和 macOS 的預設 `libdir` 值是 `/usr/local/lib`，讓預設安裝目錄成為 */usr/local/lib*/aws-cli
- `bindir` - AWS CLI 安裝可執行檔的目錄。預設位置為 `/usr/local/bin`。

下列 `configure` 選項可控制使用的目錄：

- `--prefix` - 設定用於安裝的目錄字首。Linux 和 macOS 的預設值為 `/usr/local`。
- `--libdir` - 設定 `libdir` 用於安裝 AWS CLI。預設值為 <prefix-value>/lib。如果 `--libdir` 和 `--prefix` 兩者都未指定，Linux 和 macOS 的預設值為 `/usr/local/lib/`。
- `--bindir` - 設定bindir用於安裝 AWS CLI aws和 `aws_completer` 可執行檔的 。預設值為 <prefix-value>/bin。如果 `bindir` 和 `--prefix` 兩者都未指定，Linux 和 macOS 的預設值為 `/usr/local/bin/`。

Linux and macOS

下列命令範例使用 --prefix 選項來執行 AWS CLI的本機使用者安裝。此命令會在 AWS CLI 中安裝 ，$HOME/.local/lib/aws-cli並在 中安裝 可執行檔$HOME/.local/bin：

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

下列命令範例會使用 --libdir 選項，將 AWS CLI 作為附加元件應用程式安裝在 /opt 目錄中。此命令會在 AWS CLI 上安裝 ，/opt/aws-cli並在其預設位置安裝可執行檔/usr/local/bin。

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

Windows PowerShell

下列命令範例使用 --prefix 選項來執行 AWS CLI的本機使用者安裝。此命令會在 AWS CLI 中安裝 ，$HOME/.local/lib/aws-cli並在 中安裝 可執行檔$HOME/.local/bin：

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

下列命令範例會使用 --libdir 選項，將 AWS CLI 作為附加元件應用程式安裝在 /opt 目錄中。此命令 AWS CLI 會在 上安裝 C:\Program Files\AWSCLI\opt\aws-cli。

Python 解譯器

> **Note**
>
> 強烈建議您在 Windows 安裝時，指定 Python 解譯器。

./configure 指令碼會自動選取已安裝的 Python 3.8 或更新版本解譯器，以便在 AWS CLI 使用 AM_PATH_PYTHON Autoconf 巨集建置和執行 時使用。

執行 configure 指令碼時，可以使用 PYTHON 環境變數明確設定要使用的 Python 解釋器：

Linux and macOS

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

Windows PowerShell

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

下載相依性

根據預設，系統 AWS CLI 必須已安裝 的所有建置和執行時間相依性。這包括任何 Python 程式庫的相依性。configure 指令碼執行時會檢查所有相依性，如果系統缺少任何 Python 相依性，configure 指令碼會發生錯誤。

您的系統缺少相依性時，會顯示以下的代碼範例錯誤：

Linux and macOS

```
$ ./configure
checking for a Python interpreter with version >= 3.8... python
checking for python... /Users/username/.envs/env3.11/bin/python
checking for python version... 3.11
checking for python platform... darwin
checking for GNU default python prefix... ${prefix}
checking for GNU default python exec_prefix... ${exec_prefix}
checking for python script directory (pythondir)... ${PYTHON_PREFIX}/lib/python3.11/
site-packages
checking for python extension module directory (pyexecdir)... ${PYTHON_EXEC_PREFIX}/
lib/python3.11/site-packages
checking for --with-install-type... system-sandbox
checking for --with-download-deps... Traceback (most recent call last):
  File "<frozen runpy>", line 198, in _run_module_as_main
  File "<frozen runpy>", line 88, in _run_code
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
 125, in <module>
    main()
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
 121, in main
    parsed_args.func(parsed_args)
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
 49, in validate
    validate_env(parsed_args.artifact)
  File "/Users/username/aws-code/aws-cli/./backends/build_system/validate_env.py",
 line 68, in validate_env
    raise UnmetDependenciesException(unmet_deps, in_venv)
```

```
validate_env.UnmetDependenciesException: Environment requires following Python
 dependencies:

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

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

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

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

Windows PowerShell

```
PS C:\> C:\msys64\usr\bin\bash -lc "./configure"
checking for a Python interpreter with version >= 3.8... python
checking for python... /Users/username/.envs/env3.11/bin/python
checking for python version... 3.11
checking for python platform... darwin
checking for GNU default python prefix... ${prefix}
checking for GNU default python exec_prefix... ${exec_prefix}
checking for python script directory (pythondir)... ${PYTHON_PREFIX}/lib/python3.11/
site-packages
checking for python extension module directory (pyexecdir)... ${PYTHON_EXEC_PREFIX}/
lib/python3.11/site-packages
checking for --with-install-type... system-sandbox
checking for --with-download-deps... Traceback (most recent call last):
  File "<frozen runpy>", line 198, in _run_module_as_main
  File "<frozen runpy>", line 88, in _run_code
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
 125, in <module>
    main()
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
 121, in main
    parsed_args.func(parsed_args)
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
 49, in validate
    validate_env(parsed_args.artifact)
  File "/Users/username/aws-code/aws-cli/./backends/build_system/validate_env.py",
 line 68, in validate_env
```

```
    raise UnmetDependenciesException(unmet_deps, in_venv)
validate_env.UnmetDependenciesException: Environment requires following Python
 dependencies:

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

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

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

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

若要自動安裝所需的 Python 相依性，請使用 `--with-download-deps` 選項。使用此標記時，建置程序會執行下列動作：

- 跳過 Python 程式庫的相依性檢查。
- 配置設定以下載所有必要的 Python 相依性，並在 `make` 建置時僅使用下載的相依性來建置 AWS CLI 。

下面的配置命令範例使用 `--with-download-deps` 選項下載和使用 Python 相依性：

Linux and macOS

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

Windows PowerShell

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

安裝類型

來源安裝程序支援下列安裝類型：

- `system-sandbox` - （預設）建立隔離的 Python 虛擬環境，將 安裝 AWS CLI 到虛擬環境中，並在虛擬環境中將 符號連結安裝到 `aws`和 `aws_completer` 可執行檔。此 的安裝直接 AWS CLI 取決於所選 Python 解譯器的執行時間。

  這是一種輕量型安裝機制，可在系統 AWS CLI 上安裝 ，並透過在虛擬環境中對安裝進行沙盒處理來遵循 Python 最佳實務。此安裝適用於希望在安裝 Python 時，以最摩擦的方式 AWS CLI 從來源安裝的客戶。

- `portable-exe` - 將 凍結 AWS CLI 為可分佈至類似架構環境的獨立可執行檔。這與用於產生 AWS CLI官方預先建置可執行檔的過程相同。`portable-exe` 凍結在 `configure` 步驟中選擇的 Python 解譯器副本，以用於 AWS CLI的執行時間。這讓它可移動到可能沒有 Python 解譯器的其他機器。

  這種類型的構建很有用，因為您可以確保您的 AWS CLI 安裝不會與環境安裝的 Python 版本耦合，並且可以將建構分發到可能尚未安裝 Python 的其他系統。這可讓您控制所使用 AWS CLI 可執行檔的相依性和安全性。

若要設定安裝類型，請使用 `--with-install-type` 選項並指定 `portable-exe` 或 `system-sandbox` 的值。

以下 `./configure` 命令範例會指定 `portable-exe` 的值：

Linux and macOS

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

Windows PowerShell

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

步驟 3：建置 AWS CLI

使用 `make`命令， AWS CLI 使用您的組態設定建置 ：

Linux and macOS

```
$ make
```

Windows PowerShell

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

> **Note**
>
> 使用 **make** 指令時，會在幕後完成下列步驟：
>
> 1. 虛擬環境使用 Python venv 模組在建置目錄中建立。虛擬環境使用 Python 標準庫中提供的 pip 版本啟動。
> 2. 複製 Python 程式庫的相依性。根據是否在 configure 命令中指定 --with-download-deps 標記，此步驟會執行下列其中一項作業：
>   - 指定 --with-download-deps 。Python 的相依性使用 pip 安裝。這包括 wheel、setuptools 以及所有 AWS CLI 執行時間相依性。如果您在建置 portable-exe，則 pyinstaller 已安裝。這些要求都在從 pip-compile 中產生的鎖定檔案中指定。
>   - 未指定 --with-download-deps。Python 解譯器站點套件中的 Python 庫以及任何指令碼 (例如 pyinstaller) 都將複製到用於建置的虛擬環境中。
> 3. pip install 直接在 AWS CLI 程式碼庫上執行 ，以執行離線的樹狀內建置，並將 安裝在 AWS CLI 建置虛擬環境中。此安裝使用 pip 標記 --no-build-isolation、--use-feature=in-tree-build、--no-cache-dir 和 --no-index。
> 4. (選擇性) 如果在 configure 命令中將 --install-type 設為 portable-exe，請使用 pyinstaller 建置獨立的可執行檔。

步驟 4：安裝 AWS CLI

make install 命令會將您建置的 安裝 AWS CLI 到系統上設定的位置。

Linux and macOS

下列命令範例 AWS CLI 使用您的組態和建置設定安裝 ：

```
$ make install
```

Windows PowerShell

下列命令範例 AWS CLI 會使用您的組態和建置設定安裝 ，然後新增具有 路徑的環境變數 AWS CLI：

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

make install 規則支援 DESTDIR 變數。如果有指定，此變數會在安裝 AWS CLI時，將指定路徑的加到已設定的安裝路徑前面。預設不會為此變數設定任何值。

Linux and macOS

下列程式碼範例會使用 --prefix=/usr/local 標記來設定安裝位置，然後使用 make install 指令和 DESTDIR=*/tmp/stage* 來變更目的地。這些命令會導致 AWS CLI 安裝在 */tmp/stage/*usr/local/lib/aws-cli及其位於 的可執行檔*/tmp/stage/*usr/local/bin。

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

Windows PowerShell

下列程式碼範例會使用 --prefix=\awscli 標記來設定安裝位置，然後使用 make install 指令和 DESTDIR=*C:\Program Files* 來變更目的地。這些命令會導致 AWS CLI 安裝在 。 *C:\Program Files*\awscli

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

> **Note**
>
> **make install** 執行時，會在幕後完成以下步驟
>
> 1. 將下列其中一項移至已設定的安裝目錄：
>
>   - 如果安裝類型為 system-sandbox，移動您建置的虛擬環境。

- 如果安裝類型為 `portable-exe`，移動您建置的獨立可執行檔。

2. 在配置的 bin 目錄中為 `aws` 和 `aws_completer` 可執行檔建立 symlink。

步驟 5：驗證 AWS CLI 安裝

使用以下命令確認 AWS CLI 已成功安裝 ：

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

如果找不到 `aws` 命令，您可能需要重新啟動終端，才能更新為新的 symlink。如果您在安裝或解除安裝之後遇到其他問題 AWS CLI，請參閱 故障診斷錯誤 以取得常見的故障診斷步驟

工作流程範例

本節提供一些從來源安裝的基本工作流程範例。

基本 Linux 和 macOS 安裝

下列範例是基本安裝工作流程，其中 AWS CLI 安裝在 的預設位置`/usr/local/lib/aws-cli`。

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

自動化 Windows 安裝

> **Note**
>
> 您必須以系統管理員身分執行 PowerShell，才能使用此工作流程。

MSYS2 可以在 CI 設定中以自動方式使用，請參閱 MSYS2 文件的在 CI 使用 MSYS2。

Downloaded Tarball

下載 `awscli.tar.gz` 檔案、解壓縮，並安裝 AWS CLI。使用下列命令時，請取代下列路徑：

- C:\msys64\usr\bin\bash 取代為您的 MSYS2 路徑的位置。
- .\awscli-2.x.x\ 取代為解壓縮後的 awscli.tar.gz 資料夾名稱。
- PYTHON='C:\path\to\python.exe' 取代為您的本機 Python 路徑。

下列程式碼範例會使用 MSYS2 AWS CLI 從 PowerShell 自動建置和安裝 ，並指定要使用的 Python 本機安裝：

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

GitHub Repository

下載 awscli.tar.gz 檔案、解壓縮，並安裝 AWS CLI。使用下列命令時，請取代下列路徑：

- C:\msys64\usr\bin\bash 取代為您的 MSYS2 路徑的位置。
- C:path\to\cli\repository\ 取代為從 GitHub 到複製 AWS CLI 儲存庫的路徑。如需更多詳細資訊，請參閱 GitHub 文件中的《延伸儲存庫》
- PYTHON='C:\path\to\python.exe' 取代為您的本機 Python 路徑。

下列程式碼範例會使用 MSYS2 AWS CLI 從 PowerShell 自動建置和安裝 ，並指定要使用的 Python 本機安裝：

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

```
PS C:\> aws --version
```

Alpine Linux 容器

以下的範例 Dockerfile，可用於在 Alpine Linux 容器取得 AWS CLI 的可運作安裝，作為 Alpine 預先建置二進位檔的替代方案。使用此範例時，請將 *AWSCLI_VERSION* 取代為您所需的 AWS CLI 版本編號：

```
FROM python:3.8-alpine AS builder

ENV AWSCLI_VERSION=2.10.1

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

FROM python:3.8-alpine

RUN apk --no-cache add groff

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

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

此映像已建置，並從類似建置於 Amazon Linux 2 的容器 AWS CLI 叫用：

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

此影像的最終大小小於官方 Docker AWS CLI 影像的大小。如需 Docker 映像檔的相關資訊，請參閱 the section called “Amazon ECR Public/Docker”。

針對 AWS CLI 安裝和解除安裝錯誤進行故障診斷

如需安裝錯誤的故障診斷步驟，請參閱 故障診斷錯誤 了解常見的故障診斷步驟。如需最相關的故障診斷步驟，請參閱 the section called “命令未找到錯誤”、the section called “「aws --version」命令傳回的版本與您安裝的版本不同” 和 the section called “解除安裝 後，「aws --version」命令會傳回版本 AWS CLI”。

對於故障診斷指南未涵蓋的任何問題，請在 GitHub 上的 AWS CLI 儲存庫中搜尋含有 `source-distribution` 標籤的問題。如果沒有現有問題涵蓋您的錯誤，請建立新的問題，以從 AWS CLI 維護器取得協助。

後續步驟

安裝 之後 AWS CLI，您應該執行 the section called “設定”。

執行 的官方 Amazon ECR Public 或 Docker 映像 AWS CLI

本主題說明如何使用官方 Amazon Elastic Container Registry Public (Amazon ECR Public) 或 Docker Hub 映像，在 Docker 上執行、版本控制和設定 AWS CLI 版本 2。如需如何使用 Docker 的詳細資訊，請參閱 Docker 文件。

官方映像提供 AWS 直接支援和維護的隔離性、可攜性和安全性。這可讓您在容器型環境中使用 第 2 AWS CLI 版，而不必自行管理安裝。

主題

- 先決條件
- 在 Amazon ECR Public 和 Docker Hub 之間進行選擇
- 執行官方第 2 AWS CLI 版映像
- 關於官方映像的介面和向後相容性的說明
- 使用特定版本和標籤
- 更新至最新的官方映像
- 分享主機檔案、憑證、環境變數和組態
- 縮短 docker run 命令

先決條件

您必須已安裝 Docker。如需安裝說明，請參閱 Docker 網站。

若要驗證 Docker 的安裝，請執行下列命令並確認有輸出。

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

在 Amazon ECR Public 和 Docker Hub 之間進行選擇

我們建議您透過 Docker Hub 使用 Amazon ECR Public 進行 AWS CLI 映像。Docker Hub 對於大眾消費者的速率限制更嚴格，可能會導致限流問題。此外，Amazon ECR Public 會在多個區域複寫映像，以提供強大的可用性並處理區域中斷問題。

如需 Docker Hub 速率限制的詳細資訊，請參閱 Docker 網站上的了解 Docker Hub 速率限制。

執行官方第 2 AWS CLI 版映像

第一次使用 `docker run` 命令時，會將最新的映像下載到您的電腦。後續每次使用 `docker run` 命令時，都會從您的本機複本執行。

若要執行第 2 AWS CLI 版 Docker 映像，請使用 `docker run`命令。

Amazon ECR Public

官方第 2 AWS CLI 版 Amazon ECR Public 映像託管在 `aws-cli/aws-cli`儲存庫中的 Amazon ECR Public。

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

Docker Hub

官方第 2 AWS CLI 版 Docker 映像託管在 `amazon/aws-cli`儲存庫的 Docker Hub 上。

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

這是命令的運作方式：

- `docker run --rm -it repository/name` – 相當於 aws 可執行檔。每次執行此命令時，Docker 都會加速運轉已下載映像檔的容器，並執行您的 aws 命令。根據預設，映像會使用最新版本的 第 2 AWS CLI 版。

  例如，若要呼叫 Docker 中的 `aws --version` 命令，您會執行下列命令。

  Amazon ECR Public

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

  Docker Hub

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

- `--rm` – 會指定在命令結束後清理容器。
- `-it` – 會指定使用 stdin 開啟 pseudo-TTY。這可讓您在容器執行時提供第 2 AWS CLI 版的輸入，例如使用 `aws configure`和 `aws help`命令。當選擇是否省略 `-it` 時，請考慮下列事項：
  - 如果您正在執行指令碼，則不需要 `-it`。
  - 如果您的指令碼發生錯誤，省略 Docker 呼叫中的 `-it` 也許可以解決問題。
  - 如果您嘗試管道輸出，`-it` 可能會導致錯誤，省略 Docker 呼叫中的 `-it` 也許可以解決此問題。如果您希望保留 `-it` 標誌，但仍然想管道輸出，在預設情況下停用用户端分頁程式該 AWS CLI 使用應該可以解決問題。

如需有關 `docker run` 命令的詳細資訊，請參閱《Docker 參考指南》。

關於官方映像的介面和向後相容性的說明

- 映像上唯一支援的工具是 AWS CLI。只有 aws 可執行檔案應該直接運行。例如，即使 `less`和 `groff` 明確安裝在映像上，也不應直接在 AWS CLI 命令之外執行。
- 此 `/aws` 工作目錄由使用者控制。除非使用者在執行 AWS CLI 命令時指示，否則映像不會寫入此目錄。
- 依賴最新標籤時沒有向後相容性保證。為了保證向後相容性，您必須固定到特定 `<major.minor.patch>` 標籤，因為這些標籤是不可變的；它們只會被推送一次。

使用特定版本和標籤

官方 AWS CLI 第 2 版映像具有多個您可以使用的版本，從版本 開始2.0.6。若要執行第 2 AWS CLI 版的特定版本，請將適當的標籤附加到您的docker run命令。第一次使用 docker run 命令搭配標籤時，會將該標籤的最新映像下載到您的電腦。後續每次使用 docker run 搭配該標籤命令時，都會從您的本機複本執行。

您可以使用兩種類型的標籤：

- latest – 定義映像的最新版本 第 2 AWS CLI 版。建議您在需要 AWS CLI 第 2 版的最新版本時使用 latest 標籤。但是，依賴此標籤時，無法保證回溯相容性。預設會在 docker run 命令中使用 latest 標籤。若要明確使用 latest 標籤，請將標籤附加至容器映像檔名稱。

  Amazon ECR Public

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

  Docker Hub

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

- <major.minor.patch> – 定義影像的特定 AWS CLI 版本 2。如果您計劃在生產環境中使用官方映像，我們建議您使用 第 2 AWS CLI 版的特定版本，以確保回溯相容性。例如，若要執行 2.0.6 版，請將版本附加至容器映像檔名稱。

  Amazon ECR Public

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

  Docker Hub

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

更新至最新的官方映像

由於最新的映像只有在您第一次使用 docker run 命令時才會下載到您的電腦，因此您需要手動提取更新的映像檔。若要手動更新至最新版本，建議您提取標記 latest 的映像檔。提取映像會將最新版本下載到您的電腦。

Amazon ECR Public

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

Docker Hub

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

分享主機檔案、憑證、環境變數和組態

由於第 2 AWS CLI 版是在容器中執行，因此 CLI 預設無法存取主機檔案系統，其中包含組態和登入資料。若要共享主機檔案系統、憑證和組態至容器，請將主機系統的 `~/.aws` 目錄掛載至位於 `/root/.aws` 的容器，對 `-v` 命令加上 `docker run` 旗標。這可讓在容器中執行的 AWS CLI 版本 2 找到主機檔案資訊。

Amazon ECR Public

Linux 和 macOS

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

Windows 命令提示

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

Windows PowerShell

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

Docker Hub

Linux 和 macOS

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

Windows 命令提示

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

Windows PowerShell

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

如需有關 -v 旗標和裝載的詳細資訊，請參閱《Docker 參考指南》。

> **Note**
>
> 如需關於 config 與 credentials 檔案的資訊，請參閱 the section called “組態設定”。

範例 1：提供憑證和組態

在此範例中，我們在執行 s3 ls 命令時提供主機憑證和組態，以列出您在 Amazon Simple Storage Service (Amazon S3) 的儲存貯體。下列範例使用 AWS CLI 登入資料和組態檔案的預設位置，若要使用不同的位置，請變更檔案路徑。

Amazon ECR Public

Linux 和 macOS

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

Windows 命令提示

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

Windows PowerShell

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

Docker Hub

Linux 和 macOS

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

Windows 命令提示

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

Windows PowerShell

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

您可以使用 -e 旗標呼叫特定系統的環境變數。若要使用環境變數，請依名稱進行呼叫。

Amazon ECR Public

Linux 和 macOS

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

Windows 命令提示

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

Windows PowerShell

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

Docker Hub

Linux 和 macOS

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

Windows 命令提示

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

Windows PowerShell

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

範例 2：將 Amazon S3 檔案下載到主機系統

對於某些第 2 AWS CLI 版命令，您可以從容器中的主機系統讀取檔案，或將檔案從容器寫入主機系統。

在這個範例中，我們透過將目前的工作目錄掛載到容器的 `/aws` 目錄，將 S3 物件 `s3://aws-cli-docker-demo/hello` 下載到您的本機檔案系統。藉由將 `hello` 物件下載至容器的 `/aws` 目錄，檔案也會儲存至主機系統目前的工作目錄。

Amazon ECR Public

Linux 和 macOS

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

Windows 命令提示

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

Windows PowerShell

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

Docker Hub

Linux 和 macOS

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

Windows 命令提示

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

Windows PowerShell

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

若要確認本機檔案系統中存在下載的檔案，請執行下列步驟。

Linux 和 macOS

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

Windows PowerShell

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

範例 3：使用 AWS_PROFILE 環境變數

您可以使用 `-e` 旗標呼叫特定系統的環境變數。呼叫您要使用的每個環境變數。在此範例中，我們在執行 `s3 ls` 命令時提供主機憑證、組態和 *`AWS_PROFILE`* 環境變數，以列出您在 Amazon Simple Storage Service (Amazon S3) 的儲存貯體。

Amazon ECR Public

Linux 和 macOS

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

Windows 命令提示

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

Windows PowerShell

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

Docker Hub

Linux 和 macOS

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

Windows 命令提示

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

Windows PowerShell

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

縮短 docker run 命令

若要縮短 `docker run` 命令，建議您使用作業系統的功能在 Linux 和 macOS 中建立 symbolic link (符號連結) 或 alias，或在 Windows 中建立 doskey。若要設定 aws 別名，您可以執行下列其中一個命令：

- 如需 aws 命令的基本存取權限，請執行下列命令。

  Amazon ECR Public

  Linux 和 macOS

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

  Windows 命令提示

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

  Windows PowerShell

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

  Docker Hub

  Linux 和 macOS

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

  Windows 命令提示

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

  Windows PowerShell

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

- 如需使用 aws 命令時主機檔案系統和組態設定的存取權，請執行以下項目。

Amazon ECR Public

Linux 和 macOS

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

Windows 命令提示

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

Windows PowerShell

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

Docker Hub

Linux 和 macOS

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

Windows 命令提示

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

Windows PowerShell

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

- 若要指定要在 aws 別名中使用的特定版本，請附加您的版本標籤：

Amazon ECR Public

Linux 和 macOS

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

Windows 命令提示

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

Windows PowerShell

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

Docker Hub

Linux 和 macOS

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

Windows 命令提示

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

Windows PowerShell

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

設定別名後，您可以從容器內執行第 2 AWS CLI 版，就好像在主機系統上安裝一樣。

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

設定 AWS CLI

本主題說明如何快速設定 AWS Command Line Interface (AWS CLI) 用來與 互動的基本設定 AWS。其中包括您的安全登入資料、預設輸出格式和預設 AWS 區域。

主題

- 收集您的憑證資訊以進行程式設計存取
- 設定新的組態和憑證
- 使用現有組態和憑證檔案

收集您的憑證資訊以進行程式設計存取

如果您想要與 AWS 外部互動，則需要程式設計存取 AWS Management Console。針對驗證和憑證指示，請從下列選項擇一：

身分驗證類型	用途	指示
IAM Identity Center 人力資源使用者短期憑證	（建議）使用 IAM Identity Center 人力資源使用者的短期憑證。 安全最佳實務是 AWS Organizations 搭配 IAM Identity Center 使用 。它將短期憑證與使用者目錄結合，例如內建 IAM Identity Center 目錄或 Active Directory。	the section called “IAM Identity Center 驗證”
IAM 使用者短期憑證	使用 IAM 使用者短期憑證，這比長期憑證更安全。如果您的登入資料遭到入侵，在過期之前，可以使用這些登入資料的時間有限。	the section called “短期憑證”

身分驗證類型	用途	指示
Amazon EC2 執行個體上的 IAM 或 IAM Identity Center 使用者。	使用 Amazon EC2 執行個體中繼資料，使用指派給 Amazon EC2 執行個體的角色來查詢臨時登入資料。	the section called “Amazon EC2 中繼資料”
擔任 許可的角色	配對另一個登入資料方法，並擔任角色以暫時存取 AWS 服務 您的使用者，可能無法存取 。	the section called “IAM 角色”
IAM 使用者長期憑證	（不建議）使用沒有過期的長期登入資料。	the section called “IAM 使用者”
IAM 或 IAM Identity Center 人力資源使用者的外部儲存	（不建議）配對其他登入資料方法，但將登入資料值存放在外部的位置 AWS CLI。此方法僅與存放登入資料的外部位置一樣安全。	the section called “外部憑證”

設定新的組態和憑證

會將您的組態和登入資料資訊 AWS CLI 存放在 `credentials`和 `config`檔案中的設定檔 （設定集合）。

您主要能透過兩種方法進行快速設定：

- 使用 AWS CLI 命令進行設定
- 手動編輯憑證和設定檔

下列範例會針對每個驗證方式使用範例值。使用您自己的值來取代範例值。

使用 AWS CLI 命令進行設定

一般來說，您偏好的終端中的 `aws configure` 或 `aws configure sso` 命令是設定您的 AWS CLI 安裝最快的方法。會根據您偏好的登入資料方法， AWS CLI 提示您提供相關資訊。根據預設，當您執行未明確指定要使用之設定檔的 AWS CLI 命令時，會使用此設定檔中的資訊。

如需 `credentials` 和 `config` 檔案的詳細資訊，請參閱 中的組態和登入資料檔案設定 AWS CLI。

IAM Identity Center (SSO)

此範例適用於 AWS IAM Identity Center 使用`aws configure sso`精靈。如需詳細資訊，請參閱the section called “IAM Identity Center 驗證”。

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

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

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

Using the account ID 111122223333

There are 2 roles available to you.
> ReadOnly
  FullAccess

Using the role name "ReadOnly"

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

IAM Identity Center (Legacy SSO)

此範例適用於 AWS IAM Identity Center 使用`aws configure sso`精靈的舊版方法。若要使用舊版 SSO，請將工作階段名稱留白。如需詳細資訊，請參閱the section called “IAM Identity Center 驗證”。

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

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

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

Using the account ID 111122223333

There are 2 roles available to you.
> ReadOnly
  FullAccess

Using the role name "ReadOnly"

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

Short-term credentials

此範例適用於來自 AWS Identity and Access Management的短期憑證。aws 配置精靈是用於設定初始值，而 `aws configure set` 命令會指派所需的最後值。如需詳細資訊，請參閱the section called “短期憑證”。

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

IAM role

此範例適用於 IAM 角色之擔任。使用 IAM 角色的設定檔會從其他設定檔提取憑證，接著套用 IAM 角色許可。在下列範例中，`default` 是憑證的來源設定檔，且 `user1` 借用相同憑證來擔任新角色。此程序中沒有精靈，因此每個值皆使用 `aws configure set` 命令設定。如需詳細資訊，請參閱the section called “IAM 角色”。

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

```
$ aws configure set output json
```

Amazon EC2 instance metadata credentials

此範例適用從託管 Amazon EC2 執行個體中繼資料取得的憑證。此程序中沒有精靈，因此每個值皆使用 `aws configure set` 命令設定。如需詳細資訊，請參閱the section called “Amazon EC2 中繼資料”。

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

Long-term credentials

> **Warning**
>
> 為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 AWS IAM Identity Center。

此範例適用於來自 AWS Identity and Access Management的長期憑證。如需詳細資訊，請參閱the section called “IAM 使用者”。

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

如需深入了解驗證與憑證的詳細資訊，請參閱 驗證與存取憑證。

手動編輯憑證和設定檔

複製貼上資訊時，建議您手動編輯 `config` 和 `credentials` 檔案。根據您偏好的憑證方法，檔案會以不同方式設定。

檔案會儲存在 `.aws` 資料夾下的主目錄中。主目錄的位置根據作業系統而不同，但在 Windows 中是使用環境變數 `%UserProfile%` 來參考，而在 Unix 系統中是使用 `$HOME` 或 ~ (波狀符號) 來參考。如需這些設定儲存位置的詳細資訊，請參閱 the section called “組態設定存放在何處？”。

以下範例使用範例值，且顯示 `default` 設定檔和名為 `user1` 的設定檔。使用您自己的值來取代範例值。如需 `credentials` 和 `config` 檔案的詳細資訊，請參閱 中的組態和登入資料檔案設定 AWS CLI。

IAM Identity Center (SSO)

此範例適用於 AWS IAM Identity Center。如需詳細資訊，請參閱the section called “IAM Identity Center 驗證”。

憑證檔案

此驗證方法不會使用 `credentials` 檔案。

組態檔案

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

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

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

IAM Identity Center (Legacy SSO)

此範例適用於 的舊版方法 AWS IAM Identity Center。如需詳細資訊，請參閱the section called “IAM Identity Center 驗證”。

憑證檔案

此驗證方法不會使用 `credentials` 檔案。

組態檔案

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

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

Short-term credentials

此範例適用於來自 AWS Identity and Access Management的短期憑證。如需詳細資訊，請參閱the section called “短期憑證”。

憑證檔案

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

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

組態檔案

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

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

IAM role

此範例適用於 IAM 角色之擔任。使用 IAM 角色的設定檔會從其他設定檔提取憑證，接著套用 IAM 角色許可。在下列範例中，`default` 是憑證的來源設定檔，且 `user1` 借用相同憑證來擔任新角色。如需詳細資訊，請參閱the section called “IAM 角色”。

憑證檔案

`credentials` 檔案取決於您的來源設定檔使用哪種身分驗證。在下列範例中，來源設定檔是使用短期憑證。

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

組態檔案

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

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

Amazon EC2 instance metadata credentials

此範例適用從託管 Amazon EC2 執行個體中繼資料取得的憑證。如需詳細資訊，請參閱the section called “Amazon EC2 中繼資料”。

憑證檔案

此驗證方法不會使用 `credentials` 檔案。

組態檔案

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

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

Long-term credentials

> **Warning**
>
> 為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 AWS IAM Identity Center。

此範例適用於來自 AWS Identity and Access Management的長期憑證。如需詳細資訊，請參閱the section called “IAM 使用者”。

憑證檔案

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

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

組態檔案

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

```
output=json

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

如需深入了解驗證與憑證的詳細資訊，請參閱 驗證與存取憑證。

使用現有組態和憑證檔案

如果您有現有的組態和憑證檔案，則這些檔案可用於 AWS CLI。

若要使用 `config` 和 `credentials` 檔案，將它們移動到主目錄中名為 `.aws` 的資料夾。主目錄的位置根據作業系統而不同，但在 Windows 中是使用環境變數 `%UserProfile%` 來參考，而在 Unix 系統中是使用 `$HOME` 或 `~` (波狀符號) 來參考。

您可以將 `AWS_CONFIG_FILE` 和 `AWS_SHARED_CREDENTIALS_FILE` 環境變數設為其他本機路徑，以指定 `config` 和 `credentials` 檔案的非預設位置。如需詳細資訊，請參閱 設定 的環境變數 AWS CLI。

如需關於組態和憑證檔案的詳細資訊，請參閱 the section called “組態設定”。

設定 的設定 AWS CLI

本節說明如何設定 AWS Command Line Interface (AWS CLI) 用來互動的設定 AWS。這些索引標籤包括以下項目：

- 憑證會識別呼叫 API 的人員。存取憑證用於加密對 AWS 伺服器的請求，以確認您的身分並擷取相關聯的許可政策。這些權限會決定您可以執行哪些動作。如需設定憑證的相關資訊，請參閱 驗證與存取憑證。
- 告知 AWS CLI 如何處理請求的其他組態詳細資訊，例如預設輸出格式和預設 AWS 區域。

> **Note**
>
> AWS 要求以密碼編譯方式簽署所有傳入的請求。會為您 AWS CLI 執行此操作。「簽章」包含日期/時間戳記。因此，您必須確定電腦的日期和時間設定正確。如果您未這麼做，且簽章中的日期/時間離 AWS 服務辨識的日期/時間太遠， AWS 會拒絕請求。

組態和憑證優先順序

登入資料和組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。某些位置的優先順序高於其他位置。 AWS CLI 登入資料和組態設定的優先順序如下：

1. 命令列選項 – 覆寫任何其他位置 (例如 `--region`、`--output` 和 `--profile` 參數) 的設定。
2. 環境變數 – 您可以將數值存放在環境變數中。
3. 擔任角色—透過組態或 `assume-role` 命令，獲得 IAM 角色許可。
4. 擔任具有 Web 身分的角色—透過組態或 `assume-role-with-web-identity` 命令，使用 Web 身分獲得 IAM 角色許可。
5. AWS IAM Identity Center – 當您執行 `aws configure sso`命令時，會更新存放在 `config` 檔案中的 IAM Identity Center 組態設定。然後，在您執行 `aws sso login`命令時驗證登入資料。`config` 檔案在 Linux 或 macOS 上位於 `~/.aws/config`，在 Windows 上位於 `C:\Users\`*`USERNAME`*`\.aws\config`。
6. 憑證檔案 – 當您執行 `aws configure` 命令時，會更新 `credentials` 和 `config` 檔案。`credentials` 檔案在 Linux 或 macOS 上位於 `~/.aws/credentials`，在 Windows 上位於 `C:\Users\`*`USERNAME`*`\.aws\credentials`。

7. 自訂程序—從外部來源取得憑證。
8. 組態檔 – 當您執行 `aws configure` 命令時，會更新 `credentials` 和 `config` 檔案。`config` 檔案在 Linux 或 macOS 上位於 `~/.aws/config`，在 Windows 上位於 `C:\Users\USERNAME\.aws\config`。
9. 容器憑證 – 您可以將 IAM 角色與您的每一個 Amazon Elastic Container Service (Amazon ECS) 任務定義關聯起來。然後，該角色的臨時憑證就可供該任務的容器使用。如需詳細資訊，請參閱《Amazon Elastic Container Service 開發人員指南》中的任務 IAM 角色。
10. Amazon EC2 執行個體設定檔憑證 – 您可以將 IAM 角色與您的每一個 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體關聯起來。然後，該角色的臨時憑證就可供執行個體中執行的程式碼使用。憑證是透過 Amazon EC2 中繼資料服務傳遞。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Amazon EC2 的 IAM 角色和《IAM 使用者指南》中的使用執行個體描述檔。 Amazon EC2

本區段的其他主題

- the section called “組態設定”
- the section called “環境變數”
- the section called “命令列選項”
- the section called “ 命令完成”
- the section called “重試”
- the section called “HTTP 代理”

中的組態和登入資料檔案設定 AWS CLI

您可以將常用的組態設定和憑證儲存在 AWS CLI維護的檔案中。

這些文件被分為 `profiles`。根據預設， AWS CLI 會使用在名為 的設定檔中找到的設定`default`。若要使用替代設定，您可以建立和參考額外的描述檔。

您可以設定其中一個支援的環境變數，或使用命令列參數，以覆寫個別設定。如需組態設定優先順序的詳細資訊，請參閱 設定 的設定 AWS CLI。

> **Note**
>
> 如需設定憑證的相關資訊，請參閱 驗證與存取憑證。

主題

- 組態和憑證檔案的格式
- 組態設定存放在何處？
- 使用命名設定檔
- 使用命令設定和檢視組態設定
- 設定新的組態和憑證命令範例
- 支援的 config 檔案設定

組態和憑證檔案的格式

config 和 credentials 檔案會組織成區段。區段包括設定檔、SSO 工作階段和服務。區段是具名的設定集合，且會持續到發生其他區段定義列為止。許多設定檔和區段可以儲存於 config 和 credentials 檔案。

這些檔案是使用下列格式的純文字檔案：

- 區段名稱會以方括號 [] 括起來，例如 [default]、[profile *user1*] 和 [sso-session]。
- 區段中的所有項目均採用 setting_name=value 的一般形式。
- 您可以在列的開頭使用雜湊字元 (#)，為列加上註解。

config 和 credentials 檔案包含下列區段類型：

- 區段類型：profile
- 區段類型：sso-session
- 區段類型：services

區段類型：**profile**

根據檔案，設定檔區段名稱會使用下列格式：

- 組態檔案：[default] [profile *user1*]
- 憑證檔案：[default] [*user1*]

  在 credentials 檔案中建立項目時，請勿使用 profile 一字。

每個描述檔可以指定不同的憑證，也可以指定不同的 AWS 區域和輸出格式。為 config 檔案中的設定檔命名時，請包含前綴詞「profile」，但不要將其包含在 credentials 檔案中。

下列範例顯示 credentials 和 config 檔案，兩者皆具有兩個設定檔、區域和指定輸出。當您執行不含指定設定檔的 AWS CLI 命令時會使用第一個 [預設]。當您使用 --profile user1 參數執行 AWS CLI 命令時，會使用第二個。

IAM Identity Center (SSO)

此範例適用於 AWS IAM Identity Center。如需詳細資訊，請參閱the section called “IAM Identity Center 驗證”。

憑證檔案

此驗證方法不會使用 credentials 檔案。

組態檔案

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

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

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

IAM Identity Center (Legacy SSO)

此範例適用於 的舊版方法 AWS IAM Identity Center。如需詳細資訊，請參閱the section called “IAM Identity Center 驗證”。

憑證檔案

此驗證方法不會使用 `credentials` 檔案。

組態檔案

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

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

Short-term credentials

此範例適用於來自 AWS Identity and Access Management的短期憑證。如需詳細資訊，請參閱the section called “短期憑證”。

憑證檔案

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

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

組態檔案

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

```
output=json

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

IAM role

此範例適用於 IAM 角色之擔任。使用 IAM 角色的設定檔會從其他設定檔提取憑證，接著套用 IAM 角色許可。在下列範例中，`default` 是憑證的來源設定檔，且 `user1` 借用相同憑證來擔任新角色。如需詳細資訊，請參閱the section called “IAM 角色”。

憑證檔案

`credentials` 檔案取決於您的來源設定檔使用哪種身分驗證。在下列範例中，來源設定檔是使用短期憑證。

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

組態檔案

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

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

Amazon EC2 instance metadata credentials

此範例適用從託管 Amazon EC2 執行個體中繼資料取得的憑證。如需詳細資訊，請參閱the section called “Amazon EC2 中繼資料”。

憑證檔案

此驗證方法不會使用 `credentials` 檔案。

組態檔案

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

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

Long-term credentials

> **Warning**
>
> 為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 AWS IAM Identity Center。

此範例適用於來自 AWS Identity and Access Management的長期憑證。如需詳細資訊，請參閱the section called “IAM 使用者”。

憑證檔案

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

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

組態檔案

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

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

如需詳細資訊以及其他驗證和憑證方式，請參閱 the section called “IAM 使用者”。

區段類型：**sso-session**

config 檔案的 sso-session區段用於將用於取得 SSO 存取權杖的組態變數分組，然後可用於取得 AWS 登入資料。使用下列的設定：

- (必要) sso_start_url
- (必要) sso_region
- sso_account_id
- sso_role_name
- sso_registration_scopes

定義 sso-session 區段並將其與設定檔相關聯。sso_region 和 sso_start_url 必須在 sso-session 區段內設定。一般而言，sso_account_id 和 sso_role_name 必須在 profile 區段中設定，以讓開發套件請求 SSO 憑證。

下列範例會將開發套件設定為請求 SSO憑認證，並支援自動字符重新整理：

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

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

這也讓 sso-session 組態能在多個設定檔中重複使用：

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

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

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

然而，SSO 字符組態並非所有情況都需要 sso_account_id 和 sso_role_name。如果您的應用程式只使用支援承載身分驗證的 AWS 服務，則不需要傳統的 AWS 憑證。承載身分驗證是一種 HTTP 身分驗證結構描述，使用稱為承載字符的安全字符。在這種情況下，sso_account_id 和 sso_role_name 並非必要資訊。請參閱您 AWS 服務的個別指南，以判斷其是否支援承載字符授權。

此外，註冊範圍也可以設定為 sso-session 的一部分。範圍是 OAuth 2.0 中的一種機制，用於限制應用程式對使用者帳戶的存取。應用程式可以請求一個或多個範圍，則合法給應用程式的存取字符將僅限於授予的範圍。這些範圍定義為已註冊 OIDC 用戶端和用戶端擷取的存取字符進行授權所需請求的許可。下列範例設定 sso_registration_scopes 以提供列出帳戶/角色的存取權限：

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

身分驗證字符會以基於工作階段名稱的檔案名稱快取至 ~/.aws/sso/cache 目錄下的磁碟。

如需有關組態類型的詳細資訊，請參閱 the section called “IAM Identity Center 驗證”。

區段類型：**services**

services 區段是一組設定，可設定 AWS 服務 請求的自訂端點。然後，設定檔會連結至 services 區段。

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

該 services 區段由 <SERVICE> = 列分隔成子區段，其中 <SERVICE> 是 AWS 服務 識別碼金鑰。 AWS 服務 識別符是以 API 模型的 為基礎serviceId，方法是將所有空格替換為底線，並縮小所有字母大小寫。如需要在 services 區段中使用的所有服務識別碼金鑰的清單，請參閱 在 中使用端點 AWS CLI。服務識別碼金鑰後面接續巢狀化設定，每個設定獨佔一列並縮排兩個空格。

以下範例將端點設定為用於向 *dev* 設定檔中使用的 *my-services* 區段中的 Amazon DynamoDB 服務發出的請求。任何緊接著縮排的列都包含在該子區段中，並適用於該服務。

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

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

如需服務特定端點詳細資訊，請參閱 在 中使用端點 AWS CLI。

如果您的設定檔具有透過 source_profile 參數設定用於 IAM 假設角色功能，以角色為基礎的憑證，則 SDK 只會使用指定設定檔的服務組態。它不會使用與其連結的角色的設定檔。例如，使用下列共用 config 檔案：

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

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

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

如果您使用設定檔 B 並在程式碼中呼叫 Amazon EC2，則端點會解析為 https://profile-b-ec2-endpoint.aws。如果您的程式碼向任何其他服務發出要求，端點解析將不會遵循任何自訂邏輯。端點未解析為設定檔 A 中定義的全域端點。若要讓全域端點對設定檔 B 生效，您需要直接在 B 設定檔中設定 endpoint_url。

組態設定存放在何處？

會將您以 指定的敏感登入資料資訊 AWS CLI 儲存在主目錄中名為 的資料夾中credentials，名為 .aws的aws configure本機檔案中。您使用 aws configure 指定的較不敏感組態選項，則存放在名為 config 的本機檔案中，而此檔案也是存放在主目錄中的 .aws 資料夾。

> **將憑證存放到組態檔中**
>
> 您可以將所有設定檔設定保留在單一檔案中，因為 AWS CLI 可以從config檔案讀取登入資料。如果兩個檔案中都有共用相同名稱之設定檔的憑證，則憑證檔案中的金鑰具有高優先順序。建議您在 credentials 檔案中保留憑證。各種語言軟體開發套件 (SDK) 也使用這些檔案。如果您除了 之外還使用其中一個 SDKs AWS CLI，請確認登入資料是否應存放在自己的檔案中。

主目錄的位置根據作業系統而不同，但在 Windows 中是使用環境變數 %UserProfile% 來參考，而在 Unix 系統中是使用 $HOME 或 ~ (波狀符號) 來參考。您可以將 AWS_CONFIG_FILE 和 AWS_SHARED_CREDENTIALS_FILE 環境變數設為其他本機路徑，以指定檔案的非預設位置。如需詳細資訊，請參閱 設定 的環境變數 AWS CLI。

當您使用指定 AWS Identity and Access Management (IAM) 角色的共用設定檔時， 會 AWS CLI 呼叫 AWS STS AssumeRole操作來擷取臨時登入資料。然後會存放這些憑證 (在 ~/.aws/cli/cache 中)。後續 AWS CLI 命令會使用快取的暫時登入資料，直到過期為止，此時 AWS CLI 會自動重新整理登入資料。

使用命名設定檔

若未明確定義任何設定檔，系統會使用 default 設定檔。

若要使用命名設定檔，請新增 `--profile` *`profile-name`* 選項到命令中。下列範例列出使用 user1 設定檔中定義的憑證和設定的所有 Amazon EC2 執行個體。

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

如果您要將一個具名設定檔用於多個命令，請將 AWS_PROFILE 環境變數設定為預設設定檔，即可避免在每個命令中指定設定檔。您可以使用 --profile 參數來覆寫此設定。

Linux or macOS

```
$ export AWS_PROFILE=user1
```

Windows

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

使用 `set` 設定環境變數會變更使用的數值，直到目前命令提示工作階段結束或直到您將該變數設為其他數值。

使用 `setx` 來設定環境變數時，將會變更您在執行命令後建立的所有命令 Shell 中的數值。不會影響您執行命令當時已執行的任何命令 Shell。關閉並重新啟動命令 Shell，以查看變更的效果。

設定環境變數會變更預設的設定檔，直到 Shell 工作階段結束或直到您將該變數設為其他數值。您可以將環境變數放入 Shell 的啟動指令碼中，讓它們跨未來的工作階段持續生效。如需詳細資訊，請參閱設定 的環境變數 AWS CLI。

使用命令設定和檢視組態設定

有數種方法可以使用命令，來檢視和設定您的組態設定。

aws configure

執行此命令以快速設定和檢視您的 憑證、區域和輸出格式。下列範例顯示範本值。

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

aws configure set

您可以使用 `aws configure set` 來設定任何憑證或組態設定。使用 `--profile` 設定來指定您想要檢視或修改的設定檔。

例如，以下命令會在名為 `integ` 的設定檔中進行 `region` 設定。

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

若要移除設定，請在文字編輯器中手動刪除 `config`和 `credentials` 檔案中的設定。

aws configure get

您可以擷取已使用 `aws configure get` 設定的任何憑證或組態設定。使用 `--profile` 設定來指定您想要檢視或修改的設定檔。

例如，以下命令會在名為 `region` 的設定檔中擷取 `integ` 設定。

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

如果輸出是空的，則設定未明確設定，將會使用預設值。

aws configure import

匯入從 IAM Web 主控台產生的 CSV 憑證。這不適用於從 IAM Identity Center 產生的憑證；使用 IAM Identity Center 的客戶應使用 aws 來設定 sso。CSV 檔案將會匯入，且設定檔名稱與使用者名稱相符。CSV 檔案必須包含以下標頭。

- 使用者名稱
- 存取金鑰 ID
- 私密存取金鑰

> **Note**
>
> 在初始金鑰對建立過程中，一旦關閉 Download .csv file (下載 .csv 檔案) 對話方塊中，您將無法在關閉對話方塊之後存取私密金鑰。如果您需要 `.csv` 檔案，您必須自己建立一個包含所需標頭和您所儲存金鑰對資訊的檔案。如果您無法存取您的金鑰對資訊，則必須建立新的金鑰對。

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

aws configure list

若要列出組態資料，請使用 `aws configure list` 命令。此命令會列出設定檔、存取金鑰、私密金鑰，以及用於所指定設定檔的區域組態資訊。對於每個組態項目，它會顯示值、擷取組態值的位置，以及組態變數名稱。

例如，如果您在 環境變數 AWS 區域 中提供 ，此命令會顯示您設定的區域名稱、此值來自 環境變數，以及 環境變數的名稱。

若採用臨時憑證方法 (例如，角色和 IAM Identity Center)，此命令會顯示臨時快取的存取金鑰，並顯示私密存取金鑰。

```
$ aws configure list
```

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

aws configure list-profiles

若要列出所有設定檔名稱，請使用 `aws configure list-profiles` 命令。

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

aws configure sso

執行此命令以快速設定和檢視您的 AWS IAM Identity Center 登入資料、區域和輸出格式。下列範例顯示範本值。

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

aws configure sso-session

執行此命令，在 和 `config` 檔案的 sso-session 區段中快速設定和檢視您的 AWS IAM Identity Center 登入資料、區域`credentials`和輸出格式。下列範例顯示範本值。

```
$ aws configure sso-session
SSO session name: my-sso
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]: us-east-1
SSO registration scopes [None]: sso:account:access
```

aws configure export-credentials

執行此命令以匯出目前以指定格式設定的登入資料。根據預設， 命令會以 `process` 格式匯出預設登入資料，這是 AWS SDKs和工具登入資料格式支援的 JSON 格式。

```
$ aws configure export-credentials
{
  "Version": 1,
  "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
  "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
}
```

若要匯出特定設定檔和格式，請使用 `--profile`和 `--format`選項。格式選項如下所示：

- （預設）**process** - SDKs和工具`credential_process`組態支援的 AWS JSON 格式。
- **env** - 匯出 Shell 格式的環境變數。
- **env-no-export** - shell 格式的非匯出環境變數。
- **powershell** - PowerShell 格式的環境變數。
- **windows-cmd** - Windows Command Line 格式的環境變數。

下列範例會將`user1`設定檔匯出為匯出的 shell 格式。

```
$ aws configure export-credentials --profile user1 --format env
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

設定新的組態和憑證命令範例

您可以從下列範例了解如何針對不同驗證方法，用憑證、區域和指定輸出來設定預設設定檔。

IAM Identity Center (SSO)

此範例適用於 AWS IAM Identity Center 使用`aws configure sso`精靈。如需詳細資訊，請參閱the section called “IAM Identity Center 驗證”。

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

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

There are 2 AWS accounts available to you.
```

```
> DeveloperAccount, developer-account-admin@example.com (111122223333)
  ProductionAccount, production-account-admin@example.com (444455556666)

Using the account ID 111122223333

There are 2 roles available to you.
> ReadOnly
  FullAccess

Using the role name "ReadOnly"

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

IAM Identity Center (Legacy SSO)

此範例適用於 AWS IAM Identity Center 使用`aws configure sso`精靈的舊版方法。若要使用舊版 SSO，請將工作階段名稱留白。如需詳細資訊，請參閱the section called “IAM Identity Center 驗證”。

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

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

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

Using the account ID 111122223333

There are 2 roles available to you.
> ReadOnly
  FullAccess

Using the role name "ReadOnly"

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

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

Short-term credentials

此範例適用於來自 AWS Identity and Access Management的短期憑證。aws 配置精靈是用於設定初始值，而 `aws configure set` 命令會指派所需的最後值。如需詳細資訊，請參閱the section called “短期憑證”。

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

IAM role

此範例適用於 IAM 角色之擔任。使用 IAM 角色的設定檔會從其他設定檔提取憑證，接著套用 IAM 角色許可。在下列範例中，`default` 是憑證的來源設定檔，且 `user1` 借用相同憑證來擔任新角色。此程序中沒有精靈，因此每個值皆使用 `aws configure set` 命令設定。如需詳細資訊，請參閱the section called “IAM 角色”。

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

Amazon EC2 instance metadata credentials

此範例適用從託管 Amazon EC2 執行個體中繼資料取得的憑證。此程序中沒有精靈，因此每個值皆使用 `aws configure set` 命令設定。如需詳細資訊，請參閱the section called “Amazon EC2 中繼資料”。

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

Long-term credentials

> **Warning**
>
> 為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 AWS IAM Identity Center。

此範例適用於來自 AWS Identity and Access Management的長期憑證。如需詳細資訊，請參閱the section called “IAM 使用者”。

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

支援的 `config` 檔案設定

主題

- 全域設定
- S3 自訂命令設定

`config` 檔案中支援下列設定。將會使用指定的 (或預設) 描述檔中列出的數值，除非因為有相同名稱的環境變數或相同名稱的命令列選項而被覆寫。如需有關哪些順序設定具有高優先順序的詳細資訊，請參閱 設定 的設定 AWS CLI

全域設定

`account_id_endpoint_mode`

指定是否使用 AWS 帳戶型端點 IDs來呼叫支援的 AWS 服務。如需帳戶型端點的詳細資訊，請參閱the section called “帳戶型端點”。

此設定可設定為下列項目：

- （預設）**`preferred`** – 如果可用，端點應包含帳戶 ID。
- **`disabled`** – 已解析的端點不包含帳戶 ID。

- **required** – 端點必須包含帳戶 ID。如果帳戶 ID 無法使用，軟體開發套件會擲回錯誤。

可由 AWS_ACCOUNT_ID_ENDPOINT_MODE 環境變數所覆寫。若要使用帳戶型端點，必須在AWS_ACCOUNT_ID環境變數或aws_account_id設定中設定 ID。

```
account_id_endpoint_mode = preferred
```

> **端點優先順序**
>
> 端點組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。會依特定順序 AWS CLI 檢查這些端點設定，並使用優先順序最高的端點設定。如需端點優先順序清單，請參閱 the section called “端點組態和設定優先順序”。

aws_access_key_id

指定用作登入資料一部分的 AWS 存取金鑰，以驗證命令請求。雖然這可以存放在 config 檔案中，但建議您存放在 credentials 檔案中。

可由 AWS_ACCESS_KEY_ID 環境變數所覆寫。您不能將存取金鑰 ID 指定為命令列選項。

```
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
```

aws_account_id

指定要用於對受支援 呼叫 AWS 的帳戶型端點 ID AWS 服務。如需帳戶型端點的詳細資訊，請參閱 the section called “帳戶型端點”。

可由 AWS_ACCOUNT_ID 環境變數所覆寫。AWS_ACCOUNT_ID_ENDPOINT_MODE 環境變數或account_id_endpoint_mode設定必須設定為 preferred或 required，才能使用此設定。

```
aws_account_id = 123456789EXAMPLE
```

> **端點優先順序**
>
> 端點組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。會依特定順序 AWS CLI 檢查這些端點設定，並使用優先順序最高的端點設定。如需端點優先順序清單，請參閱 the section called “端點組態和設定優先順序”。

aws_secret_access_key

指定用作登入資料一部分的 AWS 私密金鑰，以驗證命令請求。雖然這可以存放在 config 檔案中，但建議您存放在 credentials 檔案中。

可由 AWS_SECRET_ACCESS_KEY 環境變數所覆寫。您不能將私密存取金鑰指定為命令列選項。

```
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

aws_session_token

指定 AWS 工作階段字符。只有當您手動指定臨時的安全憑證時，才需要工作階段字符。雖然這可以存放在 config 檔案中，但建議您存放在 credentials 檔案中。

可由 AWS_SESSION_TOKEN 環境變數所覆寫。您不能將工作階段字符指定為命令列選項。

```
aws_session_token = AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
```

ca_bundle

指定用於驗證 SSL 憑證的憑證授權機構憑證套件 (副檔名為 .pem 的檔案)。

可由 AWS_CA_BUNDLE 環境變數或 --ca-bundle 命令列選項所覆寫。

```
ca_bundle = dev/apps/ca-certs/cabundle-2019mar05.pem
```

cli_auto_prompt

啟用 第 2 AWS CLI 版的自動提示。有兩種設定可以使用：

- 在您每次嘗試執行 aws 命令時，**on** 都會使用完整的自動提示模式。這包括在完整的命令或不完整的命令之後按下 ENTER。

  ```
  cli_auto_prompt = on
  ```

- **on-partial** 使用部分自動提示模式。如果命令不完整或因用戶端驗證錯誤而無法執行，則會使用自動提示。如果您有預先存在的指令碼、Runbook，或是您只希望系統自動提示您不熟悉的命令 (而非針對每個命令發出提示)，這個模式就特別有用。

```
cli_auto_prompt = on-partial
```

您可以使用 aws_cli_auto_prompt 環境變數或 --cli-auto-prompt 及 --no-cli-auto-prompt 命令列參數來覆寫此設定。

如需第 2 AWS CLI 版自動提示功能的詳細資訊，請參閱 在 中啟用和使用命令提示 AWS CLI。

cli_binary_format

指定 AWS CLI 版本 2 如何解譯二進位輸入參數。它可能是以下其中一個數值：

- base64 – 這是預設值。以二進位大型物件 (BLOB) 輸入的輸入參數接受 base64 編碼的字串。若要傳遞實際的二進位內容，請將內容放置在檔案中，並在參數值中使用 fileb:// 前綴，以提供檔案路徑和名稱。若要傳遞包含在檔案中的 base64 編碼文字，請在參數值中使用 file:// 前綴，以提供檔案路徑和名稱。
- raw-in-base64-out – 第 1 AWS CLI 版的預設 。如果設定的值為 raw-in-base64-out，則會將使用 file:// 字首參照的檔案讀取為文字，然後 AWS CLI 嘗試將其編碼為二進位。

此項目沒有同等環境變數。您可以使用 --cli-binary-format raw-in-base64-out 參數，在單一命令上指定數值。

```
cli_binary_format = raw-in-base64-out
```

如果您使用fileb://字首表示法參考檔案中的二進位值， AWS CLI 一律會預期檔案包含原始二進位內容，而不會嘗試轉換該值。

如果您使用file://字首表示法參考檔案中的二進位值， 會根據目前的cli_binary_format設定 AWS CLI 處理檔案。如果該設定的值為 base64（未明確設定時為預設值），則 AWS CLI 預期檔案包含 base64 編碼的文字。如果該設定的值為 raw-in-base64-out，則 AWS CLI 預期檔案包含原始二進位內容。

cli_history

預設為停用。此設定會啟用 AWS CLI命令歷史記錄。啟用此設定後， 會 AWS CLI 記錄 aws 命令的歷史記錄。

```
cli_history = enabled
```

您可以使用 aws history list 命令來列出歷史記錄，並且在 aws history show 命令中使用產生的 command_ids 取得詳細資訊。如需詳細資訊，請參閱 AWS CLI 參考指南中的 aws history。

cli_pager

指定用於輸出的分頁程式。根據預設，第 2 AWS CLI 版會透過作業系統的預設分頁程式傳回所有輸出。

可以由 AWS_PAGER 環境變數所覆寫。

```
cli_pager=less
```

cli_timestamp_format

指定輸出包含的時間戳記值格式。您可以指定下列任一數值：

- iso8601 – 第 2 AWS CLI 版的預設值。如果指定， 會根據 ISO 8601 AWS CLI 重新格式化所有時間戳記。

  ISO 8601 格式化的時間戳記如下所示。第一個範例會以國際標準時間 (UTC) 顯示時間，方法是在時間之後併入 Z。日期和時間使用 T 分隔。

  ```
  2019-10-31T22:21:41Z
  ```

  若要指定不同的時區，不要指定 Z，而是指定 + 或 - 與距離所需 UTC 之前或之後的小時數 (兩位數形式)。下列範例顯示與前一個範例相同的時間，但調整為太平洋標準時間，它是 UTC 後八小時。

  ```
  2019-10-31T14:21:41-08
  ```

- wire – 第 1 AWS CLI 版的預設值。如果指定，則 AWS CLI 會顯示與 HTTP 查詢回應中所接收完全相同的所有時間戳記值。

此項目沒有同等環境變數或命令列選項。

```
cli_timestamp_format = iso8601
```

credential_process

指定 AWS CLI 執行的外部命令，以產生或擷取要用於此命令的身分驗證憑證。命令必須以特定格式傳回憑證。如需有關如何使用此設定的詳細資訊，請參閱 在 中使用外部程序來採購登入資料 AWS CLI。

此項目沒有同等環境變數或命令列選項。

```
credential_process = /opt/bin/awscreds-retriever --username susan
```

credential_source

在 Amazon EC2 執行個體或容器中使用，以指定 AWS CLI 可以找到用於擔任您使用 role_arn 參數所指定角色的憑證。您無法在同一個描述檔中同時指定 source_profile 和 credential_source。

此參數可以有下列三個數值中的一個：

- 環境 – 指定 AWS CLI 是從環境變數擷取來源憑證。
- Ec2InstanceMetadata – 指定 AWS CLI 要使用連接到 EC2 執行個體描述檔的 IAM 角色來取得來源登入資料。
- EcsContainer – 指定 AWS CLI 使用連接到 ECS 容器的 IAM 角色做為來源憑證。

```
credential_source = Ec2InstanceMetadata
```

duration_seconds

指定角色工作階段的最大持續時間 (以秒為單位)。此數值的範圍可以從 900 秒 (15 分鐘) 到角色的最大工作階段持續時間設定的數值 (其最大值為 43200)。這是選用參數，並且依預設，此數值會設為 3600 秒。

endpoint_url

指定用於所有服務要求的端點。如果在 config 檔案的 services 區段中使用此設定，則端點僅用於指定的服務。如需詳細資訊，請參閱the section called “設定所有 的全域端點 AWS 服務”。

以下範例使用 Amazon S3 的全域端點 http://localhost:1234 和服務特定端點 http://localhost:4567。

```
[profile dev]
endpoint_url = http://localhost:1234
services = s3-specific

[services s3-specific]
s3 =
  endpoint_url = http://localhost:4567
```

> **端點優先順序**
>
> 端點組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。會依特定順序 AWS CLI 檢查這些端點設定，並使用優先順序最高的端點設定。如需端點優先順序清單，請參閱 the section called “端點組態和設定優先順序”。

ignore_configure_endpoint_urls

如果啟用， 會 AWS CLI 忽略 config 檔案中指定的所有自訂端點組態。有效值為 **true** 和 **false**。

```
ignore_configure_endpoint_urls = true
```

> **端點優先順序**
>
> 端點組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。會依特定順序 AWS CLI 檢查這些端點設定，並使用優先順序最高的端點設定。如需端點優先順序清單，請參閱 the section called “端點組態和設定優先順序”。

external_id

指定一個唯一識別符，第三方用來在其客戶帳戶擔任角色。這映射到 ExternalId 操作的 AssumeRole 參數。只有在角色的信任政策指定 ExternalId 的數值時，才需要此參數。如需詳細資訊，請參閱《IAM 使用者指南》中的如何在將 AWS 資源的存取權授予第三方時使用外部 ID。

max_attempts

指定重試 AWS CLI 處理常式使用的最大重試次數值，其中初始呼叫會計入您提供的max_attempts值。

您可以使用 AWS_MAX_ATTEMPTS 環境變數來覆寫此數值。

```
max_attempts = 3
```

mfa_serial

擔任角色時使用的 MFA 裝置識別碼。只有在所擔任角色的信任政策包含要求 MFA 身分驗證的條件時才具強制性。此數值可以是硬體裝置的序號 (例如 GAHT12345678) 或虛擬 MFA 裝置的 Amazon Resource Name (ARN) (例如 `arn:aws:iam::123456789012:mfa/user`)。

output

對於請求使用此描述檔的命令，指定預設輸出格式。您可以指定下列任何數值：

- **json** – 輸出的格式為 JSON 字串。
- **yaml** – 輸出的格式為 YAML 字串。
- **yaml-stream** – 輸出採用串流方式且格式為 YAML 字串。串流可加速處理大型資料類型。
- **text** – 輸出的格式是多行以 Tab 分隔的字串值。這對於將輸出傳遞給文字處理器 (如 grep、sed 或 awk) 非常有用。
- **table** – 輸出的格式為使用字元 +|- 形成儲存格框線的表格。它通常以「方便人類使用」的格式來呈現資訊，這種格式比其他格式更容易閱讀，但在編寫程式方面較不有用。

可由 AWS_DEFAULT_OUTPUT 環境變數或 `--output` 命令列選項所覆寫。

```
output = table
```

parameter_validation

指定 AWS CLI 用戶端是否在將參數傳送到 AWS 服務端點之前嘗試驗證參數。

- true – 這是預設值。如果指定， 會 AWS CLI 執行命令列參數的本機驗證。
- false – 如果指定，在將命令列參數傳送到 AWS 服務端點之前， AWS CLI 不會驗證命令列參數。

此項目沒有同等環境變數或命令列選項。

```
parameter_validation = false
```

region

指定 AWS 區域 針對使用此設定檔請求的命令，將請求傳送至 的 。

- 您可以指定可用於所選服務的任何區域程式碼，如 Amazon Web Services 一般參考 中的 AWS 區域與端點所列。

- aws_global 可讓您為支援全域端點的服務指定全域端點，以及區域端點，例如 AWS Security Token Service (AWS STS) 和 Amazon Simple Storage Service (Amazon S3)。

您可以使用 AWS_REGION 環境變數、 AWS_DEFAULT_REGION 環境變數或 --region 命令列選項來覆寫此數值。

```
region = us-west-2
```

request_checksum_calculation

指定何時計算請求承載的檢查總和，並具有下列選項：

- when_supported – （預設） 當 操作在其服務模型中指定檢查總和演算法或需要請求檢查總和時，會計算請求承載檢查總和。
- when_required – 當 操作需要請求檢查總和，或使用者提供由 建模的時requestAlgorithmMember，會計算請求承載檢查總和 AWS 服務。

```
request_checksum_calculation = when_supported
```

環境變數 AWS_REQUEST_CHECKSUM_CALCULATION 會覆寫此設定。

response_checksum_validation

指定對回應承載執行檢查總和驗證的時間，並具有下列選項：

- when_supported – （預設） 當 操作在其 AWS CLI 支援的服務模型中指定回應演算法時，會執行回應承載檢查總和驗證。
- when_required – 當 操作在其 AWS CLI 支援的服務模型中指定回應演算法，而且您在操作輸入ENABLED中將模型設定為 requestValidationModeMember 時，就會執行回應承載檢查總和驗證。

```
response_checksum_validation = when_supported
```

環境變數 AWS_RESPONSE_CHECKSUM_VALIDATION 會覆寫此設定。

retry_mode

指定使用哪種重試模式 AWS CLI 。有三種重試模式可用：傳統 、標準 （預設） 和調整。如需有關重試的詳細資訊，請參閱 AWS CLI 中的重試 AWS CLI。

您可以使用 AWS_RETRY_MODE 環境變數來覆寫此數值。

```
retry_mode = standard
```

role_arn

指定您要用來執行 AWS CLI 命令之 IAM 角色的 Amazon Resource Name (ARN)。您還必須指定下列其中一個參數，以識別具有許可能擔任此角色的憑證：

- source_profile
- credential_source

```
role_arn = arn:aws:iam::123456789012:role/role-name
```

環境變數 AWS_ROLE_ARN 會覆寫此設定。

如需使用 Web 身分的詳細資訊，請參閱 the section called “擔任具有 Web 身分的角色”。

role_session_name

指定要連接到角色工作階段的名稱。此數值會在 RoleSessionName 呼叫 AWS CLI 操作時提供給 AssumeRole 參數，並成為所擔任角色使用者 ARN 的一部分：arn:aws:sts::*123456789012*:assumed-role/*role_name*/*role_session_name*。這是選擇性的參數。若您未提供此數值，將會自動產生工作階段名稱。此名稱會出現在與此工作階段相關聯之項目的 AWS CloudTrail 日誌中。

```
role_session_name = maria_garcia_role
```

環境變數 AWS_ROLE_SESSION_NAME 會覆寫此設定。

如需使用 Web 身分的詳細資訊，請參閱 the section called “擔任具有 Web 身分的角色”。

services

指定要用於設定檔的服務組態。

```
[profile dev-s3-specific-and-global]
endpoint_url = http://localhost:1234
services = s3-specific

[services s3-specific]
s3 =
  endpoint_url = http://localhost:4567
```

如需 services 區段的詳細資訊，請參閱 the section called “services”。

環境變數 AWS_ROLE_SESSION_NAME 會覆寫此設定。

如需使用 Web 身分的詳細資訊，請參閱 the section called “擔任具有 Web 身分的角色”。

sdk_ua_app_id

單一 AWS 帳戶 可供多個客戶應用程式用來呼叫 AWS 服務。應用程式 ID 會識別哪些來源應用程式使用 AWS 服務. AWS SDKs和 服務進行一組呼叫，不使用或解譯此值，而不是在客戶通訊中將其呈現。例如，此值可以包含在操作電子郵件中，以唯一識別哪些應用程式與通知相關聯。

應用程式 ID 是長度上限為 50 個字元的字串。允許使用字母、數字和下列特殊字元：預設！ $ % & * + - . , ^ _ ` | ~情況下，不會指派任何值。

```
sdk_ua_app_id = prod1
```

此設定可透過使用 AWS_SDK_UA_APP_ID 環境變數來覆寫。您無法將此數值設為命令列參數。

sigv4a_signing_region_set

指定使用逗號分隔清單與 SigV4a 簽署時要使用的區域。如果未設定此變數， AWS CLI 會使用使用的預設值 AWS 服務。如果 AWS 服務 沒有預設值，則請求簽章會在使用 值的所有區域中生效*。

```
sigv4a_signing_region_set = us-west-2, us-east-1
```

如需 SigV4a 的詳細資訊，請參閱《IAM 使用者指南》中的 AWS API 請求的簽章版本 4

此設定可透過使用 AWS_SIGV4A_SIGNING_REGION_SET 環境變數來覆寫。您無法將此數值設為命令列參數。

source_profile

指定具有長期憑證、 AWS CLI 可用來擔任您使用 role_arn 參數所指定角色的具名描述檔。您無法在同一個描述檔中同時指定 source_profile 和 credential_source。

```
source_profile = production-profile
```

sso_account_id

指定包含 IAM 角色 AWS 的帳戶 ID，其中包含您要授予相關聯 IAM Identity Center 使用者的許可。

此設定沒有環境變數或命令列選項。

```
sso_account_id = 123456789012
```

sso_region

指定 AWS 包含 AWS 存取入口網站主機的區域。這與預設的 CLI `region` 參數不同，且可以是與其不同的區域。

此設定沒有環境變數或命令列選項。

```
sso_region = us_west-2
```

sso_registration_scopes

要針對 `sso-session` 授權的逗號分隔清單範圍。範圍授權對 IAM Identity Center 承載字符授權端點的存取。有效範圍是一個字串，例如 `sso:account:access`。這個設定不適用於舊版不可重新整理的組態設定。

```
sso_registration_scopes = sso:account:access
```

sso_role_name

使用此設定檔時，指定定義使用者權限的 IAM 角色的易記名稱。

此設定沒有環境變數或命令列選項。

```
sso_role_name = ReadAccess
```

sso_start_url

指定指向組織 AWS 存取入口網站的 URL。 AWS CLI 使用此 URL 與 IAM Identity Center 服務建立工作階段，以驗證其使用者。若要尋找您的 AWS 存取入口網站 URL，請使用下列其中一項：

- 開啟您的邀請電子郵件，即會列出 AWS 存取入口網站 URL。
- 在 https：//https://console.aws.amazon.com/singlesignon/ 開啟 AWS IAM Identity Center 主控台。 AWS 存取入口網站 URL 會列在您的設定中。

此設定沒有環境變數或命令列選項。

```
sso_start_url = https://my-sso-portal.awsapps.com/start
```

use_dualstack_endpoint

允許使用雙堆疊端點傳送 AWS 請求。若要進一步了解支援 IPv4 和 IPv6 流量的雙堆疊端點，請參閱《Amazon Simple Storage Service 使用者指南》中的使用 Amazon S3 雙堆疊端點。雙堆疊端點適用於部分區域的某些服務。如果服務或 不存在雙堆疊端點 AWS 區域，請求會失敗。有效設定為 `true`和 `false`。此選項根據預設為停用。如需詳細資訊，請參閱the section called “設定為所有 AWS 服務使用雙堆疊端點”。

這與 `use_accelerate_endpoint` 設定互斥。

> **端點優先順序**
>
> 端點組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。會依特定順序 AWS CLI 檢查這些端點設定，並使用優先順序最高的端點設定。如需端點優先順序清單，請參閱 the section called “端點組態和設定優先順序”。

use_fips_endpoint

有些 AWS 服務提供端點，支援聯邦資訊處理標準 (FIPS) 140-2 AWS 區域。當 AWS 服務支援 FIPS 時，此設定會指定 AWS CLI 應使用的 FIPS 端點。與標準 AWS 端點不同，FIPS 端點使用符合 FIPS 140-2 的 TLS 軟體程式庫。會與美國政府互動的企業可能需要這些端點。如需詳細資訊，請參閱 the section called “設定為對所有 AWS 服務使用 FIP 端點”。

如果啟用此設定，但 服務不存在 FIPS 端點 AWS 區域，則 AWS 命令可能會失敗。在此情況下，請使用 `--endpoint-url` 選項或使用服務特定端點手動指定要在命令中使用的端點。

> **端點優先順序**
>
> 端點組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。會依特定順序 AWS CLI 檢查這些端點設定，並使用優先順序最高的端點設定。如需端點優先順序清單，請參閱 the section called “端點組態和設定優先順序”。

web_identity_token_file

指定檔案的路徑，其包含由身分提供者提供的 OAuth 2.0 存取字符或 OpenID Connect ID 字符。 AWS CLI 會載入此檔案的內容，並將其作為 `WebIdentityToken` 引數傳遞至 `AssumeRoleWithWebIdentity` 操作。

環境變數 AWS_WEB_IDENTITY_TOKEN_FILE 會覆寫此設定。

如需使用 Web 身分的詳細資訊，請參閱 the section called “擔任具有 Web 身分的角色”。

tcp_keepalive

指定 AWS CLI 用戶端是否使用 TCP 保持連線封包。

此項目沒有同等環境變數或命令列選項。

```
tcp_keepalive = false
```

S3 自訂命令設定

Amazon S3 支援數個設定，以設定 如何 AWS CLI 執行 Amazon S3 操作。有些適用於 s3api 和 s3 命名空間中的所有 S3 命令。其他則專用於 S3「自訂」命令，這些命令摘錄常見的操作，而不僅止於一對一映射到 API 操作。aws s3 傳輸命令 cp、sync、mv 和 rm 有額外設定，供您用來控制 S3 傳輸。

您可以在 config 檔案中指定 s3 巢狀設定，以設定所有這些選項。每個設定都各自一行縮排。

> **Note**
>
> 這些設定完全是選用。您無需設定任何這些設定，就應該能夠成功使用 aws s3 傳輸命令。提供這些設定可讓您基於效能而調整，或控制您執行這些 aws s3 命令的特定環境。

這些設定全部都在 config 檔案中的最上層 s3 索引鍵之下設定，如以下 development 描述檔的範例所示。

```
[profile development]
s3 =
  max_concurrent_requests = 20
  max_queue_size = 10000
  multipart_threshold = 64MB
  multipart_chunksize = 16MB
  max_bandwidth = 50MB/s
  use_accelerate_endpoint = true
  addressing_style = path
```

以下設定適用於 s3 或 s3api 命名空間中的任何 S3 命令。

addressing_style

指定要使用的定址樣式。這控制儲存貯體名稱是否位於主機名稱或為 URL 的一部分。有效值為：path、virtual 和 auto。預設值為 auto。

建構 Amazon S3 端點有兩種樣式。第一種稱為 virtual，並且在主機名稱中包含儲存貯體名稱。例如：https://*bucketname*.s3.amazonaws.com。或者，如果使用 path 樣式，則儲存貯體名稱就如同 URI 中的路徑，例如 https://s3.amazonaws.com/*bucketname*。CLI 中的預設值是使用 auto，將會儘可能嘗試使用 virtual 樣式，但在需要時會回復為 path 樣式。例如，如果您的儲存貯體名稱與 DNS 不相容，則儲存貯體名稱不能作為主機名稱的一部分，而必須在路徑中。使用 auto 時，CLI 會偵測到這個情況，並自動切換到 path 樣式。如果您將定址樣式設定為 path，則必須確保您在 中 AWS CLI 設定的 AWS 區域符合您儲存貯體的區域。

payload_signing_enabled

指定是否以 SHA256 簽署 sigv4 承載。在預設情況下，使用 HTTPS 來串流上傳時 (UploadPart 和 PutObject) 會停用此參數。根據預設，在串流上傳時 (UploadPart 和 PutObject)，這會設定為 false，但僅限於 ContentMD5 存在 (依預設會產生) 且端點使用 HTTPS 的情況。

如果設定為 true，S3 請求會收到 SHA256 檢查總和形式的額外內容驗證，這是為您計算並且包含在請求簽章中。如果設定為 false，則不計算檢查總和。停用此參數有助於降低檢查總和計算所造成的效能負荷。

use_accelerate_endpoint

對所有 s3 和 s3api 命令使用 Amazon S3 加速端點。預設值為 false。這與 use_dualstack_endpoint 設定互斥。

如果設為 true， 會將所有 Amazon S3 請求 AWS CLI 導向至位於 的S3 Accelerate端點s3-accelerate.amazonaws.com。若要使用這個端點，您必須啟用儲存貯體來使用 S3 Accelerate。所有請求都是使用虛擬樣式的儲存貯體定址所傳送：*my-bucket*.s3-accelerate.amazonaws.com。不會將任何 ListBuckets、CreateBucket 和 DeleteBucket 請求傳送至 S3 加速端點，因為該端點不支援這些操作。如果將任何 s3 或 s3api 命令的 --endpoint-url 參數設為 https://s3-accelerate.amazonaws.com 或 http://s3-accelerate.amazonaws.com，也可以設定此行為。

use_dualstack_endpoint

允許使用雙堆疊端點傳送 s3 和 s3api請求。若要進一步了解支援 IPv4 和 IPv6 流量的雙堆疊端點，請參閱《Amazon Simple Storage Service 使用者指南》中的使用 Amazon S3 雙堆疊端點。雙堆疊端點適用於部分區域的某些服務。如果服務或 不存在雙堆疊端點 AWS 區域，請求會失敗。有

效設定為 true和 false。此選項根據預設為停用。如需詳細資訊，請參閱the section called “設定為所有 AWS 服務使用雙堆疊端點”。

這與 use_accelerate_endpoint 設定互斥。

以下設定只適用於 s3 命名空間命令集的命令。

max_bandwidth

指定往返於 Amazon S3 上傳和下載資料時可消耗的最大頻寬。預設值是無限制。

這會限制 S3 命令在往返於 Amazon S3 傳輸資料時可使用的最大頻寬。這個數值僅適用於上傳和下載，不適用於複製或刪除。數值以每秒位元組數為單位。數值可以指定為：

- 整數。例如，1048576 將最大頻寬使用量設定為每秒 1 百萬位元組。
- 整數加上速率尾碼。您可以使用 KB/s、MB/s 或 GB/s 指定速率尾碼，例如 300KB/s 和 10MB/s。

一般而言，我們建議您先降低 max_concurrent_requests，以嘗試降低頻寬耗用量。如果這還無法將限制頻寬耗用量調到所需的速率，您可以使用 max_bandwidth 設定來進一步限制頻寬耗用量。這是因為 max_concurrent_requests 控制目前執行多少個執行緒。如果您先降低 max_bandwidth 但保留較高的 max_concurrent_requests 設置，則可能導致執行緒必須等待不必要的等待。這可能會導致過多的資源消耗和連線逾時。

max_concurrent_requests

指定並行請求數量上限。預設值為 10。

aws s3 傳輸命令是多執行緒。隨時可以執行多個 Amazon S3 請求。例如，當您使用 命令aws s3 cp localdir s3://bucket/ --recursive將檔案上傳至 S3 儲存貯體時， AWS CLI 可以localdir/file3平行上傳檔案 localdir/file1、 localdir/file2和 。設定 max_concurrent_requests 會指定可同時執行的傳輸操作數量上限。

由於幾個原因，您可能需要變更這個數值：

- 降低這個數值 – 在某些環境中，10 個並行請求的預設值可能拖垮系統。這可能導致連線逾時，或使系統的回應能力變慢。降低這個數值可讓 S3 傳輸命令耗用較少的資源。缺點是 S3 傳輸可能需要更久才會完成。如果您使用工具來限制頻寬，則可能需要降低這個數值。
- 提高這個數值 – 在某些情況下，您可能希望 Amazon S3 傳輸依需要儘量使用網路頻寬，以儘快完成。在這種情況下，預設的並行請求數量可能還不足以使用所有可用的網路頻寬。提高此數值可以縮短完成 Amazon S3 傳輸所花的時間。

`max_queue_size`

指定任務佇列中的任務數量上限。預設值為 1000。

AWS CLI 內部使用模型，其會排入佇列，然後由編號受限於 的消費者執行的 Amazon S3 任務`max_concurrent_requests`。任務通常會映射到單一 Amazon S3 操作。例如，任務可能是 `PutObjectTask`、`GetObjectTask` 或 `UploadPartTask`。任務新增到佇列的速率可比消費者完成任務的速率快很多。為了避免無限制成長，任務佇列大小會受限於特定大小。這個設定會變更該數量的上限值。

您通常不需要變更此設定。此設定也對應於 AWS CLI 知道需要執行的任務數量。這表示根據預設， AWS CLI 只能預見 1000 個任務。增加此值表示 AWS CLI 可以更快地知道所需的任務總數，假設佇列率比任務完成率更快。缺點是較大的 max_queue_size 需要更多記憶體。

`multipart_chunksize`

指定 AWS CLI 用於分段傳輸個別檔案的區塊大小。預設值為 8 MB，下限為 5 MB。

當檔案傳輸超過 `multipart_threshold` 時， AWS CLI 會將檔案分割成此大小的區塊。指定這個數值所使用的語法與 `multipart_threshold` 相同，包括以整數指定位元組數量，或使用大小和尾碼。

`multipart_threshold`

指定 AWS CLI 用於分段傳輸個別檔案的大小閾值。預設值為 8 MB。

當上傳、下載或複製檔案時，如果檔案超過該大小，Amazon S3 命令會切換到分段操作。您有兩種方式可以指定此數值：

- 以位元組為單位的檔案大小。例如 `1048576`。
- 附上大小尾碼的檔案大小。您可以使用 `KB`、`MB`、`GB` 或 `TB`，例如 `10MB` 和 `1GB`。

> **Note**
>
> 對於可用於分段操作的有效值，S3 可以施加限制。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 S3 分段上傳文件。

設定 的環境變數 AWS CLI

環境變數提供另一種指定組態選項和登入資料的方式，對於編寫指令碼很有用。

選項的優先順序

- 如果您使用本主題中所述的其中一個環境變數來指定選項，它將覆寫從組態檔中描述檔載入的任何數值。
- 如果您在 AWS CLI 命令列上使用 參數指定 選項，它會覆寫組態檔案中對應環境變數或設定檔的任何值。

如需優先順序以及 如何 AWS CLI 決定要使用哪些登入資料的詳細資訊，請參閱 設定 的設定 AWS CLI。

主題

- 如何設定環境變數
- AWS CLI 支援的環境變數

如何設定環境變數

下列範例說明如何為預設使用者設定環境變數。

Linux or macOS

```
$ export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
$ export AWS_DEFAULT_REGION=us-west-2
```

設定環境變數會變更使用的數值，直到 Shell 工作階段結束或直到您將該變數設為其他數值。您可以在 Shell 的啟動指令碼中設定變數，讓它們跨未來的工作階段持續生效。

Windows Command Prompt

為所有工作階段設定

```
C:\> setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
C:\> setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
C:\> setx AWS_DEFAULT_REGION us-west-2
```

使用 `setx` 設定環境變數時，將會變更在目前命令提示工作階段及您在執行命令後建立的所有命令提示工作階段中使用的數值。不會影響您執行命令當時已執行的其他命令 Shell。您可能需要重新啟動終端機才能載入設定。

僅針對目前工作階段設定

使用 `set` 設定環境變數會變更使用的數值，直到目前命令提示工作階段結束或直到您將該變數設為其他數值。

```
C:\> set AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
C:\> set AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
C:\> set AWS_DEFAULT_REGION=us-west-2
```

PowerShell

```
PS C:\> $Env:AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
PS C:\> $Env:AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
PS C:\> $Env:AWS_DEFAULT_REGION="us-west-2"
```

如果您在 PowerShell 提示字元中設定環境變數 (如前一個範例所示)，它只會在目前的工作階段儲存該數值。若要讓環境變數設定在所有 PowerShell 和命令提示字元工作階段中持續存在，請使用 Control Panel (控制面板) 中的 System (系統) 應用程式。或者，您也可以將變數新增到 PowerShell 設定檔，為所有未來 PowerShell 工作階段設定變數。如需有關存放環境變數或跨工作階段持續存放的詳細資訊，請參閱 PowerShell 文件。

如果您在 `config`或 `credentials` 檔案中設定了登入資料，您可以使用 `aws configure export-credentials`命令將其匯出為環境變數格式。

AWS CLI 支援的環境變數

AWS CLI 支援下列環境變數。

`AWS_ACCESS_KEY_ID`

指定與 IAM 帳戶相關聯的 AWS 存取金鑰。

如果已定義，此環境變數會覆寫設定檔設定 `aws_access_key_id` 的數值。您無法使用命令列選項來指定存取金鑰 ID。

`AWS_ACCOUNT_ID`

指定要用於對受支援 呼叫 AWS 的帳戶型端點 ID AWS 服務。如需帳戶型端點的詳細資訊，請參閱 the section called “帳戶型端點”。

此設定會覆寫aws_account_id設定。AWS_ACCOUNT_ID_ENDPOINT_MODE 環境變數或account_id_endpoint_mode設定必須設定為 preferred或 required，才能使用此設定。

> **端點優先順序**
>
> 端點組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。會依特定順序 AWS CLI 檢查這些端點設定，並使用優先順序最高的端點設定。如需端點優先順序清單，請參閱 the section called “端點組態和設定優先順序”。

AWS_ACCOUNT_ID_ENDPOINT_MODE

指定是否使用 AWS 帳戶型端點 IDs來呼叫支援的 AWS 服務。如需帳戶型端點的詳細資訊，請參閱 the section called “帳戶型端點”。

此設定可設定為下列項目：

- （預設）**preferred** – 如果可用，端點應包含帳戶 ID。
- **disabled** – 已解析的端點不包含帳戶 ID。
- **required** – 端點必須包含帳戶 ID。如果帳戶 ID 無法使用，軟體開發套件會擲回錯誤。

此設定會覆寫account_id_endpoint_mode設定。若要使用帳戶型端點，必須在AWS_ACCOUNT_ID環境變數或aws_account_id設定中設定 ID。

> **端點優先順序**
>
> 端點組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。會依特定順序 AWS CLI 檢查這些端點設定，並使用優先順序最高的端點設定。如需端點優先順序清單，請參閱 the section called “端點組態和設定優先順序”。

AWS_CA_BUNDLE

指定要用於 HTTPS 憑證驗證的憑證套件路徑。

如果已定義，此環境變數會覆寫設定檔設定 ca_bundle 的數值。您可以使用 --ca-bundle 命令列參數來覆寫此環境變數。

AWS_CLI_AUTO_PROMPT

啟用 第 2 AWS CLI 版的自動提示。有兩種設定可以使用：

- 在您每次嘗試執行 aws 命令時，**on** 都會使用完整的自動提示模式。這包括在完整的命令或不完整的命令之後按下 ENTER。
- **on-partial** 使用部分自動提示模式。如果命令不完整或因用戶端驗證錯誤而無法執行，則會使用自動提示。如果您有預先存在的指令碼、執行手冊，或者您只想要對不熟悉的命令自動提示，而不是在每個命令上提示，則此模式很有用。

如果已定義，此環境變數會覆寫 cli_auto_prompt 設定檔設定的數值。您可以使用 --cli-auto-prompt 及 --no-cli-auto-prompt 命令列參數來覆寫此環境變數。

如需第 2 AWS CLI 版自動提示功能的詳細資訊，請參閱 在 中啟用和使用命令提示 AWS CLI。

AWS_CLI_FILE_ENCODING

指定用於文字檔的編碼。預設情況下，編碼會與您的語言環境相符。若要設定與地區設定不同的編碼，請使用 aws_cli_file_encoding 環境變數。例如，如果您使用 Windows 的預設編碼 CP1252，則 aws_cli_file_encoding=UTF-8 設定會將 CLI 設定為使用 UTF-8。

AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS

如果使用 s3 mv命令時來源和目的地儲存貯體相同，可以將來源檔案或物件移至其本身，這可能會導致意外刪除來源檔案或物件。AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS 環境變數和--validate-same-s3-paths選項指定是否要驗證 Amazon S3 來源或目的地 URIs 中的存取點 ARNs 或存取點別名。

> **Note**
>
> 的路徑驗證s3 mv需要額外的 API 呼叫。

AWS_CONFIG_FILE

指定 AWS CLI 用來存放組態描述檔的檔案位置。預設路徑為 ~/.aws/config。

您無法在具名設定檔設定中或使用命令列參數來指定此數值。

AWS_DATA_PATH

在載入 AWS CLI 資料~/.aws/models時，要在 的內建搜尋路徑之外檢查的其他目錄清單。設定此環境變數會指出回復內建搜尋路徑前需要先檢查的額外目錄。多個項目應使用 os.pathsep 字元分隔，在 Linux 或 macOS 上為 :，在 Windows 上則為 ;。

AWS_DEFAULT_OUTPUT

指定要使用的輸出格式。

如果已定義，此環境變數會覆寫設定檔設定 output 的數值。您可以使用 --output 命令列參數來覆寫此環境變數。

AWS_DEFAULT_REGION

根據預設， 會Default region name識別您要將請求傳送到其伺服器的 AWS 區域。這通常是最接近您的區域，但它可以是任何區域。例如，您可以輸入 us-west-2 來使用美國西部 (奧勒岡)。除非您在個別命令中另外指定，否則此為所有後續請求傳送到的區域。

> **Note**
>
> 使用 時 AWS CLI，您必須明確或透過設定預設 AWS 區域來指定 區域。如需可用區域的清單，請參閱區域和端點。所使用的區域指標 AWS CLI 與您在 AWS Management Console URLs和服務端點中看到的名稱相同。

如果已定義，此環境變數會覆寫設定檔設定 region 的數值。您可以使用--region命令列參數和 AWS SDK 相容環境變數來覆寫此AWS_REGION環境變數。

AWS_EC2_METADATA_DISABLED

停用 Amazon EC2 執行個體中繼資料服務 (IMDS)。

如果設定為 true，則不會向 IMDS 要求使用者憑證或組態 (如區域)。

AWS_ENDPOINT_URL

指定用於所有服務要求的端點。如需詳細資訊，請參閱the section called “設定所有 的全域端點 AWS 服務”。

> **端點優先順序**
>
> 端點組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。會依特定順序 AWS CLI 檢查這些端點設定，並使用優先順序最高的端點設定。如需端點優先順序清單，請參閱 the section called “端點組態和設定優先順序”。

AWS_ENDPOINT_URL_<SERVICE>

指定用於特定服務的自訂端點，其中 <SERVICE> 以 AWS 服務 識別符取代。例如，Amazon DynamoDB 的 serviceId DynamoDB。對於此服務，端點 URL 環境變數為 AWS_ENDPOINT_URL_DYNAMODB。

如需所有服務特定環境變數的清單，請參閱 服務特定識別碼清單。

> **端點優先順序**
>
> 端點組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。會依特定順序 AWS CLI 檢查這些端點設定，並使用優先順序最高的端點設定。如需端點優先順序清單，請參閱 the section called “端點組態和設定優先順序”。

AWS_IGNORE_CONFIGURED_ENDPOINT_URLS

如果啟用， 會 AWS CLI 忽略所有自訂端點組態。有效值為 **true** 和 **false**。如需詳細資訊，請參閱the section called “設定所有 的全域端點 AWS 服務”。

端點組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。如需端點優先順序，請參閱 the section called “端點組態和設定優先順序”。

> **端點優先順序**
>
> 端點組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。會依特定順序 AWS CLI 檢查這些端點設定，並使用優先順序最高的端點設定。如需端點優先順序清單，請參閱 the section called “端點組態和設定優先順序”。

AWS_MAX_ATTEMPTS

指定重試 AWS CLI 處理常式使用的最大重試次數值，其中初始呼叫會計入您提供的值。如需有關重試的詳細資訊，請參閱 AWS CLI 中的重試 AWS CLI。

如果已定義，此環境變數會覆寫設定檔設定 `max_attempts` 的數值。

AWS_METADATA_SERVICE_NUM_ATTEMPTS

嘗試在已設定 IAM 角色的 Amazon EC2 執行個體上擷取登入資料時， 會 AWS CLI 嘗試在停止之前從執行個體中繼資料服務擷取登入資料一次。如果您知道自己的程式碼將在 Amazon EC2 執行個體上執行，您可以增加這個值，讓 AWS CLI 在放棄之前多重新嘗試幾次。

AWS_METADATA_SERVICE_TIMEOUT

執行個體中繼資料服務連線逾時前的秒數。嘗試在已設定 IAM 角色的 Amazon EC2 執行個體上擷取憑證時，執行個體中繼資料服務連線預設為 1 秒之後逾時。如果您知道自己在已設定 IAM 角色的 Amazon EC2 執行個體上執行，您可以視需要增加這個值。

AWS_PAGER

指定用於輸出的分頁程式。根據預設，第 2 AWS CLI 版會透過作業系統的預設分頁程式傳回所有輸出。

若要停用所有外部分頁程式，請將變數設定為空字串。

如果已定義，此環境變數會覆寫設定檔設定 cli_pager 的數值。

AWS_PROFILE

指定 AWS CLI 設定檔的名稱，其中包含要使用的登入資料和選項。這可以是在 credentials 或 config 檔案中存放的描述檔名稱，或數值 default 以使用預設描述檔。

如果您定義此環境變數，它將覆寫使用組態檔中名為 [default] 之描述檔的行為。您可以使用 --profile 命令列參數來覆寫此環境變數。

AWS_REGION

開發套件 AWS AWS 相容的環境變數，指定要傳送請求的 區域。

如果已定義，此環境變數會覆寫環境變數 AWS_DEFAULT_REGION 及設定檔設定 region 的數值。您可以使用 --region 命令列參數來覆寫此環境變數。

AWS_REQUEST_CHECKSUM_CALCULATION

指定何時計算請求承載的檢查總和，並具有下列選項：

- when_supported – （預設） 當 操作在其服務模型中指定檢查總和演算法或需要請求檢查總和時，會計算請求承載檢查總和。
- when_required – 當 操作需要請求檢查總和，或當使用者提供由 建模的時requestAlgorithmMember，會計算請求承載檢查總和 AWS 服務。

如果已定義，此環境變數會覆寫設定檔設定 request_checksum_calculation 的數值。

AWS_RESPONSE_CHECKSUM_VALIDATION

指定對回應承載執行檢查總和驗證的時間，並具有下列選項：

- when_supported – （預設） 當 操作在其 AWS CLI 支援的服務模型中指定回應演算法時，會執行回應承載檢查總和驗證。
- when_required – 當 操作在其 AWS CLI 支援的服務模型中指定回應演算法，而且您在操作輸入ENABLED中將模型設定為 requestValidationModeMember 時，就會執行回應承載檢查總和驗證。

如果已定義，此環境變數會覆寫設定檔設定 response_checksum_validation 的數值。

AWS_RETRY_MODE

指定使用哪種重試模式 AWS CLI。有三種重試模式可用：傳統、標準（預設）和調整。如需有關重試的詳細資訊，請參閱 AWS CLI 中的重試 AWS CLI。

如果已定義，此環境變數會覆寫設定檔設定 `retry_mode` 的數值。

AWS_ROLE_ARN

使用您要用來執行 AWS CLI 命令的 Web 身分提供者，指定 IAM 角色的 Amazon Resource Name (ARN)。

與 `AWS_WEB_IDENTITY_TOKEN_FILE` 及 `AWS_ROLE_SESSION_NAME` 環境變數搭配使用。

如果已定義，此環境變數會覆寫設定檔設定 `role_arn` 的數值。您無法將角色工作階段名稱指定為命令列參數。

> **Note**
>
> 此環境變數目前僅適用於具有 Web 身分提供者的擔任角色，並不適用一般擔任角色提供者組態。

如需使用 Web 身分的詳細資訊，請參閱 the section called “擔任具有 Web 身分的角色”。

AWS_ROLE_SESSION_NAME

指定要連接到角色工作階段的名稱。此值會在 AWS CLI 呼叫 `AssumeRole`操作時提供給 `RoleSessionName` 參數，並成為擔任角色使用者 ARN 的一部分：`arn:aws:sts::123456789012:assumed-role/role_name/role_session_name`。這是選擇性的參數。若您未提供此數值，將會自動產生工作階段名稱。此名稱會出現在與此工作階段相關聯的項目 AWS CloudTrail 日誌中。

如果已定義，此環境變數會覆寫設定檔設定 `role_session_name` 的數值。

與 `AWS_ROLE_ARN` 及 `AWS_WEB_IDENTITY_TOKEN_FILE` 環境變數搭配使用。

如需使用 Web 身分的詳細資訊，請參閱 the section called “擔任具有 Web 身分的角色”。

> **Note**
>
> 此環境變數目前僅適用於具有 Web 身分提供者的擔任角色，並不適用一般擔任角色提供者組態。

AWS_SDK_UA_APP_ID

單一 AWS 帳戶 可供多個客戶應用程式用來呼叫 AWS 服務。應用程式 ID 會識別哪些來源應用程式使用 AWS 服務. AWS SDKs和 服務進行一組呼叫，不使用或解譯此值，而不是在客戶通訊中將其呈現。例如，此值可以包含在操作電子郵件中，以唯一識別哪些應用程式與通知相關聯。

根據預設，沒有值。

應用程式 ID 是長度上限為 50 個字元的字串。允許使用字母、數字和下列特殊字元：

```
! $ % & * + - . , ^ _ ` | ~
```

如果已定義，此環境變數會覆寫設定檔設定 sdk_ua_app_id 的數值。您無法將應用程式 ID 指定為命令列選項。

AWS_SECRET_ACCESS_KEY

指定與存取金鑰相關聯的私密金鑰。這基本上是存取金鑰的「密碼」。

如果已定義，此環境變數會覆寫設定檔設定 aws_secret_access_key 的數值。您不能將私密存取金鑰 ID 指定為命令列選項。

AWS_SESSION_TOKEN

指定當您直接從 AWS STS 作業擷取的暫時安全憑證時需要的工作階段字符值。如需詳細資訊，請參閱AWS CLI 命令參考中的擔任角色命令的輸出部分。

如果已定義，此環境變數會覆寫設定檔設定 aws_session_token 的數值。

AWS_SHARED_CREDENTIALS_FILE

指定 AWS CLI 用來存放存取金鑰的檔案位置。預設路徑為 ~/.aws/credentials。

您無法在具名設定檔設定中或使用命令列參數來指定此數值。

AWS_SIGV4A_SIGNING_REGION_SET

指定使用逗號分隔清單與 SigV4a 簽署時要使用的區域。如果未設定此變數， AWS CLI 會使用使用的預設值 AWS 服務。如果 AWS 服務 沒有預設值，則請求簽章會在使用 值的所有區域中生效*。

如需 SigV4a 的詳細資訊，請參閱《IAM 使用者指南》中的 AWS API 請求的簽章版本 4

如果已定義，此環境變數會覆寫設定檔設定 `sigv4a_signing_region_set` 的數值。

AWS_USE_DUALSTACK_ENDPOINT

允許使用雙堆疊端點傳送 AWS 請求。若要進一步了解支援 IPv4 和 IPv6 流量的雙堆疊端點，請參閱《Amazon Simple Storage Service 使用者指南》中的使用 Amazon S3 雙堆疊端點。雙堆疊端點適用於部分區域的某些服務。如果服務或 不存在雙堆疊端點 AWS 區域，請求會失敗。此選項根據預設為停用。如需詳細資訊，請參閱the section called “設定為所有 AWS 服務使用雙堆疊端點”。

> **端點優先順序**
>
> 端點組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。會依特定順序 AWS CLI 檢查這些端點設定，並使用優先順序最高的端點設定。如需端點優先順序清單，請參閱 the section called “端點組態和設定優先順序”。

AWS_USE_FIPS_ENDPOINT

有些 AWS 服務提供端點，支援聯邦資訊處理標準 (FIPS) 140-2 AWS 區域。 AWS 服務支援 FIPS 時，此設定會指定 AWS CLI 應使用的 FIPS 端點。與標準 AWS 端點不同，FIPS 端點使用符合 FIPS 140-2 的 TLS 軟體程式庫。會與美國政府互動的企業可能需要這些端點。如需詳細資訊，請參閱 the section called “設定為對所有 AWS 服務使用 FIP 端點”。

如果啟用此設定，但 服務不存在 FIPS 端點 AWS 區域，則 AWS 命令可能會失敗。在此情況下，請使用 `--endpoint-url` 選項或使用服務特定端點手動指定要在命令中使用的端點。

> **端點優先順序**
>
> 端點組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。會依特定順序 AWS CLI 檢查這些端點設定，並使用優先順序最高的端點設定。如需端點優先順序清單，請參閱 the section called “端點組態和設定優先順序”。

AWS_WEB_IDENTITY_TOKEN_FILE

指定檔案的路徑，其包含由身分提供者提供的 OAuth 2.0 存取字符或 OpenID Connect ID 字符。 AWS CLI 會載入此檔案的內容，並將其作為 `WebIdentityToken` 引數傳遞至 `AssumeRoleWithWebIdentity` 操作。

與 `AWS_ROLE_ARN` 及 `AWS_ROLE_SESSION_NAME` 環境變數搭配使用。

如果已定義，此環境變數會覆寫設定檔設定 `web_identity_token_file` 的數值。

如需使用 Web 身分的詳細資訊，請參閱 the section called “擔任具有 Web 身分的角色”。

> **Note**
>
> 此環境變數目前僅適用於具有 Web 身分提供者的擔任角色，並不適用一般擔任角色提供者組態。

中的命令列選項 AWS CLI

在 中 AWS CLI，命令列選項是全域參數，可用來覆寫該單一命令的預設組態設定、任何對應的設定檔設定或環境變數設定。您無法使用命令列選項來直接指定憑證，但可指定要使用的描述檔。

主題

- 如何使用命令列選項
- AWS CLI 支援的全域命令列選項
- 命令列選項的常見用途

如何使用命令列選項

大多數命令列選項都是簡單的字串，例如下列範例中的設定檔名稱 `profile1`：

```
$ aws s3 ls --profile profile1
amzn-s3-demo-bucket1
amzn-s3-demo-bucket2
...
```

每個接受引數的選項需要有空格或等號 (=) 來隔開引數與選項名稱。如果引數值是包含空格的字串，則您必須使用引號括住引數。如需關於參數類型和參數格式的詳細資訊，請參閱 在 中指定參數值 AWS CLI。

AWS CLI 支援的全域命令列選項

在 中 AWS CLI ，您可以使用下列命令列選項來覆寫該單一命令的預設組態設定、任何對應的設定檔設定或環境變數設定。

--ca-bundle ***<string>***

指定在驗證 SSL 憑證時使用的憑證授權機構 (CA) 憑證套件。

如果已定義，此選項會覆寫設定檔設定 ca_bundle 的數值及 AWS_CA_BUNDLE 環境變數。

--cli-auto-prompt

啟用單一命令的自動提示模式。如下列範例所示，您隨時都能指定它。

```
$ aws --cli-auto-prompt
$ aws dynamodb --cli-auto-prompt
$ aws dynamodb describe-table --cli-auto-prompt
```

此選項會覆寫 aws_cli_auto_prompt 環境變數及 cli_auto_prompt 設定檔設定。

如需第 2 AWS CLI 版自動提示功能的詳細資訊，請參閱 在 中啟用和使用命令提示 AWS CLI。

--cli-binary-format

指定 AWS CLI 版本 2 如何解譯二進位輸入參數。它可能是以下其中一個數值：

- base64 – 這是預設值。以二進位大型物件 (BLOB) 輸入的輸入參數接受 base64 編碼的字串。若要傳遞實際的二進位內容，請將內容放置在檔案中，並在參數值中使用 `fileb://` 前綴，以提供檔案路徑和名稱。若要傳遞包含在檔案中的 base64 編碼文字，請在參數值中使用 `file://` 前綴，以提供檔案路徑和名稱。
- raw-in-base64-out — AWS CLI 第 1 版的預設值。如果設定的值為 `raw-in-base64-out`，則使用 `file://` 字首參考的檔案會讀取為文字，然後 AWS CLI 嘗試將其編碼為二進位。

這會覆寫 cli_binary_format 檔案組態設定。

```
$ aws lambda invoke \
    --cli-binary-format raw-in-base64-out \
    --function-name my-function \
    --invocation-type Event \
    --payload '{ "name": "Bob" }' \
    response.json
```

如果您使用`fileb://`字首表示法參考檔案中的二進位值， AWS CLI 一律會預期檔案包含原始二進位內容，而不會嘗試轉換該值。

如果您使用`file://`字首表示法參考檔案中的二進位值， 會根據目前的`cli_binary_format`設定 AWS CLI 處理檔案。如果該設定的值為 `base64`（未明確設定時為預設值），則 AWS CLI 預期

檔案包含 base64 編碼的文字。如果該設定值為 `raw-in-base64-out`，AWS CLI 預期檔案會包含原始二進位內容。

--cli-connect-timeout ***<integer>***

指定通訊端連線時間上限 (以秒為單位)。如果數值設為零 (0)，通訊端連線會無限期等待 (凍結) 且不逾時。預設值為 60 秒。

--cli-read-timeout ***<integer>***

指定通訊端讀取時間上限 (以秒為單位)。如果數值設為零 (0)，通訊端讀取會無限期等待 (凍結) 且不逾時。預設值為 60 秒。

--color ***<string>***

指定支援彩色輸出。有效值為 on、off 和 auto。預設值為 auto。

--debug

啟用偵錯記錄的布林參數。 AWS CLI 根據預設， 在命令輸出中提供有關命令結果的任何成功或失敗的清除資訊。此 `--debug` 選項會提供完整的 Python 記錄。這包括命令操作的額外 `stderr` 診斷資訊，在分析解決命令為什麼產生意外結果時很有用。為了方便檢視偵錯記錄，我們建議將記錄傳送至檔案，以便更輕鬆地搜尋資訊。您可以使用下列其中一種來執行這項作業。

若要僅傳送 `stderr` 診斷資訊，請附加 `2> debug.txt`，其中 `debug.txt` 是您想要用於偵錯檔案的名稱：

```
$ aws servicename commandname options --debug 2> debug.txt
```

若要同時傳送輸出及 `stderr` 診斷資訊，請附加 `&> debug.txt`，其中 `debug.txt` 是您想要用於偵錯檔案的名稱：

```
$ aws servicename commandname options --debug &> debug.txt
```

--endpoint-url ***<string>***

指定請求送往的 URL。對於大多數命令， 會根據選取的服務和指定的 AWS 區域 AWS CLI 自動判斷 URL。但部分命令需要您指定帳戶專屬的 URL。您也可以設定一些 AWS 服務直接在私有 VPC 中託管端點，然後可能需要指定。

下列命令範例使用自訂的 Amazon S3 端點 URL。

```
$ aws s3 ls --endpoint-url http://localhost:4567
```

> **端點優先順序**
>
> 端點組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。會依特定順序 AWS CLI 檢查這些端點設定，並使用優先順序最高的端點設定。如需端點優先順序清單，請參閱 the section called “端點組態和設定優先順序”。

--no-cli-auto-prompt

停用單一命令的自動提示模式。

```
$ aws dynamodb describe-table --table-name Table1 --no-cli-auto-prompt
```

此選項會覆寫 aws_cli_auto_prompt 環境變數及 cli_auto_prompt 設定檔設定。

如需第 2 AWS CLI 版自動提示功能的詳細資訊，請參閱 在 中啟用和使用命令提示 AWS CLI。

--no-cli-pager

為命令停用輸出之分頁程式的布林參數。

--no-paginate

布林值開關，可停用 自動 AWS CLI 進行的多個呼叫，以接收建立輸出分頁的所有命令結果。這表示只會顯示輸出的第一頁。

--no-sign-request

停用對 AWS 服務端點簽署 HTTP 請求的布林值開關。這可防止載入憑證。

--no-verify-ssl

根據預設， AWS CLI 會在與 AWS 服務通訊時使用 SSL。對於每個 SSL 連線和呼叫， AWS CLI 會驗證 SSL 憑證。使用此選項會覆寫驗證 SSL 憑證的預設行為。

> **Warning**
>
> 此選項不是最佳作法。如果您使用 `--no-verify-ssl`，用戶端 AWS 和服務之間的流量將不再受到保護。這意味著您的流量存在安全風險，並且容易受到攔截式攻擊。如果您在憑證上遇到問題，最好改為解決這些問題。如需憑證疑難排解步驟，請參閱 the section called “SSL 憑證錯誤”。

--output ***<string>***

指定要用於此命令的輸出格式。您可以指定下列任何數值：

- **`json`** – 輸出的格式為 JSON 字串。
- **`yaml`** – 輸出的格式為 YAML 字串。
- **`yaml-stream`** – 輸出採用串流方式且格式為 YAML 字串。串流可加速處理大型資料類型。
- **`text`** – 輸出的格式是多行以 Tab 分隔的字串值。這對於將輸出傳遞給文字處理器 (如 `grep`、`sed` 或 `awk`) 非常有用。
- **`table`** – 輸出的格式為使用字元 +|- 形成儲存格框線的表格。它通常以「方便人類使用」的格式來呈現資訊，這種格式比其他格式更容易閱讀，但在編寫程式方面較不有用。

--profile ***<string>***

指定要用於此命令的具名描述檔。若需設置額外的具名描述檔，您可以使用 `aws configure` 命令搭配 `--profile` 選項。

```
$ aws configure --profile <profilename>
```

--query ***<string>***

指定 JMESPath 查詢以用於篩選回應資料。如需詳細資訊，請參閱 在 中篩選輸出 AWS CLI。

--region ***<string>***

指定 AWS 傳送此命令 AWS 請求的區域。如需您可指定之所有區域和端點的清單，請參閱 Amazon Web Services 一般參考 中的 AWS 區域與端點。

--version

布林值開關，顯示正在執行之 AWS CLI 程式的目前版本。

命令列選項的常見用途

命令列選項的常見用途包括在多個 AWS 區域中查看您的資源，或是於編寫指令碼時變更輸出格式以利易讀性與使用便利。在下列範例中，我們執行 describe-instances 命令，直到我們找到執行個體所在的區域。

```
$ aws ec2 describe-instances --output table --region us-west-1
-------------------
|DescribeInstances|
+-----------------+
```

```
$ aws ec2 describe-instances --output table --region us-west-2
-------------------------------------------------------------------------------
|                            DescribeInstances                                |
+-----------------------------------------------------------------------------+
||                              Reservations                                 ||
|+-----------------------------------+-----------------------------------+|
||  OwnerId                          |  012345678901                     ||
||  ReservationId                    |  r-abcdefgh                       ||
|+-----------------------------------+-----------------------------------+|
|||                               Instances                               |||
||+-------------------------+-------------------------------------------+||
|||  AmiLaunchIndex         |  0                                        |||
|||  Architecture           |  x86_64                                   |||
...
```

在 中設定命令完成 AWS CLI

AWS Command Line Interface (AWS CLI) 包含與 bash 相容的命令完成功能，可讓您使用 Tab 鍵來完成部分輸入的命令。您需要在大多數系統上手動設定此功能。

如需 第 2 AWS CLI 版自動提示功能的資訊，請參閱 在 中啟用和使用命令提示 AWS CLI。

主題

- 運作方式
- 在 Linux 或 macOS 上設定命令完成
- 在 Windows 上設定命令完成

運作方式

當您輸入命令、參數或選項的部分內容時，命令完成功能會自動完成您的命令，或顯示建議的命令清單。若要提示命令完成，請輸入命令的部分內容，然後按下完成鍵 (在大部分 Shell 中通常為 *Tab*)。

下列範例示範命令完成功能的不同使用方式：

- 輸入命令的部分內容，然後按 *Tab* 以顯示建議的命令清單。

```
$ aws dynamodb dTAB
delete-backup                       describe-global-table
delete-item                         describe-global-table-settings
delete-table                        describe-limits
```

```
describe-backup                    describe-table
describe-continuous-backups        describe-table-replica-auto-scaling
describe-contributor-insights      describe-time-to-live
describe-endpoints
```

- 輸入參數的部分內容，然後按 *Tab* 以顯示建議的參數清單。

```
$ aws dynamodb delete-table --TAB
--ca-bundle              --endpoint-url           --profile
--cli-connect-timeout    --generate-cli-skeleton  --query
--cli-input-json         --no-paginate            --region
--cli-read-timeout       --no-sign-request        --table-name
--color                  --no-verify-ssl          --version
--debug                  --output
```

- 輸入參數，然後按 *Tab* 以顯示建議的資源值清單。此功能僅適用於 第 2 AWS CLI 版。

```
$ aws dynamodb delete-table --table-name TAB
Table 1                  Table 2                  Table 3
```

在 Linux 或 macOS 上設定命令完成

若要在 Linux 或 macOS 上設定命令完成，您必須知道正在使用的 Shell 名稱及 `aws_completer` 指令碼的位置。

> **Note**
>
> 在執行 Amazon Linux 的 Amazon EC2 執行個體上，預設會自動設定和啟用命令完成。

主題

- 確認完成標籤的資料夾位於您的路徑中
- 啟用命令完成
- 驗證命令完成

確認完成標籤的資料夾位於您的路徑中

若要讓 AWS 完成器成功運作， `aws_completer` 必須在 shell 的路徑中。`which` 命令可以檢查完成標籤是否位於您的路徑中。

```
$ which aws_completer
/usr/local/bin/aws_completer
```

如果該命令找不到完成標籤，請使用下列步驟將完成標籤的資料夾新增到您的路徑中。

步驟 1：尋找 AWS 完成者

AWS 完成器的位置可能會因使用的安裝方法而有所不同。

- Package Manager - 如 pip、brew、 yum和 等程式apt-get通常會將 AWS 完成器 （或與其的符號連結） 安裝到標準路徑位置。
  - 如果您使用的是不含 --user 參數的 pip，預設路徑為 /usr/local/bin/aws_completer。
  - 如果您使用的是包含 --user 參數的 pip，預設路徑為 /home/*username*/.local/bin/aws_completer。
- Bundled Installer – 如果您使用的是 Bundled Installer，預設路徑為 /usr/local/bin/aws_completer。

如果所有其他失敗，您可以使用 find命令來搜尋檔案系統是否有 AWS 完成程式。

```
$ find / -name aws_completer
/usr/local/bin/aws_completer
```

步驟 2：識別您的 Shell

若要識別您使用的是哪一種 Shell，您可以利用下列其中一個命令。

- echo $SHELL – 會顯示 Shell 的程式檔案名稱。這通常符合使用中的 Shell 名稱，除非您在登入之後啟動了不同的 Shell。

  ```
  $ echo $SHELL
  /bin/bash
  ```

- ps – 會顯示目前使用者正在執行的程序。其中一個是 Shell。

  ```
  $ ps
    PID TTY          TIME CMD
   2148 pts/1    00:00:00 bash
   8756 pts/1    00:00:00 ps
  ```

步驟 3：將完成標籤新增到您的路徑

1. 在您的使用者資料夾中尋找 Shell 的描述檔指令碼。

```
$ ls -a ~/
.  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
```

   - Bash – .bash_profile、.profile 或 .bash_login
   - Zsh – .zshrc
   - Tcsh – .tcshrc、.cshrc 或 .login

2. 在設定檔尾端新增匯出命令，類似於以下範例。將 */usr/local/bin/* 換成您在上一節探索到的資料夾。

```
export PATH=/usr/local/bin/:$PATH
```

3. 將描述檔重新載入到目前的工作階段，讓這些變更生效。將 *.bash_profile* 換成您在第一節探索到的 Shell 指令碼的名稱。

```
$ source ~/.bash_profile
```

啟用命令完成

在確認完成標籤位於您的路徑中後，請為您正在使用的 Shell 執行適當的命令，從而啟用命令完成。您可以新增命令到 Shell 的設定檔，以在每次開啟新 Shell 時執行該檔案。在每個命令中，將 */usr/local/bin/* 路徑取代為 確認完成標籤的資料夾位於您的路徑中 中在您系統上找到的路徑。

- **bash** - 使用內建命令 complete。

```
$ complete -C '/usr/local/bin/aws_completer' aws
```

  新增先前的命令到 ~/.bashrc，以在每次開啟新 Shell 時執行該檔案。您的 ~/.bash_profile 應指定 ~/.bashrc 的來源，以確保命令也會在登入 Shell 中執行。

- **zsh** – 若要執行命令完成，您必須將下列自動載入行加到 ~/.zshrc 設定檔指令碼末尾，以執行 bashcompinit。

```
$ autoload bashcompinit && bashcompinit
$ autoload -Uz compinit && compinit
```

若要啟用命令完成，請使用內建命令 `complete`。

```
$ complete -C '/usr/local/bin/aws_completer' aws
```

新增先前的命令到 `~/.zshrc`，以在每次開啟新 Shell 時執行該檔案。

- **tcsh** - tcsh 的完成採用文字類型及模式來定義完成行為。

```
> complete aws 'p/*/`aws_completer`/'
```

新增先前的命令到 `~/.tcshrc`，以在每次開啟新 Shell 時執行該檔案。

啟用命令完成後，[驗證命令完成](#) 開始運作。

驗證命令完成

啟用命令完成之後，請重新載入您的 Shell、輸入命令的部分文字並按下 Tab，即可查看可用的命令。

```
$ aws sTAB
s3              ses             sqs             sts             swf
s3api           sns             storagegateway  support
```

在 Windows 上設定命令完成

> **Note**
>
> 有關 PowerShell 如何處理完成的資訊，包括完成時使用的各種操作鍵，請參閱 Microsoft PowerShell 文件中的 about_Tab_Expansion。

若要在 Windows 上為 PowerShell 啟用命令完成，請在 PowerShell 中完成下列步驟。

1. 使用下列命令開啟您的 `$PROFILE`。

```
PS C:\> Notepad $PROFILE
```

如果您沒有 `$PROFILE`，請使用下列命令建立使用者描述檔。

```
PS C:\> if (!(Test-Path -Path $PROFILE ))
{ New-Item -Type File -Path $PROFILE -Force }
```

如需 PowerShell 設定檔的詳細資訊，請參閱 Microsoft Docs 網站上的如何在 Windows PowerShell ISE 中使用設定檔。

2. 若要啟用命令完成，請將下列程式碼區塊新增至您的設定檔，然後儲存並關閉檔案。

```
Register-ArgumentCompleter -Native -CommandName aws -ScriptBlock {
    param($commandName, $wordToComplete, $cursorPosition)
        $env:COMP_LINE=$wordToComplete
        if ($env:COMP_LINE.Length -lt $cursorPosition){
            $env:COMP_LINE=$env:COMP_LINE + " "
        }
        $env:COMP_POINT=$cursorPosition
        aws_completer.exe | ForEach-Object {
            [System.Management.Automation.CompletionResult]::new($_, $_,
 'ParameterValue', $_)
        }
        Remove-Item Env:\COMP_LINE
        Remove-Item Env:\COMP_POINT
}
```

3. 啟用命令完成之後，請重新載入您的 Shell、輸入命令的部分文字並按下 Tab，以循環查看可用的命令。

```
$ aws sTab
```

```
$ aws s3
```

若要查看完成結果適用的所有可用命令，請輸入部分命令，然後按 Ctrl + Space。

```
$ aws sCtrl + Space
s3              ses             sqs             sts             swf
s3api           sns             storagegateway  support
```

AWS CLI 中的重試 AWS CLI

本主題說明 可能會如何 AWS CLI 看到 AWS 服務呼叫因非預期問題而失敗。這些問題可能會發生在伺服器端，也可能會因為您嘗試呼叫的 AWS 服務受到速率限制而失敗。這些種類的失敗通常不需要特殊處理，因為通常在短暫的等待時間後會自動再次進行呼叫。 AWS CLI 提供許多功能，可協助用戶端在遇到這類錯誤或例外狀況時重試對 AWS 服務的呼叫。

主題

- 可用的重試模式
- 設定重試模式
- 檢視重試嘗試的記錄

可用的重試模式

AWS CLI 有多種模式可供選擇，取決於您的版本：

- 傳統重試模式
- 標準重試模式
- 自適應重試模式

傳統重試模式

傳統模式使用較舊的重試處理常式，其有限的功能包含：

- 預設值 4 表示的是重試嘗試次數上限，總共允許嘗試 5 次呼叫。此數值可透過 `max_attempts` 組態參數加以覆寫。
- DynamoDB 中的預設值 9 表示的是重試嘗試次數上限，總共允許嘗試 10 次呼叫。此數值可透過 `max_attempts` 組態參數加以覆寫。
- 適用於下列有限數量的錯誤/例外狀況的重試嘗試次數：
  - 一般通訊埠/連線錯誤：
    - `ConnectionError`
    - `ConnectionClosedError`
    - `ReadTimeoutError`
    - `EndpointConnectionError`
  - 服務端調節/限制錯誤及例外狀況：

    - Throttling
    - ThrottlingException
    - ThrottledException
    - RequestThrottledException
    - ProvisionedThroughputExceededException
- 數個 HTTP 狀態碼 (包括 429、500、502、503、504 及 509) 上的重試嘗試次數。
- 任何重試嘗試都會包含基本係數為 2 的指數退避。

標準重試模式

標準模式是跨 AWS SDKs，其功能比舊版更多。此模式是 AWS CLI 第 2 版的預設模式。標準模式是針對第 2 AWS CLI 版建立，並回溯到第 1 AWS CLI 版。標準模式的功能包含：

- 預設值 2 表示的是重試嘗試次數上限，總共允許嘗試 3 次呼叫。此數值可透過 max_attempts 組態參數加以覆寫。
- 適用於下列展開清單中錯誤/例外狀況的重試嘗試次數：
    - 暫時性錯誤/例外狀況
        - RequestTimeout
        - RequestTimeoutException
        - PriorRequestNotComplete
        - ConnectionError
        - HTTPClientError
    - 服務端調節/限制錯誤及例外狀況：
        - Throttling
        - ThrottlingException
        - ThrottledException
        - RequestThrottledException
        - TooManyRequestsException
        - ProvisionedThroughputExceededException
        - TransactionInProgressException
        - RequestLimitExceeded
        - BandwidthLimitExceeded

  - `LimitExceededException`
  - `RequestThrottled`
  - `SlowDown`
  - `EC2ThrottledException`

- 對非描述性、暫時性錯誤代碼的重試嘗試次數。特別是下列 HTTP 狀態碼：500、502、503、504。
- 任何重試嘗試都會包含基本係數為 2 的指數退避，且退避時間最長為 20 秒。

自適應重試模式

> **Warning**
>
> 自適應模式是一種實驗模式，且功能及行為會隨時變更。

自適應重試模式是一種實驗性重試模式，其中包含標準模式的所有功能。除了標準模式功能之外，自適應模式還會透過使用字符儲存貯體和每次重試嘗試時會動態更新的速率限制變數，引進用戶端速率限制。此模式在用戶端重試中提供靈活性，以適應來自 AWS 服務的錯誤/例外狀態回應。

在每次新的重試嘗試時，適應性模式會根據 AWS 服務回應中呈現的錯誤、例外狀況或 HTTP 狀態碼來修改速率限制變數。之後，這些速率限制變數會用於計算用戶端的新呼叫速率。來自 AWS 服務的每個例外狀況/錯誤或非成功 HTTP 回應 (在上方清單中提供) 會在重試發生時更新速率限制變數，直到重試成功、字符儲存貯體用盡或達到設定嘗試數值的上限為止。

設定重試模式

AWS CLI 包含各種重試組態，以及建立用戶端物件時要考慮的組態方法。

可用組態方法

在 中 AWS CLI，使用者可以使用下列方式設定重試：

- 環境變數
- AWS CLI 組態檔案

使用者可以自訂以下重試選項：

- 重試模式 - 指定 AWS CLI 使用的重試模式。如前所述，有三種可用的重試模式：傳統、標準及自適應。AWS CLI 2 是標準值。
- 最大嘗試次數 - 指定 AWS CLI 重試處理常式使用的最大重試次數值，其中初始呼叫會計入您提供的值。預設值取決於您的重試模式。

在環境變數中定義重試組態

若要定義 的重試組態 AWS CLI，請更新作業系統的環境變數。

重試環境變數是：

- `AWS_RETRY_MODE`
- `AWS_MAX_ATTEMPTS`

如需環境變數的詳細資訊，請參閱 設定 的環境變數 AWS CLI。

在組態檔案中定義重試 AWS 組態

若要變更您的重試組態，請更新您的全域 AWS 組態檔案。 AWS 組態檔案的預設位置是 ~/.aws/config。

以下是 AWS 組態檔案的範例：

```
[default]
retry_mode = standard
max_attempts = 6
```

如需有關組態檔案的詳細資訊，請參閱 中的組態和登入資料檔案設定 AWS CLI。

檢視重試嘗試的記錄

AWS CLI 使用 Boto3 的重試方法和記錄。您可以針對任何命令，使用 `--debug` 選項來接收偵錯記錄。如需如何使用 `--debug` 選項的相關資訊，請參閱 中的命令列選項 AWS CLI。

如果您在偵錯記錄中搜尋「retry」，就能找到所需的重試資訊。重試嘗試的用戶端記錄項目視您已啟用的重試模式而定。

傳統模式：

重試訊息會由 botocore.retryhandler 產生。您會看到以下三條訊息中的某一條：

- `No retry needed`
- `Retry needed, action of: <action_name>`
- `Reached the maximum number of retry attempts: <attempt_number>`

標準或自適應模式：

重試訊息會由 botocore.retries.standard 產生。您會看到以下三條訊息中的某一條：

- `No retrying request`
- `Retry needed, retrying request after delay of: <delay_value>`
- `Retry needed but retry quota reached, not retrying request`

如需 botocore 重試的完整定義檔案，請參閱 botocore GitHub 儲存庫上的 _retry.json。

針對 使用 HTTP 代理 AWS CLI

若要 AWS 透過代理伺服器存取 ，您可以使用代理伺服器使用的 DNS 網域名稱或 IP 地址和連接埠號碼來設定 `HTTP_PROXY`和 `HTTPS_PROXY`環境變數。

主題

- 使用範例
- 驗證代理
- 在 Amazon EC2 執行個體上使用代理
- 故障診斷

使用範例

> **Note**
>
> 下列範例中，環境變數以全部大寫字母顯示。但是，如果使用不同的大小寫指定一個變數兩次，則小寫字母優先。我們建議您每個變數只定義一次，以避免混淆和意外行為。

以下範例說明如何使用代理的明確 IP 地址，或可解析代理 IP 地址的 DNS 名稱。兩者後面都可以加上冒號，以及查詢應該送往的連接埠號碼。

Linux or macOS

```
$ export HTTP_PROXY=http://10.15.20.25:1234
$ export HTTP_PROXY=http://proxy.example.com:1234
$ export HTTPS_PROXY=http://10.15.20.25:5678
$ export HTTPS_PROXY=http://proxy.example.com:5678
```

Windows Command Prompt

針對所有工作階段設定

```
C:\> setx HTTP_PROXY http://10.15.20.25:1234
C:\> setx HTTP_PROXY http://proxy.example.com:1234
C:\> setx HTTPS_PROXY http://10.15.20.25:5678
C:\> setx HTTPS_PROXY http://proxy.example.com:5678
```

使用 `setx` 設定環境變數時，將會變更在目前命令提示工作階段及您在執行命令後建立的所有命令提示工作階段中使用的數值。不會影響您執行命令當時已執行的其他命令 Shell。

僅針對目前工作階段設定

使用 `set` 設定環境變數會變更使用的數值，直到目前命令提示工作階段結束或直到您將該變數設為其他數值。

```
C:\> set HTTP_PROXY=http://10.15.20.25:1234
C:\> set HTTP_PROXY=http://proxy.example.com:1234
C:\> set HTTPS_PROXY=http://10.15.20.25:5678
C:\> set HTTPS_PROXY=http://proxy.example.com:5678
```

驗證代理

> **Note**
>
> AWS CLI 不支援 NTLM 代理。如果使用 NTLM 或 Kerberos 通訊協定代理，則可以透過如 Cntlm 這樣的身分驗證代理進行連接。

AWS CLI 支援 HTTP 基本身分驗證。在代理 URL 中指定使用者名稱和密碼，如下所示。

Linux or macOS

```
$ export HTTP_PROXY=http://username:password@proxy.example.com:1234
$ export HTTPS_PROXY=http://username:password@proxy.example.com:5678
```

Windows Command Prompt

針對所有工作階段設定

```
C:\> setx HTTP_PROXY http://username:password@proxy.example.com:1234
C:\> setx HTTPS_PROXY http://username:password@proxy.example.com:5678
```

僅針對目前工作階段設定

```
C:\> set HTTP_PROXY=http://username:password@proxy.example.com:1234
C:\> set HTTPS_PROXY=http://username:password@proxy.example.com:5678
```

在 Amazon EC2 執行個體上使用代理

如果您在以連接的 IAM 角色啟動的 Amazon EC2 執行個體上設定代理，請務必將用來存取執行個體中繼資料的地址排除。若要這麼做，請將 NO_PROXY 環境變數設為執行個體中繼資料服務的 IP 地址 (169.254.169.254)。此地址不會改變。

Linux or macOS

```
$ export NO_PROXY=169.254.169.254
```

Windows Command Prompt

針對所有工作階段設定

```
C:\> setx NO_PROXY 169.254.169.254
```

僅針對目前工作階段設定

```
C:\> set NO_PROXY=169.254.169.254
```

故障診斷

如果您遇到 問題 AWS CLI，請參閱 故障診斷錯誤 以取得故障診斷步驟。如需最相關的故障診斷步驟，請參閱 the section called “SSL 憑證錯誤”。

在 中使用端點 AWS CLI

若要以程式設計方式連線至 AWS 服務，您可以使用 端點。端點是 AWS Web 服務的進入點 URL。AWS Command Line Interface (AWS CLI) 會自動為 中的每個服務使用預設端點 AWS 區域，但您可以為 API 請求指定替代端點。

端點主題

- 設定單一命令的端點
- 設定所有 的全域端點 AWS 服務
- 設定為對所有 AWS 服務使用 FIP 端點
- 設定為所有 AWS 服務使用雙堆疊端點
- 設定服務特定端點
  - 服務特定端點：環境變數
  - 服務特定端點：共用 config 檔案
  - 服務特定端點：服務特定識別碼清單
- 帳戶型端點
- 端點組態和設定優先順序

設定單一命令的端點

若要取代單一命令的任何端點設定或環境變數，請使用 `--endpoint-url` 命令列選項。下列命令範例使用自訂的 Amazon S3 端點 URL。

```
$ aws s3 ls --endpoint-url http://localhost:4567
```

設定所有 的全域端點 AWS 服務

若要將所有服務的請求路由至自訂端點 URL，請使用下列其中一個設定：

- 環境變數：

- AWS_IGNORE_CONFIGURED_ENDPOINT_URLS - 忽略已設定的端點 URL。

  Linux or macOS

  ```
  $ export AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
  ```

  Windows Command Prompt

  針對所有工作階段設定

  ```
  C:\> setx AWS_IGNORE_CONFIGURED_ENDPOINT_URLS true
  ```

  僅針對目前工作階段設定

  ```
  C:\> set AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
  ```

  PowerShell

  ```
  PS C:\> $Env:AWS_IGNORE_CONFIGURED_ENDPOINT_URLS="true"
  ```

- AWS_ENDPOINT_URL - 設定全域端點 URL。

  Linux or macOS

  ```
  $ export AWS_ENDPOINT_URL=http://localhost:4567
  ```

  Windows Command Prompt

  針對所有工作階段設定

  ```
  C:\> setx AWS_ENDPOINT_URL http://localhost:4567
  ```

  僅針對目前工作階段設定

  ```
  C:\> set AWS_ENDPOINT_URL=http://localhost:4567
  ```

  PowerShell

  ```
  PS C:\> $Env:AWS_ENDPOINT_URL="http://localhost:4567"
  ```

- config 檔案：
  - ignore_configure_endpoint_urls - 忽略已設定的端點 URL。

```
ignore_configure_endpoint_urls = true
```

  - endpoint_url - 設定全域端點 URL。

```
endpoint_url = http://localhost:4567
```

服務特定端點和 --endpoint-url 命令列選項會覆寫任何全域端點。

設定為對所有 AWS 服務使用 FIP 端點

若要路由所有服務的請求以使用 FIP 端點，請使用下列其中一項：

- AWS_USE_FIPS_ENDPOINT 環境變數。

  Linux or macOS

```
$ export AWS_USE_FIPS_ENDPOINT=true
```

  Windows Command Prompt

    針對所有工作階段設定

```
C:\> setx AWS_USE_FIPS_ENDPOINT true
```

    僅針對目前工作階段設定

```
C:\> set AWS_USE_FIPS_ENDPOINT=true
```

  PowerShell

```
PS C:\> $Env:AWS_USE_FIPS_ENDPOINT="true"
```

- use_fips_endpoint 檔案設定。

```
use_fips_endpoint = true
```

有些 AWS 服務提供端點，支援聯邦資訊處理標準 (FIPS) 140-2 AWS 區域。當 AWS 服務支援 FIPS 時，此設定會指定 AWS CLI 應使用的 FIPS 端點。與標準的 AWS 端點不同，FIPS 端點使用的是遵守 FIPS 140-2 的 TLS 軟體程式庫。會與美國政府互動的企業可能需要這些端點。

如果啟用此設定，但 服務不存在 FIPS 端點 AWS 區域，則 AWS 命令可能會失敗。在此情況下，請使用 `--endpoint-url` 選項或使用服務特定端點手動指定要在命令中使用的端點。

如需依 指定 FIPS 端點的詳細資訊 AWS 區域，請參閱依服務指定 FIPS 端點。

設定為所有 AWS 服務使用雙堆疊端點

若要路由所有服務的請求以使用雙堆疊端點 (如果可用)，請使用下列其中一項：

- `AWS_USE_DUALSTACK_ENDPOINT` 環境變數。

  Linux or macOS

  ```
  $ export AWS_USE_DUALSTACK_ENDPOINT=true
  ```

  Windows Command Prompt

  針對所有工作階段設定

  ```
  C:\> setx AWS_USE_DUALSTACK_ENDPOINT true
  ```

  僅針對目前工作階段設定

  ```
  C:\> set AWS_USE_DUALSTACK_ENDPOINT=true
  ```

  PowerShell

  ```
  PS C:\> $Env:AWS_USE_DUALSTACK_ENDPOINT="true"
  ```

- `use_dualstack_endpoint` 檔案設定。

  ```
  use_dualstack_endpoint = true
  ```

允許使用雙堆疊端點傳送 AWS 請求。若要進一步了解支援 IPv4 和 IPv6 流量的雙堆疊端點，請參閱《Amazon Simple Storage Service 使用者指南》中的使用 Amazon S3 雙堆疊端點。雙堆疊端點適用

於部分區域的某些服務。如果服務或 不存在雙堆疊端點 AWS 區域，則請求會失敗。此選項根據預設為停用。

設定服務特定端點

服務特定的端點組態提供選項，讓您選擇將持久性端點用於 AWS CLI 請求。這些設定提供彈性，以支援本機端點、VPC 端點和第三方本機 AWS 開發環境。不同的端點可用於測試和生產環境。您可以為個人 AWS 服務指定端點 URL。

可以使用下列方式來指定服務特定端點：

- 單一指令的命令列選項 `--endpoint-url`。
- 環境變數：
  - `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` - 忽略所有已設定的端點 URL，除非在命令列中指定。
  - `AWS_ENDPOINT_URL_<SERVICE>` - 指定用於特定服務的自訂端點，其中 <SERVICE> 會以 AWS 服務 識別碼取代。如需所有服務特定變數，請參閱 the section called “服務特定識別碼清單”。
- config 檔案：
  - `ignore_configure_endpoint_urls` - 忽略所有已設定的端點 URL，除非使用環境變數或在命令列中指定。
  - config 檔案的 `services` 部分與 `endpoint_url` 檔案設定結合。

服務特定端點主題：

- 服務特定端點：環境變數
- 服務特定端點：共用 config 檔案
- 服務特定端點：服務特定識別碼清單

服務特定端點：環境變數

環境變數會覆寫組態檔中的設定，但不會覆寫命令列上指定的選項。如果您希望所有設定檔都使用裝置上的相同端點，請使用環境變數。

以下是服務特定的環境變數：

- `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` - 忽略所有已設定的端點 URL，除非在命令列中指定。

Linux or macOS

```
$ export AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

Windows Command Prompt

針對所有工作階段設定

```
C:\> setx AWS_IGNORE_CONFIGURED_ENDPOINT_URLS true
```

僅針對目前工作階段設定

```
C:\> set AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

PowerShell

```
PS C:\> $Env:AWS_IGNORE_CONFIGURED_ENDPOINT_URLS="true"
```

- AWS_ENDPOINT_URL_<SERVICE> - 指定用於特定服務的自訂端點，其中 <SERVICE> 以 AWS 服務 識別符取代。如需所有服務特定變數，請參閱 the section called “服務特定識別碼清單”。

下列環境變數範例會設定 AWS Elastic Beanstalk的端點。

Linux or macOS

```
$ export AWS_ENDPOINT_URL_ELASTIC_BEANSTALK=http://localhost:4567
```

Windows Command Prompt

針對所有工作階段設定

```
C:\> setx AWS_ENDPOINT_URL_ELASTIC_BEANSTALK http://localhost:4567
```

僅針對目前工作階段設定

```
C:\> set AWS_ENDPOINT_URL_ELASTIC_BEANSTALK=http://localhost:4567
```

PowerShell

```
PS C:\> $Env:AWS_ENDPOINT_URL_ELASTIC_BEANSTALK="http://localhost:4567"
```

如需有關設定環境變數的詳細資訊，請參閱 the section called “環境變數”。

服務特定端點：共用 `config` 檔案

在共用 `config` 檔案中，`endpoint_url` 用於多個部分。若要設定服務特定端點，請使用巢狀化在 `services` 區段中服務識別碼金鑰下的 `endpoint_url` 設定。如需在共用 `config` 檔案中定義 `services` 區段的詳細資訊，請參閱 the section called “`services`”。

下列範例使用 `services` 區段為 Amazon S3 設定服務特定端點 URL，以及用於所有其他服務的自訂全球端點：

```
[profile dev1]
endpoint_url = http://localhost:1234
services = s3-specific

[services testing-s3]
s3 =
  endpoint_url = http://localhost:4567
```

單一設定檔可以設定多個服務的端點。下列範例會在相同的設定檔中設定 Amazon S3 和 AWS Elastic Beanstalk 的服務特定端點 URL。

如需要在 `services` 區段中使用的所有服務識別碼金鑰的清單，請參閱 服務特定識別碼清單。

```
[profile dev1]
services = testing-s3-and-eb

[services testing-s3-and-eb]
s3 =
  endpoint_url = http://localhost:4567
elastic_beanstalk =
  endpoint_url = http://localhost:8000
```

服務組態區段可用於多個設定檔。以下範例有兩個設定檔使用相同的 `services` 定義：

```
[profile dev1]
```

```
output = json
services = testing-s3

[profile dev2]
output = text
services = testing-s3

[services testing-s3]
s3 =
  endpoint_url = https://localhost:4567
```

服務特定端點：服務特定識別碼清單

AWS 服務 識別符是以 API 模型的 為基礎`serviceId`，方法是以底線取代所有空格，並縮小所有字母大小。

下表列出所有服務特定識別碼、`config` 檔案金鑰和環境變數。

帳戶型端點

帳戶型端點可以透過下列方式指定：

- 環境變數
  - AWS_ACCOUNT_ID - AWS 指定用於呼叫支援的 帳戶型端點 ID AWS 服務。

    Linux or macOS

    ```
    $ export AWS_ACCOUNT_ID=<account-id>
    ```

    Windows Command Prompt

    針對所有工作階段設定

    ```
    C:\> setx AWS_ACCOUNT_ID <account-id>
    ```

    僅針對目前工作階段設定

    ```
    C:\> set AWS_ACCOUNT_ID=<account-id>
    ```

PowerShell

```
PS C:\> $Env:AWS_ACCOUNT_ID="<account-id>"
```

- AWS_ACCOUNT_ID_ENDPOINT_MODE - 指定是否使用 AWS 帳戶型端點 IDs來呼叫支援的 AWS 服務。可設定為 preferred、disabled 或 required。預設值為 preferred。

  Linux or macOS

  ```
  $ export AWS_ACCOUNT_ID_ENDPOINT_MODE=preferred
  ```

  Windows Command Prompt

  針對所有工作階段設定

  ```
  C:\> setx AWS_ACCOUNT_ID_ENDPOINT_MODE preferred
  ```

  僅針對目前工作階段設定

  ```
  C:\> set AWS_ACCOUNT_ID_ENDPOINT_MODE=preferred
  ```

  PowerShell

  ```
  PS C:\> $Env:AWS_ACCOUNT_ID_ENDPOINT_MODE="preferred"
  ```

- config 檔案：
  - aws_account_id - 指定用於呼叫支援的 AWS 帳戶型端點 ID AWS 服務。

    ```
    aws_account_id = <account-id>
    ```

  - account_id_endpoint_mode - 指定是否使用 AWS 帳戶型端點 IDs來呼叫支援的 AWS 服務。可設定為 preferred、disabled 或 required。預設值為偏好。

    ```
    account_id_endpoint_mode = preferred
    ```

帳戶型端點使用 AWS 您的帳戶 ID 簡化支援此功能之服務的 AWS 服務 請求路由，有助於確保高效能和可擴展性。當您使用登入資料提供者和支援以帳戶為基礎的端點的服務時， AWS CLI 會自動建構並使用以帳戶為基礎的端點，而不是區域性端點。

以帳戶為基礎的端點使用下列格式，其中 <account-id> 會取代為 AWS 您的帳戶 ID，而 <region> 會取代為您的 AWS 區域：

```
https://<account-id>.myservice.<region>.amazonaws.com
```

在 中 AWS CLI，帳戶型端點模式預設為 preferred。

端點組態和設定優先順序

端點組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。 AWS CLI 端點組態設定的優先順序如下：

1. --endpoint-url 命令列選項。
2. 如果啟用，則 AWS_IGNORE_CONFIGURED_ENDPOINT_URLS 全域端點環境變數或設定檔設定 ignore_configure_endpoint_urls 將忽略自訂端點。
3. 服務特定環境變數 AWS_ENDPOINT_URL_<SERVICE> 所提供的值，例如 AWS_ENDPOINT_URL_DYNAMODB。
4. AWS_USE_DUALSTACK_ENDPOINT、AWS_USE_FIPS_ENDPOINT 和 AWS_ENDPOINT_URL 環境變數所提供的值。
5. AWS_ACCOUNT_ID_ENDPOINT_MODE 環境變數設定為 preferred required 或使用AWS_ACCOUNT_ID環境變數或aws_account_id設定中的帳戶 ID。
6. 共用 config 檔案 services 區段內的 endpoint_url 設定所提供的服務特定端點值。
7. 共用 config 檔案的 profile 中的 endpoint_url 設定所提供的值。
8. use_dualstack_endpoint、use_fips_endpoint 和 endpoint_url 設定。
9. account_id_endpoint_mode 設定設定為 required preferred或使用AWS_ACCOUNT_ID環境變數或aws_account_id設定中的帳戶 ID。
10. 最後 AWS 服務 會使用個別 的任何預設端點 URL。如需每個區域可用的標準服務端點清單，請參閱 Amazon Web Services 一般參考 中的 AWS 區域與端點。

的身分驗證和存取憑證 AWS CLI

當您使用 服務開發 AWS 時，您必須建立 如何向 AWS 進行 AWS CLI 身分驗證。若要為 設定程式設計存取的登入資料 AWS CLI，請選擇下列其中一個選項。選項是按照建議順序排列的。

身分驗證類型	用途	指示
IAM Identity Center 人力資源使用者短期憑證	（建議）使用 IAM Identity Center 人力資源使用者的短期憑證。 安全最佳實務是 AWS Organizations 搭配 IAM Identity Center 使用 。它結合了短期登入資料與使用者目錄，例如內建的 IAM Identity Center 目錄或 Active Directory 。	the section called “IAM Identity Center 驗證”
IAM 使用者短期憑證	使用比長期憑證更安全的 IAM 使用者短期憑證。如果您的登入資料遭到入侵，在過期之前，可以使用這些登入資料的時間有限。	the section called “短期憑證”
Amazon EC2 執行個體上的 IAM 或 IAM Identity Center 使用者。	使用 Amazon EC2 執行個體中繼資料，使用指派給 Amazon EC2 執行個體的角色來查詢臨時登入資料。	the section called “Amazon EC2 中繼資料”
擔任 許可的角色	配對另一個登入資料方法，並擔任暫時存取 AWS 服務 使用者的角色，可能無法存取 。	the section called “IAM 角色”
IAM 使用者長期憑證	（不建議）使用沒有過期的長期登入資料。	the section called “IAM 使用者”

身分驗證類型	用途	指示
IAM 或 IAM Identity Center 人力資源使用者的外部儲存	（不建議）配對其他登入資料方法，但將登入資料值存放在外部的位置 AWS CLI。此方法僅與存放登入資料的外部位置一樣安全。	the section called “外部憑證”

組態和憑證優先順序

登入資料和組態設定位於多個位置，例如系統或使用者環境變數、本機 AWS 組態檔案，或在命令列上明確宣告為參數。某些身分驗證優先於其他身分驗證。 AWS CLI 身分驗證設定優先順序如下：

1. 命令列選項 – 覆寫任何其他位置 (例如 `--region`、`--output` 和 `--profile` 參數) 的設定。
2. 環境變數 – 您可以將數值存放在環境變數中。
3. 擔任角色—透過組態或 `assume-role` 命令，獲得 IAM 角色許可。
4. 擔任具有 Web 身分的角色—透過組態或 `assume-role-with-web-identity` 命令，使用 Web 身分獲得 IAM 角色許可。
5. AWS IAM Identity Center – 當您執行 `aws configure sso`命令時，會更新存放在 `config` 檔案中的 IAM Identity Center 組態設定。然後，在您執行 `aws sso login`命令時驗證登入資料。`config` 檔案在 Linux 或 macOS 上位於 `~/.aws/config`，在 Windows 上位於 `C:\Users\`*`USERNAME`*`\.aws\config`。
6. 憑證檔案 – 當您執行 `aws configure` 命令時，會更新 `credentials` 和 `config` 檔案。`credentials` 檔案在 Linux 或 macOS 上位於 `~/.aws/credentials`，在 Windows 上位於 `C:\Users\`*`USERNAME`*`\.aws\credentials`。
7. 自訂程序—從外部來源取得憑證。
8. 組態檔 – 當您執行 `aws configure` 命令時，會更新 `credentials` 和 `config` 檔案。`config` 檔案在 Linux 或 macOS 上位於 `~/.aws/config`，在 Windows 上位於 `C:\Users\`*`USERNAME`*`\.aws\config`。
9. 容器憑證 – 您可以將 IAM 角色與您的每一個 Amazon Elastic Container Service (Amazon ECS) 任務定義關聯起來。然後，該角色的臨時憑證就可供該任務的容器使用。如需詳細資訊，請參閱《Amazon Elastic Container Service 開發人員指南》中的任務 IAM 角色。
10. Amazon EC2 執行個體設定檔憑證 – 您可以將 IAM 角色與您的每一個 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體關聯起來。然後，該角色的臨時憑證就可供執行個體中執行的程式碼

使用。憑證是透過 Amazon EC2 中繼資料服務傳遞。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Amazon EC2 的 IAM 角色和《IAM 使用者指南》中的使用執行個體設定檔。 Amazon EC2

本區段的其他主題

- the section called “IAM Identity Center 驗證”
- the section called “短期憑證”
- the section called “IAM 角色”
- the section called “IAM 使用者”
- the section called “Amazon EC2 中繼資料”
- the section called “外部憑證”

使用 設定 IAM Identity Center 身分驗證 AWS CLI

本主題說明如何 AWS CLI 使用 AWS IAM Identity Center (IAM Identity Center) 設定 以擷取登入資料來執行 AWS CLI 命令。使用 IAM Identity Center 對使用者進行身分驗證，以取得登入資料以透過 `config` 檔案執行 AWS CLI 命令的方式主要有兩種：

- （建議）SSO 權杖提供者組態。
- 舊版不可重新整理的配置。

如需有關使用不使用帳戶 ID 和角色的承載身分驗證的資訊，請參閱《Amazon CodeCatalyst 使用者指南》中的設定 以 AWS CLI 搭配 CodeCatalyst 使用 。 CodeCatalyst

> **Note**
>
> 如需搭配 AWS CLI 命令使用 IAM Identity Center 的引導程序，請參閱 the section called “教學課程： AWS IAM Identity Center 和 Amazon S3”。

主題

- the section called “先決條件”
- the section called “使用 `aws configure sso` 精靈設定您的設定檔”

- the section called “使用 aws configure sso-session 精靈只設定 sso-session 區段”
- the section called “使用 config 檔案手動設定”
- the section called “登入 IAM Identity Center 工作階段”
- the section called “使用 IAM Identity Center 設定檔執行命令”
- the section called “登出 IAM Identity Center 工作階段”
- the section called “故障診斷”
- the section called “相關資源”

先決條件

- 安裝 AWS CLI。如需詳細資訊，請參閱the section called “安裝/更新”。
- 您必須先有辦法存取 IAM Identity Center 中的 SSO 身分驗證。選擇下列其中一種方法來存取您的 AWS 登入資料。

我沒有透過 IAM Identity Center 建立存取權

請遵循 AWS IAM Identity Center 使用者指南的入門說明。此程序可以啟動 IAM Identity Center、建立管理使用者，並新增適當的最低權限許可集。

> **Note**
>
> 建立套用最低權限許可的許可集。建議您使用預先定義的 PowerUserAccess 許可集，除非您的雇主已為此目的建立了自訂許可集。

結束入口網站並再次登入，以查看 AWS 帳戶或 Administrator 的程式設計存取詳細資訊和選項PowerUserAccess。在使用 SDK 時選取 PowerUserAccess。

我已經 AWS 可以透過我的雇主管理的聯合身分提供者 （例如 Azure AD 或 Okta) 存取

透過 AWS 身分提供者的入口網站登入 。如果您的雲端管理員已授予您 PowerUserAccess（開發人員） 許可，您會看到您有權存取 AWS 帳戶 的 和許可集。您會在許可集名稱旁，看到使用該許可集手動或以程式設計方式存取帳戶的選項。

若您自訂實作，可能會產生不同體驗，例如不同的許可集名稱。若您不確定要使用哪個許可集，請聯絡您的 IT 團隊尋求協助。

我已經 AWS 可以透過我的雇主管理的 AWS 存取入口網站存取

AWS 透過您的 AWS 存取入口網站登入 。如果您的雲端管理員已授予您 PowerUserAccess（開發人員）許可，您會看到您有權存取 AWS 帳戶 的 和許可集。您會在許可集名稱旁，看到使用該許可集手動或以程式設計方式存取帳戶的選項。

我已經能夠 AWS 透過由我的雇主管理的聯合身分自訂身分提供者存取

請聯絡您的 IT 團隊尋求協助。

取得 IAM Identity Center 的存取權後，請執行下列動作來收集您的 IAM Identity Center 資訊：

1. 收集您需要執行的 `SSO Start URL`和 `SSO Region`值 `aws configure sso`

   a. 在您的 AWS 存取入口網站中，選取您用於開發的許可集，然後選取存取金鑰連結。

   b. 在取得登入資料對話方塊中，選擇符合您作業系統的索引標籤。

   c. 選擇 IAM Identity Center 登入資料方法，以取得 `SSO Start URL`和 `SSO Region`值。

2. 或者，從 2.22.0 版開始，您可以使用發行者 URL，而不是啟動 URL。發行者 URL 位於 AWS IAM Identity Center 主控台中下列其中一個位置：
   - 在儀表板頁面上，發行者 URL 位於設定摘要中。
   - 在設定頁面上，發行者 URL 位於身分來源設定中。
3. 如需要在哪些範圍註冊值的詳細資訊，請參閱《IAM Identity Center 使用者指南》中的 OAuth 2.0 存取範圍。

使用 `aws configure sso` 精靈設定您的設定檔

若要為您的 設定 IAM Identity Center 設定檔 AWS CLI：

1. 在您偏好的終端機中，執行 `aws configure sso`命令。

   (Recommended) IAM Identity Center

   建立工作階段名稱、提供您的 IAM Identity Center 啟動 URL 或發行者 URL、託管 IAM Identity Center 目錄 AWS 區域 的 ，以及註冊範圍。

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

```
SSO registration scopes [None]: sso:account:access
```

根據預設，從 2.22.0 版 AWS CLI 開始的 會使用 Code Exchange (PKCE) 授權的驗證金鑰，並且必須在具有瀏覽器的裝置上使用。 若要繼續使用裝置授權，請附加 `--use-device-code`選項。

```
$ aws configure sso --use-device-code
```

Legacy IAM Identity Center

略過工作階段名稱，並提供您的 IAM Identity Center 啟動 URL 和 AWS 託管 Identity Center 目錄的區域。

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

2. 會 AWS CLI 嘗試開啟預設瀏覽器，以進行 IAM Identity Center 帳戶的登入程序。此程序可能會提示您允許 AWS CLI 存取您的資料。由於 AWS CLI 建置在適用於 Python 的 SDK 之上，因此許可訊息可能包含`botocore`名稱的變化。
   - 如果 AWS CLI 無法開啟瀏覽器，系統會根據您使用的授權類型，顯示手動啟動登入程序的指示。

     PKCE authorization

     根據預設，從 2.22.0 版 AWS CLI 開始， 會使用 Code Exchange (PKCE) 授權的驗證金鑰。顯示的 URL 是從 *https://oidc.us-east-1.amazonaws.com/authorize* 開始的唯一 URL

     PKCE URLs 必須在您登入的相同裝置上開啟，且必須用於具有瀏覽器的裝置。

```
Attempting to automatically open the SSO authorization
 page in your
default browser.
If the browser does not open or you wish to use a
 different device to
authorize the request, open the following URL:
```

```
https://oidc.us-east-1.amazonaws.com/authorize?
<abbreviated>
```

Device authorization

2.22.0 之前的版本使用 AWS CLI OAuth 2.0 裝置授權。您可以使用 `--use-device-code`選項，在較新版本上啟用此方法。

裝置授權 URLs 不需要在您登入的相同裝置上開啟，可用於使用或不使用瀏覽器的裝置。

```
If the browser does not open or you wish to use a
 different device to
authorize this request, open the following URL:
https://device.sso.us-west-2.amazonaws.com/

Then enter the code:
QCFK-N451
```

3. 從顯示的清單中選擇要使用 AWS 的帳戶。如果您有權僅使用一個帳戶，則 AWS CLI 會自動選取該帳戶並略過提示。

```
There are 2 AWS accounts available to you.
> DeveloperAccount, developer-account-admin@example.com (123456789011)
  ProductionAccount, production-account-admin@example.com (123456789022)
```

4. 從顯示的清單中選擇要使用的 IAM 角色。如果只有一個角色可用， AWS CLI 會自動選取該角色並略過提示。

```
Using the account ID 123456789011
There are 2 roles available to you.
> ReadOnly
  FullAccess
```

5. 指定預設輸出格式、要傳送命令的預設值 AWS 區域，以及設定檔的名稱。如果您指定 `default`做為設定檔名稱，此設定檔會成為使用的預設設定檔。在下列範例中，使用者會輸入預設區域、預設輸出格式和設定檔的名稱。

```
Default client Region [None]: us-west-2<ENTER>
CLI default output format (json if not specified) [None]: json<ENTER>
Profile name [123456789011_ReadOnly]: my-dev-profile<ENTER>
```

6. 最後一則訊息說明完成的設定檔組態。您現在可以使用此設定檔來請求登入資料。使用 `aws sso login` 命令來請求和擷取執行命令所需的憑證。如需說明，請參閱登入 IAM Identity Center 工作階段。

產生的組態檔案

這些步驟會在 `config` 檔案中建立 `sso-session`區段和具名設定檔，如下所示：

IAM Identity Center

```
[profile my-dev-profile]
sso_session = my-sso
sso_account_id = 123456789011
sso_role_name = readOnly
region = us-west-2
output = json

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

Legacy IAM Identity Center

```
[profile my-dev-profile]
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_region = us-east-1
sso_account_id = 123456789011
sso_role_name = readOnly
region = us-west-2
output = json
```

使用 `aws configure sso-session` 精靈只設定 `sso-session` 區段

> **Note**
>
> 此組態與舊版 IAM Identity Center 不相容。

aws configure sso-session 命令會更新 ~/.aws/config 檔案中的sso-session區段。執行 aws configure sso-session命令並提供您的 IAM Identity Center 啟動 URL 或發行者 URL，以及託管 AWS IAM Identity Center 目錄的區域。

```
$ aws configure sso-session
SSO session name: my-sso
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]: us-east-1
SSO registration scopes [None]: sso:account:access
```

使用 config 檔案手動設定

IAM Identity Center 組態資訊會存放在 config 檔案中，並且可以使用文字編輯器進行編輯。若要手動將 IAM Identity Center 支援新增至具名設定檔，您必須將金鑰和值新增至 config 檔案。

IAM Identity Center 組態檔案

config 檔案的 sso-session區段用於將用於取得 SSO 存取權杖的組態變數分組，然後可用於取得 AWS 登入資料。使用下列的設定：

- (必要) sso_start_url
- （必要） sso_region
- sso_account_id
- sso_role_name
- sso_registration_scopes

您可以定義 sso-session區段，並將其與設定檔建立關聯。sso_region 和 sso_start_url設定必須在 sso-session區段中設定。一般而言，sso_account_id 和 sso_role_name 必須在 profile 區段中設定，以讓開發套件請求 SSO 憑證。

下列範例會將開發套件設定為請求 SSO憑認證，並支援自動字符重新整理：

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

[sso-session my-sso]
sso_region = us-east-1
```

```
sso_start_url = https://my-sso-portal.awsapps.com/start
```

這也讓 sso-session 組態能在多個設定檔中重複使用：

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

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

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

然而，SSO 字符組態並非所有情況都需要 sso_account_id 和 sso_role_name。如果您的應用程式僅使用支援承載身分驗證 AWS 的服務，則不需要傳統 AWS 登入資料。承載身分驗證是一種 HTTP 身分驗證結構描述，使用稱為承載字符的安全字符。在這種情況下，sso_account_id 和 sso_role_name 並非必要資訊。請參閱您 AWS 服務的個別指南，以判斷其是否支援承載字符授權。

此外，註冊範圍也可以設定為 sso-session 的一部分。範圍是 OAuth 2.0 中的一種機制，用於限制應用程式對使用者帳戶的存取。應用程式可以請求一個或多個範圍，則合法給應用程式的存取字符將僅限於授予的範圍。這些範圍定義為已註冊 OIDC 用戶端和用戶端擷取的存取字符進行授權所需請求的許可。下列範例設定 sso_registration_scopes 以提供列出帳戶/角色的存取權限：

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

身分驗證字符會以基於工作階段名稱的檔案名稱快取至 sso/cache 目錄下的磁碟。

舊版 IAM Identity Center 組態檔案

> **Note**
>
> 使用舊版不可重新整理的組態，不支援自動字符重新整理。我們建議使用 SSO 字符組態。

若要手動將 IAM Identity Center 支援新增至具名設定檔，您必須將下列索引鍵和值新增至 `config` 檔案中的設定檔定義。

- `sso_start_url`
- `sso_region`
- `sso_account_id`
- `sso_role_name`

您可以在 `.aws/config` 檔案中包含任何其他有效的索引鍵和值。下列範例是 IAM Identity Center 設定檔：

```
[profile my-sso-profile]
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_region = us-west-2
sso_account_id = 111122223333
sso_role_name = SSOReadOnlyRole
region = us-west-2
output = json
```

若要執行命令，您必須先the section called “登入 IAM Identity Center 工作階段”請求和擷取暫時登入資料。

如需 `config` 和 `credentials` 檔案的詳細資訊，請參閱 the section called “組態設定”。

登入 IAM Identity Center 工作階段

> **Note**
>
> 登入程序可能會提示您允許 AWS CLI 存取您的資料。由於 AWS CLI 建置在適用於 Python 的 SDK 之上，因此許可訊息可能包含botocore名稱的變化。

若要擷取和快取一組 IAM Identity Center 登入資料，請執行下列命令 AWS CLI ，讓 開啟您的預設瀏覽器並驗證您的 IAM Identity Center 登入。

```
$ aws sso login --profile my-dev-profile
SSO authorization page has automatically been opened in your default browser.
Follow the instructions in the browser to complete this authorization request.
```

```
Successfully logged into Start URL: https://my-sso-portal.awsapps.com/start
```

您的 IAM Identity Center 工作階段登入資料會快取，而 AWS CLI 會使用它們來安全地擷取設定檔中指定之 IAM 角色的 AWS 登入資料。

如果 AWS CLI 無法開啟您的瀏覽器

如果 AWS CLI 無法自動開啟您的瀏覽器，系統會根據您使用的授權類型，顯示手動啟動登入程序的指示。

PKCE authorization

根據預設，從 2.22.0 版 AWS CLI 開始， 會使用 Code Exchange (PKCE) 授權的驗證金鑰。顯示的 URL 是以下列開頭的唯一 URL：https：//*https://oidc.us-east-1.amazonaws.com/authorize*

PKCE URLs 必須在您登入的相同裝置上開啟，且必須用於具有瀏覽器的裝置。

```
Attempting to automatically open the SSO authorization
 page in your
default browser.
If the browser does not open or you wish to use a
 different device to
authorize the request, open the following URL:

https://oidc.us-east-1.amazonaws.com/authorize?
<abbreviated>
```

Device authorization

2.22.0 之前的 AWS CLI 版本使用 OAuth 2.0 裝置授權。您可以使用 `--use-device-code`選項，在較新版本上啟用此方法。

裝置授權 URLs 不需要在您登入的相同裝置上開啟，可用於使用或不使用瀏覽器的裝置。

```
If the browser does not open or you wish to use a
 different device to
authorize this request, open the following URL:
https://device.sso.us-west-2.amazonaws.com/

Then enter the code:
```

```
QCFK-N451
```

您也可以指定使用 `aws sso login` 指令的 `--sso-session` 參數登入時要使用的 `sso-session` 設定檔。`sso-session` 選項不適用於舊版 IAM Identity Center。

```
$ aws sso login --sso-session my-dev-session
```

從 2.22.0 版開始，PKCE 授權為預設值。若要使用裝置授權登入，請新增 `--use-device-code`選項。

```
$ aws sso login --profile my-dev-profile --use-device-code
```

身分驗證字符會快取至`~/.aws/sso/cache`目錄下的磁碟，並以 為基礎的檔案名稱`sso_start_url`。

使用 IAM Identity Center 設定檔執行命令

登入後，您可以使用 登入資料來呼叫具有相關聯具名設定檔的 AWS CLI 命令。下列範例顯示使用設定檔的命令：

```
$ aws sts get-caller-identity --profile my-dev-profile
```

只要您已登入 IAM Identity Center 且這些快取的登入資料尚未過期， 就會視需要 AWS CLI 自動續約過期的 AWS 登入資料。但是，如果您的 IAM Identity Center 憑證過期，必須重新登入您的 IAM Identity Center 帳戶，以明確更新它們。

登出 IAM Identity Center 工作階段

使用 IAM Identity Center 設定檔完成後，您可以讓登入資料過期，或執行下列命令來刪除快取的登入資料。

```
$ aws sso logout
Successfully signed out of all SSO profiles.
```

故障診斷

如果您使用 遇到問題 AWS CLI，請參閱 故障診斷錯誤 以取得疑難排解步驟。

相關資源

其他資源如下所示。

- the section called “IAM Identity Center 概念”
- the section called “教學課程： AWS IAM Identity Center 和 Amazon S3”
- the section called “安裝/更新”
- the section called “組態設定”
- `aws configure sso` 第 AWS CLI 2 版參考中的
- `aws configure sso-session` 第 AWS CLI 2 版參考中的
- `aws sso login` 第 AWS CLI 2 版參考中的
- `aws sso logout` 第 AWS CLI 2 版參考中的
- 《Amazon CodeCatalyst 使用者指南》中的設定 以 AWS CLI 搭配 CodeCatalyst 使用 CodeCatalyst
- IAM Identity Center 使用者指南中的 OAuth 2.0 存取範圍
- IAM Identity Center 使用者指南中的入門教學課程

AWS IAM Identity Center 的概念 AWS CLI

本主題說明 AWS IAM Identity Center (IAM Identity Center) 的重要概念。IAM Identity Center 是一種雲端型 IAM 服務，透過與現有的身分提供者 (IdP) 整合 AWS 帳戶，簡化跨多個應用程式、SDKs和工具的使用者存取管理。它透過集中式使用者入口網站啟用安全的單一登入、許可管理和稽核，簡化組織的身分和存取控管。

主題

- 什麼是 IAM Identity Center
- 條款
- IAM Identity Center 的運作方式
- 其他資源

什麼是 IAM Identity Center

IAM Identity Center 是一種雲端型身分和存取管理 (IAM) 服務，可讓您集中管理對多個 AWS 帳戶 和商業應用程式的存取。

它提供使用者入口網站，授權使用者可以使用其現有的公司登入資料，存取他們獲得許可的 AWS 帳戶和應用程式。這可讓組織強制執行一致的安全政策，並簡化使用者存取管理。

無論您使用哪種 IdP，IAM Identity Center 都會將這些區別擷取出來。例如，您可以按照部落格文章 IAM Identity Center 的下一個演變中所述的方法連線 Microsoft Azure AD。

> **Note**
>
> 如需有關使用不使用帳戶 ID 和角色的承載身分驗證的資訊，請參閱《Amazon CodeCatalyst 使用者指南》中的設定 以 AWS CLI 搭配 CodeCatalyst 使用 。 CodeCatalyst

條款

使用 IAM Identity Center 時的常用術語如下：

身分提供者 (IdP)

身分管理系統，例如 IAM Identity Center、Microsoft Azure AD、Okta 或您自己的公司目錄服務。

AWS IAM Identity Center

IAM Identity Center 是 AWS 擁有的 IdP 服務。以前稱為 AWS 單一登入，SDKs和工具會保留 `sso` API 命名空間，以確保回溯相容性。如需詳細資訊，請參閱AWS IAM Identity Center 《 使用者指南》中的 IAM Identity Center 重新命名。

AWS 存取入口網站 URL、SSO 啟動 URL、啟動 URL

您組織的唯一 IAM Identity Center URL，以存取授權的 AWS 帳戶、 服務和資源。

發行者 URL

您組織的唯一 IAM Identity Center 發行者 URL AWS 帳戶，可用於授權、服務和資源的程式設計存取。從 2.22.0 版開始 AWS CLI，發行者 URL 可與啟動 URL 互換使用。

聯合

在 IAM Identity Center 與身分提供者之間建立信任的程序，以啟用單一登入 (SSO)。

AWS 帳戶

您提供使用者透過 存取 AWS 帳戶 的 AWS IAM Identity Center。

許可集、 AWS 憑證、憑證、sigv4 憑證

可指派給使用者或群組以授予存取權的預先定義許可集合 AWS 服務。

註冊範圍、存取範圍、範圍

範圍是 OAuth 2.0 中的一種機制，用於限制應用程式對使用者帳戶的存取。應用程式可以請求一或多個範圍，而核發給應用程式的存取權杖僅限於授予的範圍。如需範圍的資訊，請參閱《IAM Identity Center 使用者指南》中的存取範圍。

權杖、重新整理權杖、存取權杖

權杖是在身分驗證時核發給您的臨時安全登入資料。這些字符包含有關您的身分和已授予許可的資訊。

當您透過 IAM Identity Center 入口網站存取 AWS 資源或應用程式時，您的字符會呈現給 AWS 以進行身分驗證和授權。這可讓 AWS 驗證您的身分，並確保您擁有執行所請求動作的必要許可。

身分驗證字符會根據工作階段名稱，以 JSON 檔案名稱快取至 `~/.aws/sso/cache`目錄下的磁碟。

Session (工作階段)

IAM Identity Center 工作階段是指使用者經過身分驗證並獲授權存取 AWS 資源或應用程式的期間。當使用者登入 IAM Identity Center 入口網站時，會建立工作階段，且使用者的字符在指定的持續時間內有效。如需設定工作階段持續時間的詳細資訊，請參閱AWS IAM Identity Center 《 使用者指南》中的設定工作階段持續時間。

在工作階段期間，您可以在不同的 AWS 帳戶和應用程式之間導覽，而不必重新驗證，只要其工作階段保持作用中狀態即可。當工作階段過期時，請再次登入以續約您的存取權。

IAM Identity Center 工作階段有助於提供順暢的使用者體驗，同時透過限制使用者存取憑證的有效性來強制執行安全最佳實務。

使用 PKCE、PKCE、Code Exchange 的驗證金鑰授予授權碼

從 2.22.0 版開始，Code Exchange (PKCE) 的驗證金鑰是具有瀏覽器之裝置的 OAuth 2.0 身分驗證授予流程。PKCE 是一種簡單且安全的方式，可讓您透過 Web 瀏覽器，從桌面和行動裝置驗證並取得存取資源 AWS 的同意。這是預設的授權行為。如需 PKCE 的詳細資訊，請參閱AWS IAM Identity Center 《 使用者指南》中的使用 PKCE 授予授權碼。

裝置授權授予

使用或不使用 Web 瀏覽器之裝置的 OAuth 2.0 身分驗證授予流程。如需設定工作階段持續時間的詳細資訊，請參閱AWS IAM Identity Center 《 使用者指南》中的裝置授權授予。

IAM Identity Center 的運作方式

IAM Identity Center 會與您組織的身分提供者整合，例如 IAM Identity Center、Microsoft Azure AD 或 Okta。使用者對此身分提供者進行身分驗證，然後 IAM Identity Center 將這些身分映射到您 AWS 環境中的適當許可和存取權。

下列 IAM Identity Center 工作流程假設您已 AWS CLI 將 設定為使用 IAM Identity Center：

1. 在您偏好的終端機中，執行 `aws sso login`命令。
2. 登入您的 AWS 存取入口網站 以啟動新的工作階段。
   - 當您啟動新的工作階段時，會收到重新整理字符和快取的存取字符。
   - 如果您已經有作用中的工作階段，則會重複使用現有的工作階段，並在現有的工作階段過期時過期。
3. IAM Identity Center 會根據您在 `config` 檔案中設定的設定檔，取得適當的許可集，授予相關 AWS 帳戶 和應用程式的存取權。
4. AWS CLI、 SDKs和工具會使用您擔任的 IAM 角色來呼叫 ， AWS 服務 例如建立 Amazon S3 儲存貯體，直到該工作階段過期為止。
5. IAM Identity Center 的存取權杖會每小時檢查一次，並使用重新整理權杖自動重新整理。
   - 如果存取權杖已過期，開發套件或工具會使用重新整理權杖來取得新的存取權杖。然後會比較這些字符的工作階段持續時間，如果重新整理字符不是過期的 IAM Identity Center 會提供新的存取字符。
   - 如果重新整理字符已過期，則不會提供新的存取字符，並且您的工作階段已結束。
6. 工作階段會在重新整理權杖過期後結束，或者當您使用 `aws sso logout`命令手動登出時結束。快取的登入資料會移除。若要繼續使用 IAM Identity Center 存取服務，您必須使用 `aws sso login`命令啟動新的工作階段。

其他資源

其他資源如下所示。

- the section called “IAM Identity Center 驗證”
- the section called “教學課程： AWS IAM Identity Center 和 Amazon S3”
- the section called “安裝/更新”
- the section called “組態設定”
- `aws configure sso` 第 AWS CLI 2 版參考中的

- `aws configure sso-session` 第 AWS CLI 2 版參考中的
- `aws sso login` 第 AWS CLI 2 版參考中的
- `aws sso logout` 第 AWS CLI 2 版參考中的
- 《Amazon CodeCatalyst 使用者指南》中的設定 以 AWS CLI 搭配 CodeCatalyst 使用 CodeCatalyst
- 《 AWS IAM Identity Center 使用者指南》中的 IAM Identity Center 重新命名
- IAM Identity Center 使用者指南中的 OAuth 2.0 存取範圍
- AWS IAM Identity Center 《 使用者指南》中的設定工作階段持續時間
- IAM Identity Center 使用者指南中的入門教學課程

教學課程：使用 IAM Identity Center 在 中執行 Amazon S3 命令 AWS CLI

本主題說明如何設定 AWS CLI 以使用目前的 AWS IAM Identity Center (IAM Identity Center) 驗證使用者，以擷取 AWS Command Line Interface (Amazon S3) 的登入資料以執行 Amazon Simple Storage Service (AWS CLI) 命令。

主題

- 步驟 1：IAM Identity Center 中的身分驗證
- 步驟 2：收集您的 IAM Identity Center 資訊
- 步驟 3：建立 Amazon S3 儲存貯體
- 步驟 4：安裝 AWS CLI
- 步驟 5：設定您的 AWS CLI 設定檔
- 步驟 6：登入 IAM Identity Center
- 步驟 7：執行 Amazon S3 命令
- 步驟 8：登出 IAM Identity Center
- 步驟 9：清除資源
- 故障診斷
- 其他資源

步驟 1：IAM Identity Center 中的身分驗證

存取 IAM Identity Center 中的 SSO 身分驗證。選擇下列其中一種方法來存取您的 AWS 登入資料。

我沒有透過 IAM Identity Center 建立存取權

請遵循 AWS IAM Identity Center 使用者指南的入門說明。此程序可以啟動 IAM Identity Center、建立管理使用者，並新增適當的最低權限許可集。

> **Note**
>
> 建立套用最低權限許可的許可集。建議您使用預先定義的 PowerUserAccess 許可集，除非您的雇主已為此目的建立了自訂許可集。

結束入口網站並再次登入，以查看 AWS 帳戶或 Administrator 的程式設計存取詳細資訊和選項PowerUserAccess。在使用 SDK 時選取 PowerUserAccess。

我已經 AWS 可以透過我的雇主管理的聯合身分提供者 （例如 Azure AD 或 Okta) 存取

透過 AWS 身分提供者的入口網站登入 。如果您的雲端管理員已授予您 PowerUserAccess（開發人員）許可，您會看到您有權存取 AWS 帳戶 的 和許可集。您會在許可集名稱旁，看到使用該許可集手動或以程式設計方式存取帳戶的選項。

若您自訂實作，可能會產生不同體驗，例如不同的許可集名稱。若您不確定要使用哪個許可集，請聯絡您的 IT 團隊尋求協助。

我已經 AWS 可以透過我的雇主管理的 AWS 存取入口網站存取

AWS 透過您的 AWS 存取入口網站登入 。如果您的雲端管理員已授予您 PowerUserAccess（開發人員）許可，您會看到您有權存取 AWS 帳戶 的 和許可集。您會在許可集名稱旁，看到使用該許可集手動或以程式設計方式存取帳戶的選項。

我已經能夠 AWS 透過由我的雇主管理的聯合身分自訂身分提供者存取

請聯絡您的 IT 團隊尋求協助。

步驟 2：收集您的 IAM Identity Center 資訊

取得 的存取權後 AWS，請執行下列動作來收集您的 IAM Identity Center 資訊：

1. 收集您需要執行的 SSO Start URL和 SSO Region值 aws configure sso
    a. 在您的 AWS 存取入口網站中，選取您用於開發的許可集，然後選取存取金鑰連結。
    b. 在取得登入資料對話方塊中，選擇符合您作業系統的標籤。

c. 選擇 IAM Identity Center 登入資料方法，以取得 `SSO Start URL`和 `SSO Region`值。

2. 或者，從 2.22.0 版開始，您可以使用新的發行者 URL，而不是啟動 URL。發行者 URL 位於 AWS IAM Identity Center 主控台中下列其中一個位置：
   - 在儀表板頁面上，發行者 URL 位於設定摘要中。
   - 在設定頁面上，發行者 URL 位於身分來源設定中。
3. 如需要在哪些範圍註冊值的詳細資訊，請參閱《IAM Identity Center 使用者指南》中的 OAuth 2.0 存取範圍。

步驟 3：建立 Amazon S3 儲存貯體

登入 AWS Management Console 並開啟位於 https：//https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。

在本教學課程中，請建立幾個儲存貯體以供稍後在清單中擷取。

步驟 4：安裝 AWS CLI

為您的作業系統安裝 AWS CLI 下列指示。如需詳細資訊，請參閱the section called “安裝/更新”。

安裝完成後，您可以開啟偏好的終端機並執行下列命令來驗證安裝。這應該會顯示您已安裝的 版本 AWS CLI。

```
$ aws --version
```

步驟 5：設定您的 AWS CLI 設定檔

使用下列其中一種方法設定您的設定檔

使用 **`aws configure sso`** 精靈設定您的設定檔

`config` 檔案的 `sso-session`區段用於將用於取得 SSO 存取權杖的組態變數分組，然後可用於取得 AWS 登入資料。使用下列的設定：

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

您可以定義 sso-session區段，並將其與設定檔建立關聯。sso_region 和 sso_start_url設定必須在 sso-session區段中設定。一般而言，sso_account_id 和 sso_role_name 必須在 profile 區段中設定，以讓開發套件請求 SSO 憑證。

下列範例會將開發套件設定為請求 SSO憑認證，並支援自動字符重新整理：

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

根據預設，從 2.22.0 版 AWS CLI 開始的 會使用 Code Exchange (PKCE) 授權的驗證金鑰，並且必須在具有瀏覽器的裝置上使用。若要繼續使用裝置授權，請附加 --use-device-code選項。

```
$ aws configure sso --use-device-code
```

使用 **config** 檔案手動設定

config 檔案的 sso-session區段用於將用於取得 SSO 存取權杖的組態變數分組，然後可用於取得 AWS 登入資料。使用下列的設定：

- (必要) sso_start_url
- (必要) sso_region
- sso_account_id
- sso_role_name
- sso_registration_scopes

定義 sso-session 區段並將其與設定檔相關聯。sso_region 和 sso_start_url 必須在 sso-session 區段內設定。一般而言，sso_account_id 和 sso_role_name 必須在 profile 區段中設定，以讓開發套件請求 SSO 憑證。

下列範例會將開發套件設定為請求 SSO憑認證，並支援自動字符重新整理：

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

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

身分驗證字符會以基於工作階段名稱的檔案名稱快取至 ~/.aws/sso/cache 目錄下的磁碟。

步驟 6：登入 IAM Identity Center

> **Note**
>
> 登入程序可能會提示您允許 AWS CLI 存取您的資料。由於 AWS CLI 建置在適用於 Python 的 SDK 之上，因此許可訊息可能包含botocore名稱的變化。

若要擷取和快取 IAM Identity Center 登入資料，請執行下列命令，讓 AWS CLI 開啟預設瀏覽器並驗證 IAM Identity Center 登入。

```
$ aws sso login --profile my-dev-profile
```

從 2.22.0 版開始，PKCE 授權為預設值。若要使用裝置授權登入，請新增 --use-device-code選項。

```
$ aws sso login --profile my-dev-profile --use-device-code
```

步驟 7：執行 Amazon S3 命令

若要列出您先前建立的儲存貯體，請使用 aws s3 ls命令。以下範例列出您的所有 Amazon S3 儲存貯體。

```
$ aws s3 ls
2018-12-11 17:08:50 my-bucket
2018-12-14 14:55:44 my-bucket2
```

步驟 8：登出 IAM Identity Center

使用 IAM Identity Center 設定檔完成後，請執行下列命令來刪除快取的憑證。

```
$ aws sso logout
Successfully signed out of all SSO profiles.
```

步驟 9：清除資源

完成本教學課程後，請清除您在本教學課程中建立的任何不再需要的資源，包括 Amazon S3 儲存貯體。

故障診斷

如果您使用 遇到問題 AWS CLI，請參閱 故障診斷錯誤 以取得常見的故障診斷步驟。

其他資源

其他資源如下所示。

- the section called “IAM Identity Center 概念”
- the section called “IAM Identity Center 驗證”
- the section called “安裝/更新”
- the section called “組態設定”
- `aws configure sso` 第 AWS CLI 2 版參考中的
- `aws configure sso-session` 第 AWS CLI 2 版參考中的
- `aws sso login` 第 AWS CLI 2 版參考中的
- `aws sso logout` 第 AWS CLI 2 版參考中的
- 《Amazon CodeCatalyst 使用者指南》中的設定 以 AWS CLI 搭配 CodeCatalyst 使用 CodeCatalyst
- IAM Identity Center 使用者指南中的 OAuth 2.0 存取範圍
- IAM Identity Center 使用者指南中的入門教學課程

使用 的短期登入資料進行驗證 AWS CLI

建議您設定 SDK 或工具，以延長的工作階段持續時間選項使用 IAM Identity Center 驗證。不過，您可以複製並使用 AWS 存取入口網站中提供的臨時登入資料。若這些憑證過期，便需要複製新憑證。您可以在設定檔中使用臨時憑證，或用作系統屬性和環境變數的值。

1. 登入 AWS 存取入口網站。
2. 請依照這些指示，從 AWS 存取入口網站複製 IAM 角色登入資料。
    1. 針對連結指示中的步驟 2，選擇帳戶 AWS 和 IAM 角色名稱，為您的開發需求授予存取權。此角色的名稱通常如 PowerUserAccess 或 Developer。

2. 針對步驟 4，請選取將設定檔新增至 AWS 憑證檔案，然後複製該內容。

3. 建立或開啟共用的 `credentials` 檔案。這個檔案是位於 Linux 和 macOS 系統上的 `~/.aws/credentials` 和 Windows 上的 `%USERPROFILE%\.aws\credentials`。如需詳細資訊，請參閱the section called “組態設定”。

4. 將以下文字新增至共用的 `credentials` 檔案。以您複製的憑證取代範例值。

```
[default]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token =
 IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRINGEXAMPLE
```

5. 將您偏好的預設區域和格式新增至共用的 `config` 檔案。

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

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

SDK 在建立服務用戶端時，會存取這些臨時憑證並用於每個要求。在步驟 2a 中選擇的 IAM 角色設定會決定臨時憑證的有效時長。最長持續時間為 12 小時。

每當您的憑證過期時，請重複這些步驟。

在 中使用 IAM 角色 AWS CLI

AWS Identity and Access Management (IAM) 角色是一種授權工具，可讓使用者取得額外的 （或不同的） 許可，或取得在不同 AWS 帳戶中執行動作的許可。

主題

- 先決條件
- 使用 IAM 角色概觀
- 設定和使用角色
- 使用多重要素驗證

- 跨帳戶角色和外部 ID
- 指定角色工作階段名稱以更輕鬆地稽核
- 擔任具有 Web 身分的角色
- 清除快取的憑證

先決條件

若要執行 iam 命令，您需要安裝和設定 AWS CLI。這包括設定已設定的設定檔，因為假設角色與另一個登入資料方法配對。如需詳細資訊，請參閱the section called “安裝/更新”。

使用 IAM 角色概觀

您可以透過在 ~/.aws/config 檔案中定義角色的描述檔，將 AWS Command Line Interface (AWS CLI) 設定為使用 IAM 角色。

以下範例顯示名為 marketingadmin 的角色描述檔。如果您使用 --profile marketingadmin（或使用 AWS_PROFILE 環境變數指定）執行命令，AWS CLI 會使用在個別設定檔中定義的登入資料user1來擔任具有 Amazon Resource Name (ARN) 的角色arn:aws:iam::*123456789012*:role/*marketingadminrole*。您可以執行指派給該角色的許可所允許的任何操作。

```
[profile marketingadmin]
role_arn = arn:aws:iam::123456789012:role/marketingadminrole
source_profile = user1
```

接著，您就可以指定 source_profile 來指向另一個具名描述檔，其中包含有許可使用此角色的使用者登入資料。在上述範例中，marketingadmin 描述檔使用 user1 描述檔中的憑證。當您指定 AWS CLI 命令要使用設定檔 時marketingadmin，AWS CLI 會自動查詢連結user1設定檔的登入資料，並使用它們來請求指定 IAM 角色的臨時登入資料。CLI 在背景中使用 sts:AssumeRole 操作實現這一點。然後，這些臨時憑證就會用來執行所要求的 AWS CLI 命令。指定的角色必須已連接允許執行請求 AWS CLI 命令的 IAM 許可政策。

若要從 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體或 Amazon Elastic Container Service (Amazon ECS) 容器內執行 AWS CLI 命令，您可以使用連接至執行個體設定檔或容器的 IAM 角色。如果您未指定描述檔或未設定環境變數，則會直接使用該角色。這可讓您避免將長時間存留的存取金鑰存放在您的執行個體。您也可以使用這些執行個體或容器角色來僅取得另一個角色的憑證。若要這樣做，請使用 credential_source (而不是 source_profile) 以指定如何尋找憑證。credential_source 屬性支援下列數值：

- `Environment` – 從環境變數擷取來源憑證。
- `Ec2InstanceMetadata` – 使用連接到 Amazon EC2 執行個體描述檔的 IAM 角色。
- `EcsContainer` – 使用連接到 Amazon ECS 容器的 IAM 角色。

以下範例顯示經由參考 Amazon EC2 執行個體描述檔而使用的相同 `marketingadminrole` 角色。

```
[profile marketingadmin]
role_arn = arn:aws:iam::123456789012:role/marketingadminrole
credential_source = Ec2InstanceMetadata
```

當您叫用角色時，您會有其他選項可以取得，例如使用多重要素驗證、外部 ID (由第三方公司用來存取其用戶端資源)。您也可以指定可在 AWS CloudTrail 日誌中更輕鬆地稽核的唯一角色工作階段名稱。

設定和使用角色

當您使用指定 IAM 角色的設定檔執行命令時， AWS CLI 會使用來源設定檔的登入資料來呼叫 AWS Security Token Service (AWS STS) 並請求指定角色的臨時登入資料。來源描述檔中的使用者必須有許可針對指定的描述檔中的角色呼叫 `sts:assume-role`。此角色必須有信任關係而允許來源描述檔中的使用者使用該角色。針對角色來擷取並使用臨時憑證的過程就稱為擔任角色。

在 IAM 中建立一個角色，並使用您希望使用者按照《AWS Identity and Access Management 使用者指南》中的建立角色以將許可委派給 IAM 使用者下的過程來承擔的許可。如果角色和來源描述檔的 使用者在同一帳戶中，則您可以在設定角色的信任關係時輸入自己的帳戶 ID。

建立角色後，修改信任關係以允許 使用者承擔此角色。

以下範例顯示可連接到角色的信任政策。此政策允許帳戶 (123456789012) 中的任何使用者擔任角色，但前提是該帳戶的管理員明確將 `sts:AssumeRole` 許可授予使用者。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
```

```
      "Action": "sts:AssumeRole"
    }
  ]
}
```

信任政策不會實際授予許可。帳戶的管理員必須連結含有適當許可的政策，才能將擔任角色的許可委派給個別使用者。以下範例顯示可連接到使用者的政策，該政策只允許使用者擔任 marketingadminrole 角色。如需有關授予使用者擔任角色之權限的詳細資訊，請參閱《IAM 使用者指南》中的授予使用者許可以切換角色。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::123456789012:role/marketingadminrole"
    }
  ]
}
```

使用者不需要額外的許可，即可使用角色描述檔執行 AWS CLI 命令。反之，執行命令所需的許可來自於連接到角色的許可。您可以將許可政策連接至角色，以指定可以對哪些 AWS 資源執行哪些動作。如需有關將許可附加至角色 (運作方式與使用者相同) 的詳細資訊，請參閱《IAM 使用者指南》中的變更 IAM 使用者的許可。

既然您已正確設定角色描述檔、角色許可、角色信任關係和使用者許可，您可以在命令列叫用 --profile 選項來使用該角色。例如，以下使用本主題開頭範例所定義的 marketingadmin 角色所連接的許可，呼叫 Amazon S3 ls 命令。

```
$ aws s3 ls --profile marketingadmin
```

若要使用多個呼叫的角色，可以從命令列為目前工作階段設定 AWS_PROFILE 環境變數。因為已定義環境變數，您不需要在每個命令中指定 --profile 選項。

Linux 或 macOS

```
$ export AWS_PROFILE=marketingadmin
```

Windows

```
C:\> setx AWS_PROFILE marketingadmin
```

如需設定使用者和角色的詳細資訊，請參閱《IAM 使用者指南》中的 IAM 身分（使用者、使用者群組和角色）和 IAM 角色。

使用多重要素驗證

為了提高安全性，當使用者嘗試使用角色設定檔進行呼叫時，您可以要求使用者提供由多重要素驗證 (MFA) 裝置、U2F 裝置或行動應用程式產生的一次性金鑰。

首先，您可以選擇修改 IAM 角色的信任關係來要求 MFA。這可防止任何人在使用該角色之前未先使用 MFA 進行驗證。相關範例請參閱下列範例中的 Condition 行。此政策可讓名為 anika 的使用者擔任政策連接的角色，前提是使用者已使用 MFA 進行驗證。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::123456789012:user/anika" },
      "Action": "sts:AssumeRole",
      "Condition": { "Bool": { "aws:multifactorAuthPresent": true } }
    }
  ]
}
```

接著，在角色設定檔中新增一行，指定使用者的 MFA 裝置的 ARN。以下範例 config 檔案項目顯示兩個角色設定檔，都使用使用者 anika 的存取金鑰來請求角色 cli-role 的臨時登入資料。使用者 anika 具有擔任角色的許可，由角色的信任政策所授予。

```
[profile role-without-mfa]
```

```
region = us-west-2
role_arn= arn:aws:iam::128716708097:role/cli-role
source_profile=cli-user

[profile role-with-mfa]
region = us-west-2
role_arn= arn:aws:iam::128716708097:role/cli-role
source_profile = cli-user
mfa_serial = arn:aws:iam::128716708097:mfa/cli-user

[profile cli-user]
region = us-west-2
output = json
```

`mfa_serial` 設定可以採取 ARN，如圖所示，或硬體 MFA 字符的序號。

第一個設定檔 `role-without-mfa` 不需要 MFA。不過，因為連接到該角色的上述範例信任政策要求 MFA，因此使用此設定檔來執行命令的任何嘗試都會失敗。

```
$ aws iam list-users --profile role-without-mfa

An error occurred (AccessDenied) when calling the AssumeRole operation: Access denied
```

第二個設定檔項目 `role-with-mfa` 識別要使用的 MFA 裝置。當使用者嘗試使用此設定檔執行 AWS CLI 命令時， 會 AWS CLI 提示使用者輸入 MFA 裝置提供的一次性密碼 (OTP)。如果 MFA 驗證成功，命令會執行請求操作。OTP 不會顯示在螢幕上。

```
$ aws iam list-users --profile role-with-mfa
Enter MFA code for arn:aws:iam::123456789012:mfa/cli-user:
{
    "Users": [
        {
            ...
```

跨帳戶角色和外部 ID

您可以藉由將角色設定為跨帳戶角色，使 使用者能夠使用屬於不同帳戶的角色。在角色建立期間，將角色類型設定為 Another AWS account (其他 AWS 帳戶)，如建立角色以將許可委派給 IAM 使用者所述。或者，您可以選取 Require MFA (需要 MFA)。如 使用多重要素驗證 中所描述的，Require MFA (需要 MFA) 選項在信任關係中設定適當的條件。

如果您使用外部 ID 以進一步控制誰可使用跨帳戶的角色，則您還必須將 external_id 參數新增至角色描述檔。這通常是在另一個帳戶由公司或組織外的某人控制時才使用。

```
[profile crossaccountrole]
role_arn = arn:aws:iam::234567890123:role/SomeRole
source_profile = default
mfa_serial = arn:aws:iam::123456789012:mfa/saanvi
external_id = 123456
```

指定角色工作階段名稱以更輕鬆地稽核

當多人共用一個角色時，稽核就更具挑戰。您想要將呼叫的每個操作與呼叫該動作的個人建立關聯。不過，當該個人使用某個角色時，個人擔任的角色會是與呼叫操作不同的動作，因此您必須手動將兩者建立關聯。

您可以在使用者擔任角色時指定唯一的角色工作階段名稱，藉此簡化此操作。您可以將 role_session_name 參數新增至指定角色的 config 檔案中的每個具名描述檔。role_session_name 值會傳遞到 AssumeRole 操作，並成為角色工作階段的 ARN 的一部分。它也包含在所有記錄操作的 AWS CloudTrail 日誌中。

例如，您可以建立以角色為基礎的設定檔，如下所示。

```
[profile namedsessionrole]
role_arn = arn:aws:iam::234567890123:role/SomeRole
source_profile = default
role_session_name = Session_Maria_Garcia
```

這會導致角色工作階段中具有下列 ARN。

```
arn:aws:iam::234567890123:assumed-role/SomeRole/Session_Maria_Garcia
```

此外，所有 AWS CloudTrail 日誌都會在每個操作擷取的資訊中包含角色工作階段名稱。

擔任具有 Web 身分的角色

您可以設定設定檔，以指示 AWS CLI 應該使用 Web 聯合身分和 Open ID Connect (OIDC) 擔任角色。當您在設定檔中指定此項目時， AWS CLI 會自動為您進行對應的 AWS STS AssumeRoleWithWebIdentity呼叫。

> **Note**
>
> 當您指定使用 IAM 角色的設定檔時， AWS CLI 會進行適當的呼叫來擷取臨時登入資料。這些憑證會儲存在 ~/.aws/cli/cache 中。指定相同設定檔的後續 AWS CLI 命令會使用快取的暫時登入資料，直到過期為止。此時， AWS CLI 會自動重新整理登入資料。

若要使用 Web 聯合身分擷取並使用臨時憑證，您可以在共用描述檔中指定以下組態值。

role_arn

指定要擔任的角色 ARN。

web_identity_token_file

指定檔案的路徑，該檔案包含由身分提供者提供的 OAuth 2.0 存取字符或 OpenID Connect ID 字符。 AWS CLI 會載入此檔案，並傳遞其內容作為 `WebIdentityToken` 操作的 `AssumeRoleWithWebIdentity` 引數。

role_session_name

指定套用到此擔任角色工作階段的選用名稱。

下列是擔任具有 Web 身分角色描述檔所需的最少量組態的組態範例。

```
# In ~/.aws/config

[profile web-identity]
role_arn=arn:aws:iam:123456789012:role/RoleNameToAssume
web_identity_token_file=/path/to/a/token
```

您也可以使用 環境變數 提供此組態。

AWS_ROLE_ARN

要擔任的角色 ARN。

AWS_WEB_IDENTITY_TOKEN_FILE

Web 身分字符檔案的路徑。

AWS_ROLE_SESSION_NAME

套用到此擔任角色工作階段的名稱。

> **Note**
>
> 這些環境變數目前只適用於擔任具有 Web 身分提供者的角色。它們不適用於一般擔任角色供應商組態。

清除快取的憑證

當您使用角色時， 會在本機 AWS CLI 快取臨時登入資料，直到過期為止。下次嘗試使用它們時，AWS CLI 會嘗試代表您續約它們。

如果角色的臨時憑證被撤銷，則不會自動續約，而嘗試使用它們會失敗。不過，您可以刪除快取，強制 AWS CLI 擷取新的登入資料。

Linux 或 macOS

```
$ rm -r ~/.aws/cli/cache
```

Windows

```
C:\> del /s /q %UserProfile%\.aws\cli\cache
```

使用 的 IAM 使用者登入資料進行驗證 AWS CLI

> **Warning**
>
> 為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 AWS IAM Identity Center。

本節說明如何以 IAM 使用者進行基本設定。其中包含使用 `config` 和 `credentials` 檔案的安全憑證。要改為查看 AWS IAM Identity Center的配置說明，請參閱 the section called “IAM Identity Center 驗證”。

主題

- 步驟 1：建立 IAM 使用者
- 步驟 2：取得您的存取金鑰

- 設定 AWS CLI
  - 使用 aws configure
  - 透過 .CSV 檔案匯入存取金鑰
  - 直接編輯 config 和 credentials 檔案

步驟 1：建立 IAM 使用者

按照 IAM 使用者指南中的建立 IAM 使用者 (主控台) 程序來建立 IAM 使用者。

- 針對權限選項，請選擇直接連接政策做為您指派權限給此使用者的方式。
- 多數「入門」SDK 教學都使用 Amazon S3 服務做為範例。若要讓應用程式能夠完整存取 Amazon S3，請選取要連接至此使用者的 `AmazonS3FullAccess` 政策。

步驟 2：取得您的存取金鑰

1. 登入 AWS Management Console ，並在 https://console.aws.amazon.com/iam/ ：//www. 開啟 IAM 主控台。
2. 在 IAM 主控台的導覽窗格中，選取使用者，然後選取您先前建立使用者的 **`User name`**。
3. 在使用者頁面上，選取安全憑證頁面。接著，在存取金鑰下，選取建立存取金鑰。
4. 針對建立存取金鑰步驟 1，請選擇命令列界面 (CLI)。
5. 在建立存取金鑰步驟 2 中，輸入選用標籤並選取下一步。
6. 在建立存取金鑰步驟 3 中，選取下載 .csv 檔案，以儲存包含 IAM 使用者存取金鑰和私密存取金鑰的 `.csv` 檔案。您之後將會用到此資訊。
7. 選取完成。

設定 AWS CLI

對於一般用途， AWS CLI 需要下列資訊：

- 存取金鑰 ID
- 私密存取金鑰
- AWS 區域
- 輸出格式

會將此資訊 AWS CLI 儲存在credentials名為 的設定檔 （設定集合） default 中。根據預設，當您執行未明確指定要使用之設定檔的 AWS CLI 命令時，會使用此設定檔中的資訊。如需有關 credentials 檔案的詳細資訊，請參閱中的組態和登入資料檔案設定 AWS CLI。

若要設定 AWS CLI，請使用下列其中一個程序：

主題

- 使用 aws configure
- 透過 .CSV 檔案匯入存取金鑰
- 直接編輯 config 和 credentials 檔案

使用 aws configure

對於一般用途， aws configure命令是設定 AWS CLI 安裝的最快方式。此設定精靈會針對您要使用的每項資訊給予提示。除非使用 --profile選項另有指定，否則 會將此資訊 AWS CLI 存放在 default 設定檔中。

下列範例使用範例值來設定 default 設定檔。將範本值取代為您自己的值，如下列章節所述。

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

下列範例使用範例值來設定名為 userprod 的設定檔。將範本值取代為您自己的值，如下列章節所述。

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

透過 .CSV 檔案匯入存取金鑰

您可以匯入建立存取金鑰後下載的純文字.csv檔案，而不是使用 aws configure輸入存取金鑰。

.csv 檔案必須包含以下標頭。

- User Name (使用者名稱) - 此欄必須加入您的 .csv。這用於建立匯入時用於 config和 credentials 檔案的設定檔名稱。
- 存取金鑰 ID
- 私密存取金鑰

> **Note**
>
> 在初始存取金鑰建立過程中，一旦關閉下載 .csv 檔案對話方塊中，您將無法在關閉對話方塊之後存取私密存取金鑰。如果您需要 .csv 檔案，您必須自己建立一個包含所需標頭和您所儲存存取金鑰資訊的檔案。如果您無法存取您的存取金鑰資訊，則必須建立新的存取金鑰。

若要匯入 .csv 檔案，請使用 aws configure import 命令與 --csv 選項，如以下所示：

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

如需詳細資訊，請參閱aws_configure_import。

直接編輯 config 和 credentials 檔案

若要直接編輯 config 和 credentials 檔案，請執行下列動作。

1. 建立或開啟共用的 AWS credentials 檔案。這個檔案是位於 Linux 和 macOS 系統上的 ~/.aws/credentials 和 Windows 上的 %USERPROFILE%\.aws\credentials。如需詳細資訊，請參閱the section called “組態設定”。
2. 將以下文字新增至共用的 credentials 檔案。取代您先前下載 .csv 檔案中的範例值，然後儲存檔案。

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

在 中使用 Amazon EC2 執行個體中繼資料做為登入資料 AWS CLI

當您 AWS CLI 從 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體內執行 時，您可以簡化將登入資料提供給命令的程序。每個 Amazon EC2 執行個體都包含中繼資料，可供 AWS CLI 直接查詢

臨時憑證。當 IAM 角色連接至執行個體時， AWS CLI 會自動且安全地從執行個體中繼資料擷取登入資料。

若要停用此服務，請使用 AWS_EC2_METADATA_DISABLED 環境變數。

主題

- 先決條件
- 設定用於 Amazon EC2 中繼資料的設定檔

先決條件

若要搭配 使用 Amazon EC2 登入資料 AWS CLI，您需要完成下列操作：

- 安裝及設定 AWS CLI。如需詳細資訊，請參閱the section called “安裝/更新”及的身分驗證和存取憑證 AWS CLI。
- 您瞭解組態檔案和命名設定檔。如需詳細資訊，請參閱中的組態和登入資料檔案設定 AWS CLI。
- 您已建立可存取所需資源的 AWS Identity and Access Management (IAM) 角色，並在啟動該角色時將該角色連接到 Amazon EC2 執行個體。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Amazon EC2 的 IAM 政策，以及《IAM 使用者指南》中的授予在 Amazon EC2 執行個體上執行的應用程式存取 AWS 資源。 Amazon EC2

設定用於 Amazon EC2 中繼資料的設定檔

若要指定您想要使用託管 Amazon EC2 執行個體設定檔中可用的憑證，請在組態檔案中的具名設定檔使用下列語法。如需詳細說明，請參閱下列步驟。

```
[profile profilename]
role_arn = arn:aws:iam::123456789012:role/rolename
credential_source = Ec2InstanceMetadata
region = region
```

1. 在組態檔案中建立設定檔。

   ```
   [profile profilename]
   ```

2. 新增可存取所需資源的 IAM ARN 角色。

```
role_arn = arn:aws:iam::123456789012:role/rolename
```

3. 指定 Ec2InstanceMetadata 作為憑證來源。

```
credential_source = Ec2InstanceMetadata
```

4. 設定您的區域。

```
region = region
```

範例

下列範例會假設 *marketingadminrole* 角色，並在名為 *marketingadmin* 的 Amazon EC2 執行個體描述檔中使用 *us-west-2* 區域。

```
[profile marketingadmin]
role_arn = arn:aws:iam::123456789012:role/marketingadminrole
credential_source = Ec2InstanceMetadata
region = us-west-2
```

在 中使用外部程序來採購登入資料 AWS CLI

> **Warning**
>
> 本主題討論從外部程序取得憑證。如果用於產生憑證的命令可由未經核准的程序或使用者所存取，這可能會有安全風險。我們建議您使用 提供的支援、安全替代方案 AWS CLI AWS ，並降低入侵憑證的風險。務必保護 config 檔案及任何支援檔案和工具，以防止洩露。
> 請確定您的自訂登入資料工具不會將任何秘密資訊寫入 ，StdErr因為 SDKs和 AWS CLI 可以擷取並記錄這類資訊，進而可能將其公開給未經授權的使用者。

如果您有方法可以產生或查詢 未直接支援的登入資料 AWS CLI，您可以設定 AWS CLI ，以透過在 config 檔案中設定 credential_process來使用登入資料。

例如，您在 config 檔案中可能包含類似以下的項目。

```
[profile developer]
```

```
credential_process = /opt/bin/awscreds-custom --username helen
```

語法

若要以與任何作業系統相容的方式建立此字串，請遵循這些規則：

- 如果路徑或檔案名稱包含空格，完整的路徑和檔案名稱請以雙引號 (「」) 括住。路徑和檔案名稱只能包含字元：A-Z a-z 0-9 – _. 空格
- 如果參數名稱或參數值包含空格，則該元素請以雙引號 (「」) 括住。僅括住名稱或值，而非對組。
- 請勿在字串中包含任何環境變數。例如，您無法包含 $HOME 或 %USERPROFILE%。
- 請勿將主資料夾指定為 ~。您必須指定完整路徑。

適用於 Windows 的範例

```
credential_process = "C:\Path\To\credentials.cmd" parameterWithoutSpaces "parameter
 with spaces"
```

適用於 Linux 或 macOS 的範例

```
credential_process = "/Users/Dave/path/to/credentials.sh" parameterWithoutSpaces
 "parameter with spaces"
```

登入資料程式的預期輸出

會執行設定檔中指定的 AWS CLI 命令，然後從 讀取資料STDOUT。您指定的命令必須在 STDOUT 上產生符合以下語法的 JSON 輸出。

```
{
  "Version": 1,
  "AccessKeyId": "an AWS access key",
  "SecretAccessKey": "your AWS secret access key",
  "SessionToken": "the AWS session token for temporary credentials",
  "Expiration": "ISO8601 timestamp when the credentials expire"
}
```

> **Note**
>
> 截至本文編寫時，Version 索引鍵必須設定為 1。這可能隨著結構演進而逐漸遞增。

`Expiration` 索引鍵是 ISO8601 格式的時間戳記。如果 `Expiration` 索引鍵未出現在工具的輸出中，CLI 會假設憑證是不重新整理的長期憑證。否則，憑證就視為臨時憑證，在過期之前可透過重新執行 `credential_process` 命令來自動重新整理。

> **Note**
>
> AWS CLI 不會像擔任角色登入資料一樣快取外部程序登入資料。如果需要快取，您必須在外部程序中實作它。

外部程序可能傳回非零傳回碼，以表示擷取憑證時發生錯誤。

使用 AWS CLI

本節提供 AWS Command Line Interface (AWS CLI) 中一般使用、常見功能和選項的完整概觀，超出組態the section called “端點”一節涵蓋的詳細資訊。

本指南深入探討撰寫 AWS CLI 命令的基本層面，包括其基本結構、格式化和篩選功能。透過了解這些核心元素，您將能夠建構可精確鎖定所需資源和動作的命令，而無需導覽複雜的 Web 主控台。

此外，這會反白顯示 可用的說明內容和文件 AWS CLI。從內建命令列說明到完整的第 AWS CLI 2 版參考指南，您將可以存取資訊，協助您探索 的功能 AWS CLI。

如需 AWS 服務 特定範例和使用案例，請參閱 代碼範例或第 AWS CLI 2 版參考指南。這些提供命令特定資訊，並示範如何將 AWS CLI 用於各種 的範例 AWS 服務。

> **Note**
>
> 根據預設， AWS 服務 會在 TCP 連接埠 443 上使用 HTTPS 將請求 AWS CLI 傳送至 。若要確保成功使用 AWS CLI，您必須能夠在此連接埠上進行傳出連線。

本指南的主題

- 存取 的說明和資源 AWS CLI
- 中的命令結構 AWS CLI
- 在 中指定參數值 AWS CLI
- 在 中啟用和使用命令提示 AWS CLI
- 在 中控制命令輸出 AWS CLI
- 中的命令列傳回代碼 AWS CLI
- 使用自訂精靈在 中執行互動式命令 AWS CLI
- 在 中建立和使用別名 AWS CLI

存取 的說明和資源 AWS CLI

本主題說明如何存取 AWS Command Line Interface () 的說明內容AWS CLI。

主題

- 內建 AWS CLI 說明命令

- AWS CLI 參考指南
- API 文件
- 故障診斷錯誤
- 其他協助

內建 AWS CLI 說明命令

使用 AWS Command Line Interface () 時，您可以取得任何命令的說明AWS CLI。若要這樣做，只需在命令名稱結尾輸入 `help`。

例如，下列命令會顯示一般 AWS CLI 選項和可用頂層命令的說明。

```
$ aws help
```

以下命令會顯示可用的 Amazon Elastic Compute Cloud (Amazon EC2) 特定命令。

```
$ aws ec2 help
```

以下範例會顯示 Amazon EC2 `DescribeInstances` 操作的詳細說明。說明包括輸入參數的描述、可用的篩選條件以及輸出包含的內容。此外也包含範例，顯示如何輸入命令的常見變化。

```
$ aws ec2 describe-instances help
```

各個命令的說明均分為六個區段：

名稱

命令的名稱。

```
NAME
       describe-instances -
```

描述

命令叫用之 API 操作的描述。

```
DESCRIPTION
       Describes one or more of your instances.
```

```
       If you specify one or more instance IDs, Amazon EC2 returns information
       for those instances. If you do not specify  instance  IDs,  Amazon  EC2
       returns  information  for  all  relevant  instances.  If you specify an
       instance ID that is not valid, an error is returned. If you specify  an
       instance  that  you  do  not  own,  it  is not included in the returned
       results.
...
```

概要

使用命令和其選項的基本語法。如果有選項顯示在方括號內，代表為選用、具有預設值，或是具有可使用的替代選項。

```
SYNOPSIS
            describe-instances
          [--dry-run | --no-dry-run]
          [--instance-ids <value>]
          [--filters <value>]
          [--cli-input-json <value>]
          [--starting-token <value>]
          [--page-size <value>]
          [--max-items <value>]
          [--generate-cli-skeleton]
```

例如，`describe-instances` 具有預設行為，可描述目前帳戶和 AWS 區域中的所有執行個體。您可選擇指定 `dry-run` 清單來描述一個或多個執行個體。`instance-ids` 是一個選用的布林值旗標，其不採用數值。若要使用布林值旗標，請指定顯示數值，在此範例中為 `--dry-run` 或 `--no-dry-run`。同樣地，`--generate-cli-skeleton` 不採用數值。如果選項的使用具有條件，則這些條件會在 OPTIONS 區段中說明，或顯示在範例中。

選項

概要中顯示之各個選項的說明。

```
OPTIONS
       --dry-run | --no-dry-run (boolean)
          Checks whether you have the required  permissions  for  the  action,
          without actually making the request, and provides an error response.
          If you have the required permissions, the error response is  DryRun-
          Operation . Otherwise, it is UnauthorizedOperation .

       --instance-ids (list)
          One or more instance IDs.
```

```
          Default: Describes all your instances.
...
```

範例

命令使用及其選項的範例。如果沒有範例適用於您需要的命令或使用案例，請使用此頁面上的意見回饋連結請求，或在 AWS CLI 命令說明頁面上的命令參考中請求。

```
    EXAMPLES
    To describe an Amazon EC2 instance

    Command:

    aws ec2 describe-instances --instance-ids i-5203422c

    To describe all instances with the instance type m1.small

    Command:

    aws ec2 describe-instances --filters "Name=instance-type,Values=m1.small"

    To describe all instances with an Owner tag

    Command:

    aws ec2 describe-instances --filters "Name=tag-key,Values=Owner"
...
```

輸出

來自 AWS的回應中所包含各個欄位與資料類型的說明。

對於 `describe-instances`，輸出是保留物件的清單，每個清單都包含若干欄位和物件，而這些欄位和物件包含與其關聯之執行個體的資訊。此資訊來自 Amazon EC2 所使用的保留資料類型的 API 文件。

```
OUTPUT
       Reservations -> (list)
          One or more reservations.

          (structure)
              Describes a reservation.
```

```
            ReservationId -> (string)
               The ID of the reservation.

            OwnerId -> (string)
               The ID of the AWS account that owns the reservation.

            RequesterId -> (string)
               The ID of the requester that launched the instances  on  your
               behalf (for example, AWS Management Console or Auto Scaling).

            Groups -> (list)
               One or more security groups.

               (structure)
                   Describes a security group.

                   GroupName -> (string)
                      The name of the security group.

                   GroupId -> (string)
                      The ID of the security group.

            Instances -> (list)
               One or more instances.

               (structure)
                   Describes an instance.

                   InstanceId -> (string)
                      The ID of the instance.

                   ImageId -> (string)
                      The ID of the AMI used to launch the instance.

                   State -> (structure)
                      The current state of the instance.

                      Code -> (integer)
                          The  low  byte represents the state. The high byte
                          is an opaque internal value and should be ignored.
...
```

當 將輸出 AWS CLI 轉譯為 JSON 時，它會成為保留物件的陣列，類似下列範例。

```
{
    "Reservations": [
        {
            "OwnerId": "012345678901",
            "ReservationId": "r-4c58f8a0",
            "Groups": [],
            "RequesterId": "012345678901",
            "Instances": [
                {
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "PublicDnsName": "ec2-52-74-16-12.us-
west-2.compute.amazonaws.com",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
...
```

每個保留物件均包含說明該保留的欄位，以及執行個體物件列陣，每個執行個體物件都有提供說明的專屬欄位 (例如 `PublicDnsName`) 和物件 (例如 `State`)。

> **Windows 使用者**
>
> 您可將說明命令的輸出輸送 (|) 到 `more` 命令，以便一次檢視一個說明檔案頁面。按下空格鍵或 PgDn 鍵檢視更多文件內容，按 **q** 鍵即可退出。
>
> ```
> C:\> aws ec2 describe-instances help | more
> ```

AWS CLI 參考指南

說明檔案包含連結，無法從命令列檢視或瀏覽。您可以使用線上第 第 AWS CLI 2 版參考指南，檢視這些連結並與之互動。參考也包含所有 AWS CLI 命令的說明內容。提供的說明可在行動裝置、平板電腦或桌上型電腦螢幕上輕鬆導覽和檢視。

API 文件

中的所有命令 AWS CLI 都會對應至對 AWS 服務的公有 API 提出的請求。每個具備公有 API 的服務均有 API 參考，這些文件可以在 AWS 文件網站上該服務的首頁中找到。API 參考的內容根據 API 的建構方式和所用協議而有所差異。一般來說，API 參考包含詳細資訊，說明 API 支援的操作、傳送到服務或從服務傳送的資料，以及服務可能報告的任何錯誤情況。

API 文件區段

- 動作 – 關於每個操作及其參數 (包含長度限制或內容限制以及預設值) 的詳細資訊。它會列出此操作可能發生的錯誤。每個操作都會對應至 中的子命令 AWS CLI。
- 資料類型 – 結構的詳細資訊，命令可能需要這些結構作為參數，或傳回以回應請求。
- 常用參數 – 服務的所有動作共用之參數的詳細資訊。
- 常見錯誤 – 服務的任何操作可能傳回之錯誤的詳細資訊。

每個區段的名稱或可用性，會依據服務而有所差異。

> **服務特定的 CLI**
>
> 某些服務具有單獨的 CLI，其日期從建立單一 AWS CLI 之前起算，以使用所有 服務。這些服務特定的 CLI 具有個別專屬的文件，可以從該服務的文件頁面中連結取得。服務特定 CLIs 的文件不適用於 AWS CLI。

故障診斷錯誤

如需診斷和修正 AWS CLI 錯誤的說明，請參閱 故障診斷錯誤。

其他協助

如需 AWS CLI 問題的其他協助，請造訪 GitHub 上的 AWS CLI 社群。

中的命令結構 AWS CLI

本主題涵蓋如何建構 AWS Command Line Interface (AWS CLI) 命令，以及如何使用等待命令。

主題

- 命令結構

- 等候命令

命令結構

在命令列 AWS CLI 上使用分段結構，必須依此順序指定：

1. 對 aws 程式的基本呼叫。
2. 最上層命令，通常對應至 支援的 AWS 服務 AWS CLI。
3. 指定執行哪些操作的子命令。
4. 操作所需的一般 AWS CLI 選項或參數。只要位在前三個部分後面，您可以依任意順序指定這些項目。如果排他參數受到多次指定，只會套用最終數值。

```
$ aws <command> <subcommand> [options and parameters]
```

參數採用各種類型的輸入數值，例如數字、字串、清單、映射和 JSON 結構。支援哪些項目取決於您指定的命令和子命令。

範例

Amazon Simple Storage Service (Amazon S3)

以下範例列出您的所有 Amazon S3 儲存貯體。

```
$ aws s3 ls
2018-12-11 17:08:50 amzn-s3-demo-bucket1
2018-12-14 14:55:44 amzn-s3-demo-bucket2
```

如需 Amazon S3 命令的詳細資訊，請參閱 AWS CLI 命令參考中的 aws s3。

AWS CloudFormation

以下 create-change-set 命令範例會將 cloudformation 堆疊名稱變更為 *my-change-set*。

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set-name my-
change-set
```

如需 AWS CloudFormation 命令的詳細資訊，請參閱《 AWS CLI 命令參考aws cloudformation》中的 。

等候命令

有些 AWS 服務有可用的wait命令。任何使用 aws wait 的命令通常都會等到命令完成後，再進行下一個步驟。這對於分段命令或指令碼特別有用，因為如果等候命令失敗，您可以使用等候命令來避免進行後續步驟。

對於必須依此順序指定的wait命令， 會在命令列 AWS CLI 上使用分段結構：

1. 對 aws 程式的基本呼叫。
2. 最上層命令，通常對應至 支援的 AWS 服務 AWS CLI。
3. wait 命令。
4. 指定執行哪些操作的子命令。
5. 操作所需的一般 CLI 選項或參數。只要位在前三個部分後面，您可以依任意順序指定這些項目。如果排他參數受到多次指定，只會套用最終數值。

```
$ aws <command> wait <subcommand> [options and parameters]
```

參數採用各種類型的輸入數值，例如數字、字串、清單、映射和 JSON 結構。支援哪些項目取決於您指定的命令和子命令。

> **Note**
>
> 並非所有 AWS 服務都支援 wait 命令。請參閱第 AWS CLI 2 版參考指南，查看您的服務是否支援 wait 命令。

範例

AWS CloudFormation

以下 wait change-set-create-complete 命令範例會暫停，且僅在可確認 *my-stack* 堆疊中的 *my-change-set* 變更集已準備好執行後，才會繼續執行。

```
$ aws cloudformation wait change-set-create-complete --stack-name my-stack --change-set-name my-change-set
```

如需 AWS CloudFormation wait 命令的詳細資訊，請參閱 AWS CLI 命令參考中的 wait。

AWS CodeDeploy

以下 `wait deployment-successful` 命令範例會暫停，直到 *d-A1B2C3111* 成功完成部署。

```
$ aws deploy wait deployment-successful --deployment-id d-A1B2C3111
```

如需 AWS CodeDeploy `wait` 命令的詳細資訊，請參閱 AWS CLI 命令參考中的 `wait`。

在 中指定參數值 AWS CLI

AWS Command Line Interface (AWS CLI) 中使用的許多參數都是簡單的字串或數值，例如`my-key-pair`下列`aws ec2 create-key-pair`命令範例中的金鑰對名稱。

```
$ aws ec2 create-key-pair --key-name my-key-pair
```

命令的格式可能因終端機而異。例如，大多數終端機是區分大小寫的，但 Powershell 不區分大小寫。這代表以下兩個命令範例會為區分大小寫的終端產生不同的結果，因為它們將 `MyFile*.txt` 和 `myfile*.txt` 視為不同的參數。

但是，PowerShell 會處理這些請求，因為它會將 `MyFile*.txt` 和 `myfile*.txt`視作相同的參數。下列命令範例使用 `aws s3 cp`命令示範這些參數：

```
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "MyFile*.txt"
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "myfile*.txt"
```

有關 PowerShell 不區分大小寫的詳細資訊，請參閱 PowerShell 文件中的關於不區分大小寫。

有時您需要在包含特殊或空格字元的字串周圍使用引號或文字。有關此格式的規則也可能在終端機之間有所不同。如需有關在複雜參數周圍使用引號的詳細資訊，請參閱 在 中使用引號和常值搭配字串 AWS CLI。

這些主題涵蓋最常見的終端機格式化規則。如果您的終端機辨識參數值時發生問題，請務必檢閱本節中的主題，並檢查終端機的文件是否有其特定的語法規則。

參數主題

- 中的常見參數類型 AWS CLI
- 在 中使用引號和常值搭配字串 AWS CLI
- 從 中的檔案載入參數 AWS CLI

- AWS CLI 中的骨架和輸入檔案 AWS CLI
- 在 中使用速記語法 AWS CLI

中的常見參數類型 AWS CLI

本節說明一些常見的參數類型和典型必要格式。

如果您對特定命令的參數格式有問題，請在命令名稱後輸入 **help** 來檢視說明。每個子命令的幫助包括選項的名稱和說明。該選項的參數類型在括弧中列出。如需檢視說明的詳細資訊，請參閱 the section called “取得協助”。

參數類型包含：

- 字串
- 時間戳記
- 清單
- Boolean
- Integer
- Binary/Blob (二進位大型物件)和串流 Blob
- Map
- 文件

字串

字串參數可以包含 ASCII 字元集的英數字元、符號和空格。包含空格的字串必須用引號括起來。我們建議您不要使用標準空格字元以外的符號和空格，並觀察終端機的引用規則以防止意外結果。

某些字串參數可以接受檔案中的二進位資料。如需範例，請參閱二進位檔案。

時間戳記

根據 ISO 8601 標準對時間戳記進行格式化。這些通常稱為「DateTime」或「Date」參數。

```
$ aws ec2 describe-spot-price-history --start-time 2014-10-13T19:00:00Z
```

可接受的格式包括：

- *YYYY*-*MM*-*DD*T*hh*:*mm*:*ss.sssTZD (UTC)*，例如 2014-10-01T20:30:00.000Z
- *YYYY*-*MM*-*DD*T*hh*:*mm*:*ss.sssTZD (####)*，例如 2014-10-01T12:30:00.000-08:00
- *YYYY*-*MM*-*DD*，例如 2014-10-01
- Unix 時間 (以秒為單位)，例如 1412195400。這有時稱為 Unix Epoch 時間，表示為 1970 年 1 月 1 日午夜 (UTC 時間) 以來的秒數。

根據預設， AWS CLI 版本 2 會將所有回應 DateTime 值轉譯為 ISO 8601 格式。

您可以透過使用 `cli_timestamp_format` 檔案設定來設定時間戳記格式。

清單

由空格分隔的一個或多個字串。如果任何字串項目包含空格，就必須使用引號括住該項目。遵守您的終端機的引號規則以防止意外結果。

```
$ aws ec2 describe-spot-price-history --instance-types m1.xlarge m1.medium
```

Boolean

可開啟或關閉選項的二進位標記。例如，`ec2 describe-spot-price-history` 有一個布林值 `--dry-run` 參數，當指定時，可以在不實際執行查詢的情況下針對服務驗證查詢。

```
$ aws ec2 describe-spot-price-history --dry-run
```

輸出表明該命令的格式是否正確。此命令還包含一個 `--no-dry-run` 版本的參數，可用於明確指示該命令應該正常執行。包括它並不是必須的，因為這是預設行為。

Integer

一個不帶正負號的整數。

```
$ aws ec2 describe-spot-price-history --max-items 5
```

Binary/Blob (二進位大型物件)和串流 Blob

在 中 AWS CLI，您可以直接在命令列上以字串形式傳遞二進位值。Blob 有兩種：

- Blob

- 串流 Blob

Blob

若要將值傳遞至類型為 blob 的參數，您必須使用 fileb:// 字首為包含二進位資料的本機檔案指定路徑。使用 fileb:// 字首參考的檔案一律視為原始未編碼的二進位。指定的路徑會解譯為相對於目前的工作目錄。例如，aws kms encrypt 的 --plaintext 參數是一個 blob。

```
$ aws kms encrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --plaintext fileb://ExamplePlaintextFile \
    --output text \
    --query CiphertextBlob | base64 \
    --decode > ExampleEncryptedFile
```

> **Note**
>
> 為了向後相容性，您可以使用 file:// 字首。根據檔案設定 cli_binary_format 或 --cli-binary-format 命令列選項，會使用兩種格式：
>
> - 第 2 AWS CLI 版的預設值。如果設定值為 base64，使用 file:// 字首參考的檔案會視為 base64 編碼的文字。
> - 第 1 AWS CLI 版的預設值。如果設定的值為 raw-in-base64-out，則會將使用 file:// 字首參照的檔案讀取為文字，然後 AWS CLI 嘗試將其編碼為二進位。
>
> 如需詳細資訊，請參閱檔案設定 cli_binary_format 或 --cli-binary-format 命令列選項。

串流 Blob

aws cloudsearchdomain upload-documents 等串流 blob 不使用字首。而是使用直接檔案路徑來格式化串流 Blob 參數。下列範例將直接檔案路徑 document-batch.json 用於 aws cloudsearchdomain upload-documents 命令：

```
$ aws cloudsearchdomain upload-documents \
    --endpoint-url https://doc-my-domain.us-west-1.cloudsearch.amazonaws.com \
    --content-type application/json \
```

```
    --documents document-batch.json
```

Map

以 JSON 或使用 CLI 速記語法指定的鍵/值對組。以下 JSON 範例從名為 my-table 的 Amazon DynamoDB 表格中讀取一個項目，並附帶地圖參數 `--key`。該參數指定一個名為 id 的主金鑰，其在巢狀的 JSON 結構中具有數值 1。

若要在命令列中使用更進階的 JSON，請考慮使用 `jq` 之類的命令列 JSON 處理器來建立 JSON 字串。如需 `jq` 的詳細資訊，請參閱 GitHub 上的 jq 儲存庫。

```
$ aws dynamodb get-item --table-name my-table --key '{"id": {"N":"1"}}'

{
    "Item": {
        "name": {
            "S": "John"
        },
        "id": {
            "N": "1"
        }
    }
}
```

文件

> **Note**
>
> 速記語法與文件類型不相容。

文件類型用於傳送資料，無需將 JSON 嵌入字串中。文件類型使服務能夠提供任意架構，以便您使用更靈活的資料類型。

這允許在不需要轉義值的情況下傳送 JSON 資料。例如，不使用以下轉義的 JSON 輸入：

```
{"document": "{\"key\":true}"}
```

您可以使用以下文件類型：

```
{"document": {"key": true}}
```

文件類型的有效值

由於文件類型的靈活性，有多種有效的值類型。有效值包括以下項目：

字串

```
--option '"value"'
```

Number

```
--option 123
--option 123.456
```

Boolean

```
--option true
```

Null

```
--option null
```

Array

```
--option '["value1", "value2", "value3"]'
--option '["value", 1, true, null, ["key1", 2.34], {"key2": "value2"}]'
```

物件

```
--option '{"key": "value"}'
--option '{"key1": "value1", "key2": 123, "key3": true, "key4": null, "key5":
 ["value3", "value4"], "key6": {"value5": "value6"}'
```

在 中使用引號和常值搭配字串 AWS CLI

在 AWS CLI中使用單引號和雙引號的方式主要有兩種。

- 在包含空格的字串前後使用引號
- 在字串內使用引號

在包含空格的字串前後使用引號

參數名稱及其值在命令列上以空格分隔。如果字串值包含內嵌空間，則必須使用引號括住整個字串，以防止 AWS CLI 將空間誤判為值和下一個參數名稱之間的分隔符號。您使用的引號類型取決於您執行 AWS CLI 的作業系統。

Linux and macOS

使用單引號 ' '

```
$ aws ec2 create-key-pair --key-name 'my key pair'
```

如需使用引號的詳細資訊，請依據您偏好的 Shell 參閱相關使用者文件。

PowerShell

單引號 (建議使用)

單引號 ' ' 稱為 verbatim 字串。此字串將照您的輸入傳遞至命令，這表示不會傳遞 PowerShell 變數。

```
PS C:\> aws ec2 create-key-pair --key-name 'my key pair'
```

雙引號

雙引號 " " 稱為 expandable 字串。變數可在可擴充的字串中傳遞。

```
PS C:\> aws ec2 create-key-pair --key-name "my key pair"
```

如需使用引號的詳細資訊，請參閱 Microsoft PowerShell 文件中的關於引號規則。

Windows command prompt

使用雙引號 " "。

```
C:\> aws ec2 create-key-pair --key-name "my key pair"
```

您也可以使用等號 = 而非空格來將參數名稱與數值分開。通常只有當參數值的開頭為連字號時，才需要這麼做。

```
$ aws ec2 delete-key-pair --key-name=-mykey
```

在字串內使用引號

字串可能包含引號，且您的 Shell 可能需要逸出引號才能使其正常運作。其中一個常見的參數值類型是 JSON 字串。這很複雜，因為其在 JSON 結構中的每個元素名稱和數值前後都包含空格和雙引號 " "。您在命令列輸入 JSON 格式參數的方式會因您的作業系統而異。

若要在命令列中使用更進階的 JSON，請考慮使用 `jq` 之類的命令列 JSON 處理器來建立 JSON 字串。如需 `jq` 的詳細資訊，請參閱 GitHub 上的 jq 儲存庫。

Linux and macOS

為了讓 Linux 和 macOS 按字面含義解釋字串，請使用單引號 ' ' 將 JSON 資料結構括住，如以下範例所示。您不需要逸出 JSON 字串中內嵌的雙引號，因為它們會以字面含義處理。由於 JSON 會以單引號括住，因此字串中的任何單引號都需要逸出；為此，我們通常會在單引號前面使用反斜線 \'。

```
$ aws ec2 run-instances \
    --image-id ami-12345678 \
    --block-device-mappings '[{"DeviceName":"/dev/sdb","Ebs":
{"VolumeSize":20,"DeleteOnTermination":false,"VolumeType":"standard"}}]'
```

如需使用引號的詳細資訊，請依據您偏好的 Shell 參閱相關使用者文件。

PowerShell

使用單引號 ' ' 或雙引號 " "。

單引號 (建議使用)

單引號 ' ' 稱為 `verbatim` 字串。此字串將照您的輸入傳遞至命令，這表示不會傳遞 PowerShell 變數。

由於 JSON 資料結構包含雙引號，我們建議使用單引號 ' ' 將其括住。如果您使用單引號，則不需要逸出內嵌在 JSON 字串中的雙引號。但在 JSON 結構中，您必須使用反引號 ` 來逸出每個單引號。

```
PS C:\> aws ec2 run-instances `
    --image-id ami-12345678 `
    --block-device-mappings '[{"DeviceName":"/dev/sdb","Ebs":
{"VolumeSize":20,"DeleteOnTermination":false,"VolumeType":"standard"}}]'
```

雙引號

雙引號 " " 稱為 expandable 字串。變數可在可擴充的字串中傳遞。

如果您使用雙引號，則不需要逸出內嵌在 JSON 字串中的單引號。但在 JSON 結構中，您必須使用反引號 ` 來逸出每個雙引號，如以下範例所示。

```
PS C:\> aws ec2 run-instances `
    --image-id ami-12345678 `
    --block-device-mappings "[{`"DeviceName`":`"/dev/sdb`",`"Ebs`":
{`"VolumeSize`":20,`"DeleteOnTermination`":false,`"VolumeType`":`"standard`"}}]"
```

如需使用引號的詳細資訊，請參閱 Microsoft PowerShell 文件中的關於引號規則。

> **Warning**
>
> 在 PowerShell 傳送命令至 之前 AWS CLI，它會判斷您的命令是使用一般 PowerShell CommandLineToArgvW 還是引號規則來解譯。在 PowerShell 使用 CommandLineToArgvW 進行處理時，您必須使用反斜線 \ 逸出字元。
> 如需 PowerShell 中 CommandLineToArgvW 的詳細資訊，請參閱 Microsoft DevBlogs 中的 CommandLineToArgvW 對引號和反斜線的奇怪處理是怎麼回事 (What's up with the strange treatment of quotation marks and backslashes by CommandLineToArgvW)、Microsoft Docs Blog 中的大家都以錯誤方式引用命令列引數 (Everyone quotes command line arguments the wrong way)，以及 Microsoft Docs 中的 CommandLineToArgvW 函數。
> 單引號
> 單引號 ' ' 稱為 verbatim 字串。此字串將照您的輸入傳遞至命令，這表示不會傳遞 PowerShell 變數。使用反斜線 \ 逸出字元。
>
> ```
> PS C:\> aws ec2 run-instances `
>     --image-id ami-12345678 `
>     --block-device-mappings '[{\"DeviceName\":\"/dev/sdb\",\"Ebs\":
> {\"VolumeSize\":20,\"DeleteOnTermination\":false,\"VolumeType\":\"standard\"}}]'
> ```

雙引號

雙引號 " " 稱為 expandable 字串。變數可在 expandable 字串中傳遞。對於雙引號字串，您必須在每個引號使用 `\ 逸出兩次，而不是只用反引號。反引號會逸出反斜線，然後使用反斜線做為 CommandLineToArgvW 程序的逸出字元。

```
PS C:\> aws ec2 run-instances `
    --image-id ami-12345678 `
    --block-device-mappings "[{`\"DeviceName`\":`\"/dev/sdb`\",`\"Ebs`\":
{`\"VolumeSize`\":20,`\"DeleteOnTermination`\":false,`\"VolumeType`\":`
\"standard`\"}}]"
```

Blob (建議)

若要繞過 JSON 資料輸入的 PowerShell 引號規則，請使用 Blob 將 JSON 資料直接傳遞至 AWS CLI。如需 Blobs 的詳細資訊，請參閱 Blob。

Windows command prompt

Windows 命令提示要求使用雙引號 " " 括住 JSON 資料結構。此外，若要防止命令處理器錯誤解譯 JSON 中內嵌的雙引號，您也必須在 JSON 資料結構本身中逸出 (前面加上反斜線 \ 字元) 每個雙引號 "，如下列範例所示。

```
C:\> aws ec2 run-instances ^
    --image-id ami-12345678 ^
    --block-device-mappings "[{\"DeviceName\":\"/dev/sdb\",\"Ebs\":
{\"VolumeSize\":20,\"DeleteOnTermination\":false,\"VolumeType\":\"standard\"}}]"
```

只有最外面的雙引號不用逸出。

從 中的檔案載入參數 AWS CLI

有些參數預期檔案名稱為引數， 會從中 AWS CLI 載入資料。其他參數可讓您以在命令列中輸入文字或從檔案讀取的形式指定參數值。無論檔案是必要或選用，您都必須正確編碼檔案， AWS CLI 才能了解該檔案。檔案的編碼必須符合讀取系統的預設地區設定。您可以透過使用 Python `locale.getpreferredencoding()` 方法判斷。

此方法用於載入單一參數的檔案。如需使用單一檔案載入多個參數的資訊，請參閱 the section called "產生 CLI 骨架範本"。

> **Note**
>
> 根據預設，Windows PowerShell 會將文字輸出為 UTF-16，這會與 JSON 檔案和許多 Linux 系統所使用的 UTF-8 編碼衝突。我們建議您`-Encoding ascii`搭配 PowerShell `Out-File`命令使用 ，以確保 AWS CLI 可以讀取產生的檔案。

主題

- 如何從檔案載入參數
- 二進位檔案
- 將檔案載入為速記語法值

如何從檔案載入參數

從檔案載入參數值有時很方便，而不要嘗試完全以命令列參數值輸入，例如當參數是複雜的 JSON 字串時。若要指定包含數值的檔案，請以下列格式指定檔案 URL。

```
file://complete/path/to/file
```

- 前兩個斜線「/」字元是規格的一部分。如果所需的路徑以「/」開頭，則結果為三個斜線字元：`file:///folder/file`。
- 此 URL 提供檔案的路徑，該檔案包含實際的參數內容。
- 使用具有空格或特殊字元的檔案時，請遵循終端機的引號和逸出規則。

在下列範例中，檔案路徑是以相對於目前的工作目錄來解譯。

Linux or macOS

```
// Read from a file in the current directory
$ aws ec2 describe-instances --filters file://filter.json

// Read from a file in /tmp
$ aws ec2 describe-instances --filters file:///tmp/filter.json

// Read from a file with a filename with whitespaces
$ aws ec2 describe-instances --filters 'file://filter content.json'
```

Windows command prompt

```
// Read from a file in C:\temp
C:\> aws ec2 describe-instances --filters file://C:\temp\filter.json

// Read from a file with a filename with whitespaces
C:\> aws ec2 describe-instances --filters "file://C:\temp\filter content.json"
```

`file://` 前綴選項支援 Unix 風格的擴展，包括 "~/"、"./" 和 "../"。在 Windows 上，"~/" 表達式擴展到您的使用者目錄，儲存在 %USERPROFILE% 環境變數中。例如，在 Windows 10 中，通常在 %USERPROFILE% 下有一個使用者目錄。

您仍然必須逸出內嵌為其他 JSON 文件數值的 JSON 文件。

```
$ aws sqs create-queue --queue-name my-queue --attributes file://attributes.json
```

attributes.json

```
{
  "RedrivePolicy": "{\"deadLetterTargetArn\":\"arn:aws:sqs:us-
west-2:0123456789012:deadletter\", \"maxReceiveCount\":\"5\"}"
}
```

二進位檔案

對於以二進位資料作為參數的命令，使用 `fileb://` 前綴所指定的資料是二進位內容。接受二進位資料的命令包括：

- **`aws ec2 run-instances`:** `--user-data` 參數。
- **`aws s3api put-object`:** `--sse-customer-key` 參數。
- **`aws kms decrypt`:** `--ciphertext-blob` 參數。

以下範例使用 Linux 命令列工具產生二進位 256 位元 AES 金鑰，然後將其提供給 Amazon S3 來加密上傳的檔案伺服器端。

```
$ dd if=/dev/urandom bs=1 count=32 > sse.key
32+0 records in
32+0 records out
32 bytes (32 B) copied, 0.000164441 s, 195 kB/s
```

```
$ aws s3api put-object \
    --bucket amzn-s3-demo-bucket \
    --key test.txt \
    --body test.txt \
    --sse-customer-key fileb://sse.key \
    --sse-customer-algorithm AES256
{
    "SSECustomerKeyMD5": "iVg8oWa8sy714+FjtesrJg==",
    "SSECustomerAlgorithm": "AES256",
    "ETag": "\"a6118e84b76cf98bf04bbe14b6045c6c\""
}
```

關於參考包含 JSON 格式參數的檔案的另一個範例，請參閱 將 IAM 受管政策連接至使用者。

將檔案載入為速記語法值

使用值較大或複雜的速記語法時，載入 檔案做為值通常比較容易。若要將檔案載入為速記語法值，格式會稍微變更。您將`key=value`使用 `@=` 運算子而非 `=`運算子。`@=` 表示 AWS CLI 值應該讀取為檔案路徑，而不是字串。下列範例顯示金鑰值對載入檔案的值。

Linux or macOS

```
--option key@=file://template.txt
```

Windows

```
--option "key1@=file://template.txt"
```

下列範例示範為 `aws rolesanywhere create-trust-anchor`命令載入憑證檔案。

```
$ aws rolesanywhere create-trust-anchor --name TrustAnchor \
    --source sourceData={x509CertificateData@=file://root-
ca.crt},sourceType="CERTIFICATE_BUNDLE"  \
    --enabled
```

如需速記語法的詳細資訊，請參閱 the section called “速記語法”。

AWS CLI 中的骨架和輸入檔案 AWS CLI

大多數 AWS CLI 命令都接受從檔案匯入參數輸入。這些範本可以使用 `generate-cli-skeleton`選項產生，然後使用 `--cli-input-json`和 `--cli-input-yaml` 參數匯入。

主題

- 關於骨 AWS CLI 架和輸入檔案
- 產生和匯入命令骨架
- 結合輸入檔案和命令列參數

關於骨 AWS CLI 架和輸入檔案

大多數 AWS Command Line Interface (AWS CLI) 命令都支援使用 `--cli-input-json`和 `--cli-input-yaml` 參數接受來自檔案的參數輸入。

這些相同的命令使用 `--generate-cli-skeleton` 參數來產生 JSON 或 YAML 格式的檔案，其中包含您可以編輯和填入的所有參數。然後，您可以使用 `--cli-input-json`或 `--cli-input-yaml` 參數執行 命令，並指向填充檔案。

> **Important**
>
> 自訂 AWS CLI 命令，例如`aws s3`命令不支援本主題中所述的 `--generate-cli-skeleton`或 `--cli-input-json`和 `--cli-input-yaml` 參數。若要檢查特定命令是否支援這些參數，請針對您要使用的命令執行 `help` 命令，或參閱 第 第 AWS CLI 2 版參考指南。

`--generate-cli-skeleton` 會產生並顯示參數範本，您可以在命令上自訂和用作輸入。產生的範本包含命令支援的所有參數。

`--generate-cli-skeleton` 參數接受下列其中一個數值：

- `input` – 產生的範本包括所有格式化為 JSON 的輸入參數。這是預設值。
- `yaml-input` – 產生的範本包括所有格式化為 YAML 的輸入參數。
- `output` – 產生的範本包括所有 JSON 格式的輸出參數。您目前無法請求輸出參數為 YAML。

由於 基本上 AWS CLI 是圍繞服務 API 的「包裝函式」，因此骨架檔案會要求您依其基礎 API 參數名稱來參考所有參數。這可能與 AWS CLI 參數名稱不同。例如，名為 的 AWS CLI 參數`user-name`可能會對應到名為 AWS 的服務 API 參數 `UserName`（請注意已變更的大寫和遺失破折號）。建議您使用 `--generate-cli-skeleton` 選項，以「正確」參數名稱產生範本，以免發生錯誤。您可以參考 服務的 API 參考指南，以查看預期的參數名稱。您可以從範本刪除任何非必要且不想為其提供數值的參數。

例如，如果您執行以下命令，其會為 Amazon Elastic Compute Cloud (Amazon EC2) 命令 run-instances 產生參數範本。

JSON

下列範例顯示如何使用 `--generate-cli-skeleton` 參數的預設值 (input) 生產格式化的 JSON 範本。

```
$ aws ec2 run-instances --generate-cli-skeleton
```

```
{
    "DryRun": true,
    "ImageId": "",
    "MinCount": 0,
    "MaxCount": 0,
    "KeyName": "",
    "SecurityGroups": [
        ""
    ],
    "SecurityGroupIds": [
        ""
    ],
    "UserData": "",
    "InstanceType": "",
    "Placement": {
        "AvailabilityZone": "",
        "GroupName": "",
        "Tenancy": ""
    },
    "KernelId": "",
    "RamdiskId": "",
    "BlockDeviceMappings": [
        {
            "VirtualName": "",
            "DeviceName": "",
            "Ebs": {
                "SnapshotId": "",
                "VolumeSize": 0,
                "DeleteOnTermination": true,
                "VolumeType": "",
                "Iops": 0,
                "Encrypted": true
            },
```

```
            "NoDevice": ""
        }
    ],
    "Monitoring": {
        "Enabled": true
    },
    "SubnetId": "",
    "DisableApiTermination": true,
    "InstanceInitiatedShutdownBehavior": "",
    "PrivateIpAddress": "",
    "ClientToken": "",
    "AdditionalInfo": "",
    "NetworkInterfaces": [
        {
            "NetworkInterfaceId": "",
            "DeviceIndex": 0,
            "SubnetId": "",
            "Description": "",
            "PrivateIpAddress": "",
            "Groups": [
                ""
            ],
            "DeleteOnTermination": true,
            "PrivateIpAddresses": [
                {
                    "PrivateIpAddress": "",
                    "Primary": true
                }
            ],
            "SecondaryPrivateIpAddressCount": 0,
            "AssociatePublicIpAddress": true
        }
    ],
    "IamInstanceProfile": {
        "Arn": "",
        "Name": ""
    },
    "EbsOptimized": true
}
```

YAML

下列範例顯示如何使用 yaml-input 參數的 --generate-cli-skeleton 數值產生格式化的 YAML 範本。

```
$ aws ec2 run-instances --generate-cli-skeleton yaml-input
```

```
BlockDeviceMappings:  # The block device mapping entries.
- DeviceName: ''  # The device name (for example, /dev/sdh or xvdh).
  VirtualName: '' # The virtual device name (ephemeralN).
  Ebs: # Parameters used to automatically set up Amazon EBS volumes when the
 instance is launched.
    DeleteOnTermination: true  # Indicates whether the EBS volume is deleted on
 instance termination.
    Iops: 0 # The number of I/O operations per second (IOPS) that the volume
 supports.
    SnapshotId: '' # The ID of the snapshot.
    VolumeSize: 0 # The size of the volume, in GiB.
    VolumeType: st1 # The volume type. Valid values are: standard, io1, gp2, sc1,
 st1.
    Encrypted: true # Indicates whether the encryption state of an EBS volume is
 changed while being restored from a backing snapshot.
    KmsKeyId: '' # Identifier (key ID, key alias, ID ARN, or alias ARN) for a
 customer managed KMS key under which the EBS volume is encrypted.
  NoDevice: '' # Suppresses the specified device included in the block device
 mapping of the AMI.
ImageId: '' # The ID of the AMI.
InstanceType: c4.4xlarge # The instance type. Valid values are: t1.micro, t2.nano,
 t2.micro, t2.small, t2.medium, t2.large, t2.xlarge, t2.2xlarge, t3.nano, t3.micro,
 t3.small, t3.medium, t3.large, t3.xlarge, t3.2xlarge, t3a.nano, t3a.micro,
 t3a.small, t3a.medium, t3a.large, t3a.xlarge, t3a.2xlarge, m1.small, m1.medium,
 m1.large, m1.xlarge, m3.medium, m3.large, m3.xlarge, m3.2xlarge, m4.large,
 m4.xlarge, m4.2xlarge, m4.4xlarge, m4.10xlarge, m4.16xlarge, m2.xlarge, m2.2xlarge,
 m2.4xlarge, cr1.8xlarge, r3.large, r3.xlarge, r3.2xlarge, r3.4xlarge, r3.8xlarge,
 r4.large, r4.xlarge, r4.2xlarge, r4.4xlarge, r4.8xlarge, r4.16xlarge, r5.large,
 r5.xlarge, r5.2xlarge, r5.4xlarge, r5.8xlarge, r5.12xlarge, r5.16xlarge,
 r5.24xlarge, r5.metal, r5a.large, r5a.xlarge, r5a.2xlarge, r5a.4xlarge,
 r5a.8xlarge, r5a.12xlarge, r5a.16xlarge, r5a.24xlarge, r5d.large, r5d.xlarge,
 r5d.2xlarge, r5d.4xlarge, r5d.8xlarge, r5d.12xlarge, r5d.16xlarge, r5d.24xlarge,
 r5d.metal, r5ad.large, r5ad.xlarge, r5ad.2xlarge, r5ad.4xlarge, r5ad.8xlarge,
 r5ad.12xlarge, r5ad.16xlarge, r5ad.24xlarge, x1.16xlarge, x1.32xlarge, x1e.xlarge,
 x1e.2xlarge, x1e.4xlarge, x1e.8xlarge, x1e.16xlarge, x1e.32xlarge, i2.xlarge,
 i2.2xlarge, i2.4xlarge, i2.8xlarge, i3.large, i3.xlarge, i3.2xlarge, i3.4xlarge,
 i3.8xlarge, i3.16xlarge, i3.metal, i3en.large, i3en.xlarge, i3en.2xlarge,
 i3en.3xlarge, i3en.6xlarge, i3en.12xlarge, i3en.24xlarge, i3en.metal, hi1.4xlarge,
 hs1.8xlarge, c1.medium, c1.xlarge, c3.large, c3.xlarge, c3.2xlarge, c3.4xlarge,
 c3.8xlarge, c4.large, c4.xlarge, c4.2xlarge, c4.4xlarge, c4.8xlarge, c5.large,
 c5.xlarge, c5.2xlarge, c5.4xlarge, c5.9xlarge, c5.12xlarge, c5.18xlarge,
```

```
 c5.24xlarge, c5.metal, c5d.large, c5d.xlarge, c5d.2xlarge, c5d.4xlarge,
 c5d.9xlarge, c5d.18xlarge, c5n.large, c5n.xlarge, c5n.2xlarge, c5n.4xlarge,
 c5n.9xlarge, c5n.18xlarge, cc1.4xlarge, cc2.8xlarge, g2.2xlarge, g2.8xlarge,
 g3.4xlarge, g3.8xlarge, g3.16xlarge, g3s.xlarge, g4dn.xlarge, g4dn.2xlarge,
 g4dn.4xlarge, g4dn.8xlarge, g4dn.12xlarge, g4dn.16xlarge, cg1.4xlarge, p2.xlarge,
 p2.8xlarge, p2.16xlarge, p3.2xlarge, p3.8xlarge, p3.16xlarge, p3dn.24xlarge,
 d2.xlarge, d2.2xlarge, d2.4xlarge, d2.8xlarge, f1.2xlarge, f1.4xlarge, f1.16xlarge,
 m5.large, m5.xlarge, m5.2xlarge, m5.4xlarge, m5.8xlarge, m5.12xlarge, m5.16xlarge,
 m5.24xlarge, m5.metal, m5a.large, m5a.xlarge, m5a.2xlarge, m5a.4xlarge,
 m5a.8xlarge, m5a.12xlarge, m5a.16xlarge, m5a.24xlarge, m5d.large, m5d.xlarge,
 m5d.2xlarge, m5d.4xlarge, m5d.8xlarge, m5d.12xlarge, m5d.16xlarge, m5d.24xlarge,
 m5d.metal, m5ad.large, m5ad.xlarge, m5ad.2xlarge, m5ad.4xlarge, m5ad.8xlarge,
 m5ad.12xlarge, m5ad.16xlarge, m5ad.24xlarge, h1.2xlarge, h1.4xlarge, h1.8xlarge,
 h1.16xlarge, z1d.large, z1d.xlarge, z1d.2xlarge, z1d.3xlarge, z1d.6xlarge,
 z1d.12xlarge, z1d.metal, u-6tb1.metal, u-9tb1.metal, u-12tb1.metal, u-18tb1.metal,
 u-24tb1.metal, a1.medium, a1.large, a1.xlarge, a1.2xlarge, a1.4xlarge, a1.metal,
 m5dn.large, m5dn.xlarge, m5dn.2xlarge, m5dn.4xlarge, m5dn.8xlarge, m5dn.12xlarge,
 m5dn.16xlarge, m5dn.24xlarge, m5n.large, m5n.xlarge, m5n.2xlarge, m5n.4xlarge,
 m5n.8xlarge, m5n.12xlarge, m5n.16xlarge, m5n.24xlarge, r5dn.large, r5dn.xlarge,
 r5dn.2xlarge, r5dn.4xlarge, r5dn.8xlarge, r5dn.12xlarge, r5dn.16xlarge,
 r5dn.24xlarge, r5n.large, r5n.xlarge, r5n.2xlarge, r5n.4xlarge, r5n.8xlarge,
 r5n.12xlarge, r5n.16xlarge, r5n.24xlarge.
Ipv6AddressCount: 0 # [EC2-VPC] The number of IPv6 addresses to associate with the
 primary network interface.
Ipv6Addresses: # [EC2-VPC] The IPv6 addresses from the range of the subnet to
 associate with the primary network interface.
- Ipv6Address: ''  # The IPv6 address.
KernelId: '' # The ID of the kernel.
KeyName: '' # The name of the key pair.
MaxCount: 0 # [REQUIRED] The maximum number of instances to launch.
MinCount: 0 # [REQUIRED] The minimum number of instances to launch.
Monitoring: # Specifies whether detailed monitoring is enabled for the instance.
  Enabled: true  # [REQUIRED] Indicates whether detailed monitoring is enabled.
Placement: # The placement for the instance.
  AvailabilityZone: ''  # The Availability Zone of the instance.
  Affinity: '' # The affinity setting for the instance on the Dedicated Host.
  GroupName: '' # The name of the placement group the instance is in.
  PartitionNumber: 0 # The number of the partition the instance is in.
  HostId: '' # The ID of the Dedicated Host on which the instance resides.
  Tenancy: dedicated # The tenancy of the instance (if the instance is running in a
 VPC). Valid values are: default, dedicated, host.
  SpreadDomain: '' # Reserved for future use.
RamdiskId: '' # The ID of the RAM disk to select.
SecurityGroupIds: # The IDs of the security groups.
```

```
- ''
SecurityGroups: # [default VPC] The names of the security groups.
- ''
SubnetId: '' # [EC2-VPC] The ID of the subnet to launch the instance into.
UserData: '' # The user data to make available to the instance.
AdditionalInfo: '' # Reserved.
ClientToken: '' # Unique, case-sensitive identifier you provide to ensure the
 idempotency of the request.
DisableApiTermination: true # If you set this parameter to true, you can't terminate
 the instance using the Amazon EC2 console, CLI, or API; otherwise, you can.
DryRun: true # Checks whether you have the required permissions for the action,
 without actually making the request, and provides an error response.
EbsOptimized: true # Indicates whether the instance is optimized for Amazon EBS I/O.
IamInstanceProfile: # The IAM instance profile.
  Arn: ''  # The Amazon Resource Name (ARN) of the instance profile.
  Name: '' # The name of the instance profile.
InstanceInitiatedShutdownBehavior: stop # Indicates whether an instance stops or
 terminates when you initiate shutdown from the instance (using the operating system
 command for system shutdown). Valid values are: stop, terminate.
NetworkInterfaces: # The network interfaces to associate with the instance.
- AssociatePublicIpAddress: true  # Indicates whether to assign a public IPv4
 address to an instance you launch in a VPC.
  DeleteOnTermination: true # If set to true, the interface is deleted when the
 instance is terminated.
  Description: '' # The description of the network interface.
  DeviceIndex: 0 # The position of the network interface in the attachment order.
  Groups: # The IDs of the security groups for the network interface.
  - ''
  Ipv6AddressCount: 0 # A number of IPv6 addresses to assign to the network
 interface.
  Ipv6Addresses: # One or more IPv6 addresses to assign to the network interface.
  - Ipv6Address: ''  # The IPv6 address.
  NetworkInterfaceId: '' # The ID of the network interface.
  PrivateIpAddress: '' # The private IPv4 address of the network interface.
  PrivateIpAddresses: # One or more private IPv4 addresses to assign to the network
 interface.
  - Primary: true  # Indicates whether the private IPv4 address is the primary
 private IPv4 address.
    PrivateIpAddress: '' # The private IPv4 addresses.
  SecondaryPrivateIpAddressCount: 0 # The number of secondary private IPv4
 addresses.
  SubnetId: '' # The ID of the subnet associated with the network interface.
  InterfaceType: '' # The type of network interface.
PrivateIpAddress: '' # [EC2-VPC] The primary IPv4 address.
```

```
ElasticGpuSpecification: # An elastic GPU to associate with the instance.
- Type: ''  # [REQUIRED] The type of Elastic Graphics accelerator.
ElasticInferenceAccelerators: # An elastic inference accelerator to associate with
 the instance.
- Type: ''  # [REQUIRED]  The type of elastic inference accelerator.
TagSpecifications: # The tags to apply to the resources during launch.
- ResourceType: network-interface  # The type of resource to tag. Valid values
 are: client-vpn-endpoint, customer-gateway, dedicated-host, dhcp-options, elastic-
ip, fleet, fpga-image, host-reservation, image, instance, internet-gateway,
 launch-template, natgateway, network-acl, network-interface, reserved-instances,
 route-table, security-group, snapshot, spot-instances-request, subnet, traffic-
mirror-filter, traffic-mirror-session, traffic-mirror-target, transit-gateway,
 transit-gateway-attachment, transit-gateway-route-table, volume, vpc, vpc-peering-
connection, vpn-connection, vpn-gateway.
  Tags: # The tags to apply to the resource.
  - Key: ''  # The key of the tag.
    Value: '' # The value of the tag.
LaunchTemplate: # The launch template to use to launch the instances.
  LaunchTemplateId: ''  # The ID of the launch template.
  LaunchTemplateName: '' # The name of the launch template.
  Version: '' # The version number of the launch template.
InstanceMarketOptions: # The market (purchasing) option for the instances.
  MarketType: spot  # The market type. Valid values are: spot.
  SpotOptions: # The options for Spot Instances.
    MaxPrice: ''  # The maximum hourly price you're willing to pay for the Spot
 Instances.
    SpotInstanceType: one-time # The Spot Instance request type. Valid values are:
 one-time, persistent.
    BlockDurationMinutes: 0 # The required duration for the Spot Instances (also
 known as Spot blocks), in minutes.
    ValidUntil: 1970-01-01 00:00:00 # The end date of the request.
    InstanceInterruptionBehavior: terminate # The behavior when a Spot Instance is
 interrupted. Valid values are: hibernate, stop, terminate.
CreditSpecification: # The credit option for CPU usage of the T2 or T3 instance.
  CpuCredits: ''  # [REQUIRED] The credit option for CPU usage of a T2 or T3
 instance.
CpuOptions: # The CPU options for the instance.
  CoreCount: 0  # The number of CPU cores for the instance.
  ThreadsPerCore: 0 # The number of threads per CPU core.
CapacityReservationSpecification: # Information about the Capacity Reservation
 targeting option.
  CapacityReservationPreference: none  # Indicates the instance's Capacity
 Reservation preferences. Valid values are: open, none.
  CapacityReservationTarget: # Information about the target Capacity Reservation.
```

```
    CapacityReservationId: ''  # The ID of the Capacity Reservation.
HibernationOptions: # Indicates whether an instance is enabled for hibernation.
  Configured: true  # If you set this parameter to true, your instance is enabled
 for hibernation.
LicenseSpecifications: # The license configurations.
- LicenseConfigurationArn: ''  # The Amazon Resource Name (ARN) of the license
 configuration.
```

產生和匯入命令骨架

若要產生和使用參數骨架檔案

1. 執行命令搭配 `--generate-cli-skeleton` 參數，產生 JSON 或 YAML，並將輸出導向至檔案以便儲存。

   JSON

   ```
   $ aws ec2 run-instances --generate-cli-skeleton input > ec2runinst.json
   ```

   YAML

   ```
   $ aws ec2 run-instances --generate-cli-skeleton yaml-input > ec2runinst.yaml
   ```

2. 在文字編輯器中開啟參數骨架檔案，移除您不需要的任何參數。例如，您可以將範本剔除為下列內容。移除不需要的元素後，請確認檔案仍然是有效的 JSON 或 YAML。

   JSON

   ```
   {
       "DryRun": true,
       "ImageId": "",
       "KeyName": "",
       "SecurityGroups": [
           ""
       ],
       "InstanceType": "",
       "Monitoring": {
           "Enabled": true
       }
   }
   ```

YAML

```
DryRun: true
ImageId: ''
KeyName: ''
SecurityGroups:
- ''
InstanceType:
Monitoring:
  Enabled: true
```

在此範例中，我們會保留設定為 `true` 的 `DryRun` 參數，以便使用 Amazon EC2 試轉功能。此功能可讓您安全地測試命令，而不必實際建立或修改任何資源。

3. 使用適合您方案的數值，填入其餘數值。在這個範例中，我們提供 Amazon Machine Image (AMI) 的執行個體類型、金鑰名稱、安全群組和識別符以供使用。此範例假設預設值 AWS 區域。AMI `ami-dfc39aef` 是託管於 `us-west-2` 區域中的 64 位元 Amazon Linux 映像。如果使用不同的區域，您必須 找到要使用的正確 AMI ID。

JSON

```
{
    "DryRun": true,
    "ImageId": "ami-dfc39aef",
    "KeyName": "mykey",
    "SecurityGroups": [
        "my-sg"
    ],
    "InstanceType": "t2.micro",
    "Monitoring": {
        "Enabled": true
    }
}
```

YAML

```
DryRun: true
ImageId: 'ami-dfc39aef'
KeyName: 'mykey'
SecurityGroups:
```

```
- 'my-sg'
InstanceType: 't2.micro'
Monitoring:
  Enabled: true
```

4. 使用 `file://` 前綴將完整的範本檔案傳遞到 `--cli-input-json` 或 `--cli-input-yaml`參數，以使用完成的參數執行命令。會 AWS CLI 解譯相對於您目前工作目錄的路徑。下列範例會在目前的工作目錄中 AWS CLI 尋找 檔案。

   JSON

   ```
   $ aws ec2 run-instances --cli-input-json file://ec2runinst.json
   ```

   ```
   A client error (DryRunOperation) occurred when calling the RunInstances
    operation: Request would have succeeded, but DryRun flag is set.
   ```

   YAML

   ```
   $ aws ec2 run-instances --cli-input-yaml file://ec2runinst.yaml
   ```

   ```
   A client error (DryRunOperation) occurred when calling the RunInstances
    operation: Request would have succeeded, but DryRun flag is set.
   ```

   試轉錯誤表示 JSON 或 YAML 正確產生且參數值有效。如果在輸出中報告其他問題，請修復它們並重複先前步驟，直到顯示「`Request would have succeeded`」訊息為止。

5. 現在您可以將 `DryRun` 參數設定為 `false`，以停用試轉。

   JSON

   ```
   {
       "DryRun": false,
       "ImageId": "ami-dfc39aef",
       "KeyName": "mykey",
       "SecurityGroups": [
           "my-sg"
       ],
       "InstanceType": "t2.micro",
       "Monitoring": {
           "Enabled": true
   ```

```
    }
}
```

YAML

```
DryRun: false
ImageId: 'ami-dfc39aef'
KeyName: 'mykey'
SecurityGroups:
- 'my-sg'
InstanceType: 't2.micro'
Monitoring:
  Enabled: true
```

6. 執行命令，run-instances 會實際啟動 Amazon EC2 執行個體並顯示成功啟動產生的詳細資訊。輸出的格式是由 --output 參數控制，與輸入參數範本的格式分開。

JSON

```
$ aws ec2 run-instances --cli-input-json file://ec2runinst.json --output json
```

```
{
    "OwnerId": "123456789012",
    "ReservationId": "r-d94a2b1",
    "Groups": [],
    "Instances": [
...
```

YAML

```
$ aws ec2 run-instances --cli-input-yaml file://ec2runinst.yaml --output yaml
```

```
OwnerId: '123456789012'
ReservationId: 'r-d94a2b1',
Groups":
- ''
Instances:
...
```

結合輸入檔案和命令列參數

輸入檔案可用於所有參數，也可以與 中指定的參數結合 AWS CLI。您可以使用此功能在輸入檔案中經常重複使用的設定，同時將個別設定保留在命令本身中。

下列`aws ec2 run-instances`範例結合使用 輸入檔案和參數。我們提供執行個體類型、金鑰名稱、安全群組、Amazon Machine Image (AMI) 的識別符，以供使用並擔任預設值 AWS 區域。AMI `ami-dfc39aef` 是託管於 `us-west-2` 區域中的 64 位元 Amazon Linux 映像。如果使用不同的區域，您必須 找到要使用的正確 AMI ID。

JSON

JSON 檔案的內容：

```
{
    "ImageId": "ami-dfc39aef",
    "KeyName": "mykey",
    "SecurityGroups": [
        "my-sg"
    ],
    "InstanceType": "t2.micro",
    "Monitoring": {
        "Enabled": true
    }
}
```

YAML

YAML 檔案的內容：

```
ImageId: 'ami-dfc39aef'
KeyName: 'mykey'
SecurityGroups:
- 'my-sg'
InstanceType: 't2.micro'
Monitoring:
  Enabled: true
```

下列範例使用 輸入檔案搭配 `--dry-run` 參數來執行命令的試轉，以確認您是否具有所需的許可，並已使用有效的值填入檔案。

JSON

```
$ aws ec2 run-instances --cli-input-json file://ec2runinst.json --dry-run
```

```
A client error (DryRunOperation) occurred when calling the RunInstances operation:
 Request would have succeeded, but DryRun flag is set.
```

YAML

```
$ aws ec2 run-instances --cli-input-yaml file://ec2runinst.yaml --dry-run
```

```
A client error (DryRunOperation) occurred when calling the RunInstances operation:
 Request would have succeeded, but DryRun flag is set.
```

以下範例接著會使用相同的輸入檔案，但搭配 `--no-dry-run` 參數完整執行命令。

JSON

```
$ aws ec2 run-instances --cli-input-json file://ec2runinst.json --no-dry-run --
output json
```

```
{
    "OwnerId": "123456789012",
    "ReservationId": "r-d94a2b1",
    "Groups": [],
    "Instances": [
...
```

YAML

```
$ aws ec2 run-instances --cli-input-yaml file://ec2runinst.yaml --no-dry-run --
output yaml
```

```
OwnerId: '123456789012'
ReservationId: 'r-d94a2b1',
Groups":
- ''
Instances:
```

```
...
```

在 中使用速記語法 AWS CLI

AWS Command Line Interface (AWS CLI) 可以接受 JSON 格式的許多選項參數。不過，在命令列輸入大型 JSON 清單或結構可能會變得很冗長。為了讓此操作更輕鬆， AWS CLI 也支援速記語法，相較於使用完整的 JSON 格式，它可讓您更輕鬆地呈現選項參數。

主題

- 具有鍵值對的結構參數
- 將檔案載入為速記語法值
- 搭配 使用速記語法 AWS CLI

具有鍵值對的結構參數

中的速記語法 AWS CLI 可讓使用者更輕鬆地輸入平面參數 （非巢狀結構）。格式為以逗號分隔的鍵值對清單。請務必使用您的終端機適用的引用和轉義規則，因為速記語法是字串。

Linux or macOS

```
--option key1=value1,key2=value2,key3=value3
```

等同於下列範例，格式為 JSON。

```
--option '{"key1":"value1","key2":"value2","key3":"value3"}'
```

Windows

```
--option "key1=value1,key2=value2,key3=value3"
```

等同於下列範例，格式為 JSON。

```
--option '{"key1":"value1","key2":"value2","key3":"value3"}'
```

每個逗號分隔的金鑰/數值對之間不能有空白字元。以下範例為 Amazon DynamoDB `update-table` 命令，其包含以速記語法指定的 `--provisioned-throughput` 選項。

```
$ aws dynamodb update-table \
    --provisioned-throughput ReadCapacityUnits=15,WriteCapacityUnits=10 \
    --table-name MyDDBTable
```

這等同於下列 JSON 格式的範例。

```
$ aws dynamodb update-table \
    --provisioned-throughput '{"ReadCapacityUnits":15,"WriteCapacityUnits":10}' \
    --table-name MyDDBTable
```

將檔案載入為速記語法值

當值較大或複雜時，載入 做為值通常比較容易。若要將檔案載入為速記語法值，格式會稍微變更。您將`key=value`使用 `@=` 運算子而非 `=`運算子。`@=` 表示 AWS CLI 值應該讀取為檔案路徑，而不是字串。以速記語法載入檔案時，AWS CLI 適用一般的檔案格式規則。下列範例顯示金鑰值對載入檔案的值。

Linux or macOS

```
--option key@=file://template.txt
```

Windows

```
--option "key1@=file://template.txt"
```

下列範例示範為 `aws rolesanywhere create-trust-anchor`命令載入憑證檔案。

```
$ aws rolesanywhere create-trust-anchor --name TrustAnchor \
    --source sourceData={x509CertificateData@=file://root-
ca.crt},sourceType="CERTIFICATE_BUNDLE"  \
    --enabled
```

搭配 使用速記語法 AWS CLI

您可以用兩種方式指定清單形式中的輸入參數：JSON 或速記。 AWS CLI 的速記語法易於傳入具有數字、字串或非巢狀結構的清單中。

基本格式如下所示，清單中的數值以單一空格分隔。

```
--option value1 value2 value3
```

這等同於下列 JSON 格式的範例。

```
--option '[value1,value2,value3]'
```

如同先前所述，您可以使用速記語法來指定數字清單、字串清單或非巢狀結構的清單。以下是 Amazon Elastic Compute Cloud (Amazon EC2) `stop-instances` 命令的範例，其中 `--instance-ids` 選項的輸入參數 (字串清單) 以速記語法指定。

```
$ aws ec2 stop-instances \
    --instance-ids i-1486157a i-1286157c i-ec3a7e87
```

這等同於下列 JSON 格式的範例。

```
$ aws ec2 stop-instances \
    --instance-ids '["i-1486157a","i-1286157c","i-ec3a7e87"]'
```

下列範例顯示 Amazon EC2 `create-tags` 命令，其中 `--tags` 選項採用非巢狀結構的清單。`--resources` 選項指定要標記之執行個體的 ID。

```
$ aws ec2 create-tags \
    --resources i-1286157c \
    --tags Key=My1stTag,Value=Value1 Key=My2ndTag,Value=Value2
 Key=My3rdTag,Value=Value3
```

這等同於下列 JSON 格式的範例。為確保易讀性，JSON 參數以多行文字編寫。

```
$ aws ec2 create-tags \
    --resources i-1286157c \
    --tags '[
        {"Key": "My1stTag", "Value": "Value1"},
        {"Key": "My2ndTag", "Value": "Value2"},
        {"Key": "My3rdTag", "Value": "Value3"}
    ]'
```

在 中啟用和使用命令提示 AWS CLI

您可以在執行 命令時，讓 第 2 AWS CLI 版提示您命令、參數和資源aws。

主題

- 運作方式
- 自動提示功能
- 自動提示模式
- 設定自動提示

運作方式

啟用自動提示後，您可使用 ENTER 鍵來完成部分輸入的命令。按下 ENTER 鍵後，系統會根據您繼續輸入的內容來建議命令、參數和資源。建議會在左側列出命令、參數或資源的名稱，並在右側列出相關描述。若要選取並使用建議，請使用方向鍵來反白顯示該列，然後按下 SPACE 鍵。完成輸入命令後，按下 ENTER 來使用命令。以下範例呈現自動提示的建議清單。

```
$ aws
> aws a
       accessanalyzer                Access Analyzer
       acm                           AWS Certificate Manager
       acm-pca                       AWS Certificate Manager Private Certificate
 Authority
       alexaforbusiness              Alexa For Business
       amplify                       AWS Amplify
```

自動提示功能

自動提示包含以下實用功能：

文件面板

提供有關目前命令的說明文件。若要開啟文件，請按 F3 鍵。

命令完成

建議可使用的 aws 命令。若要查看清單，請輸入命令的部分內容。以下範例是搜尋以字母 a 為開頭的服務。

```
$ aws
> aws a
       accessanalyzer                Access Analyzer
       acm                           AWS Certificate Manager
```

```
       acm-pca                          AWS Certificate Manager Private Certificate
 Authority
       alexaforbusiness                 Alexa For Business
       amplify                          AWS Amplify
```

參數完成

您輸入命令後，自動提示功能會開始建議參數。參數的描述包括數值類型，以及有關參數內容的描述。首先會列出必要的參數，再按要求進行標記。以下範例顯示 `aws dynamodb describe-table` 的參數自動提示清單。

```
$ aws dynamodb describe-table
> aws dynamodb describe-table
                               --table-name (required)  [string] The name of the
 table to describe.
                                --cli-input-json         [string] Reads arguments
 from the JSON string provided. The JSON string follows the format provide...
                                --cli-input-yaml         [string] Reads arguments
 from the YAML string provided. The YAML string follows the format provide...
                                --generate-cli-skeleton  [string] Prints a JSON
 skeleton to standard output without sending an API request. If provided wit...
```

資源完成

自動提示會使用可用的 AWS 資源屬性呼叫 AWS API 以建議資源數值。此會允許在輸入參數時，自動提示建議您擁有的潛在資源。在以下範例中，自動提示在您填入 `aws dynamodb describe-table` 命令的 `--table-name` 參數時列出表格名稱。

```
$ aws dynamodb describe-table
> aws dynamodb describe-table --table-name
                                              Table1
                                              Table2
                                              Table3
```

速記完成

若為使用速記語法的參數，自動提示會建議要使用的數值。在下列範例中，自動提示會列出 `aws ec2 run-instances` 命令中 `--placement` 參數的速記語法數值。

```
$ aws ec2 run-instances
> aws ec2 run-instances --placement
```

```
AvailabilityZone=      [string] The Availability Zone of the instance. If not
specified, an Availability Zone wil...
Affinity=              [string] The affinity setting for the instance on the
Dedicated Host. This parameter is no...
GroupName=             [string] The name of the placement group the instance is in.
PartitionNumber=       [integer] The number of the partition the instance is in.
Valid only if the placement grou...
```

檔案完成

在 aws 命令中填入參數時，自動完成會在使用前綴 file:// 或 fileb:// 後建議本機檔案名稱。在以下範例中，在為 aws ec2 run-instances 命令輸入 --item file:// 後，自動提示建議本機檔案。

```
$ aws ec2 run-instances
> aws ec2 run-instances --item file://
                        item1.txt
                        file1.json
                        file2.json
```

區域完成

在使用全域參數 --region 時，自動提示會列出可供選取的區域。下列範例示範為 aws dynamodb list-tables 命令輸入 --region 後，自動提示會建議以字母排序的區域。

```
$ aws dynamodb list-tables
> aws dynamodb list-tables --region
                                    af-south-1
                                    ap-east-1
                                    ap-northeast-1
                                    ap-northeast-2
```

設定檔完成

在使用全域參數 --profile 時，自動提示會列出您的設定檔。下列範例示範為 aws dynamodb list-tables 命令輸入 --profile 後，自動提示會建議您的設定檔。

```
$ aws dynamodb list-tables
> aws dynamodb list-tables --profile
                                    profile1
                                    profile2
                                    profile3
```

模糊搜尋

完成包含特定字元集的完整命令和數值。下列範例示範為 eu 命令輸入 `--region eu` 後，自動提示會建議包含 `aws dynamodb list-tables` 的區域。

```
$ aws dynamodb list-tables
> aws dynamodb list-tables --region west
                                        eu-west-1
                                        eu-west-2
                                        eu-west-3
                                        us-west-1
```

歷史記錄

若要在自動提示模式中檢視並執行先前使用的命令，請按 CTRL + R。歷史記錄會列出先前用過的命令，您可以使用方向鍵進行選取。以下範例顯示自動提示模式中的歷史記錄。

```
$ aws
> aws
        dynamodb list-tables
        s3 ls
```

自動提示模式

第 2 AWS CLI 版的自動提示有 2 種可設定的模式：

- 完整模式：無論您使用 `--cli-auto-prompt` 參數手動呼叫或永久啟用命令，每次嘗試執行 `aws` 命令時都會使用自動提示。這包括在完整的命令或不完整的命令之後按下 ENTER。
- 部分模式：如果命令不完整或因用戶端驗證錯誤而無法執行時，使用自動提示。如果您有預先存在的指令碼、Runbook，或是您只希望系統自動提示您不熟悉的命令 (而非針對每個命令發出提示)，這個模式就特別有用。

設定自動提示

若要設定自動提示，您可以依優先順序使用下列方法：

- 命令列選項可用於啟用或停用單一命令的自動提示。使用 `--cli-auto-prompt` 可呼叫自動提示，使用 `--no-cli-auto-prompt` 則能停用自動提示。
- 環境變數使用 `aws_cli_auto_prompt` 變數。

- 共享組態檔使用 `cli_auto_prompt` 設定。

在 中控制命令輸出 AWS CLI

本節介紹控制 AWS Command Line Interface (AWS CLI) 輸出的不同方法。自訂終端機中的 AWS CLI 輸出可以改善可讀性、簡化指令碼編寫自動化，以及更輕鬆地導覽較大的資料集。

AWS CLI 支援多種輸出格式，包括 `json`、、 `text`和 `table`。 `yaml`有些服務的資料具有伺服器端分頁，而 AWS CLI 提供其自己的用戶端功能，以用於其他分頁選項。

最後， AWS CLI 具有伺服器端和用戶端篩選，您可以個別或一起使用來篩選 AWS CLI 輸出。

主題

- 敏感輸出
- 伺服器端與用戶端輸出選項
- 在 中設定輸出格式 AWS CLI
- 在 中使用分頁選項 AWS CLI
- 在 中篩選輸出 AWS CLI

敏感輸出

的某些操作 AWS CLI 可能會傳回可能被視為敏感的資訊，包括來自環境變數的資訊。此資訊的公開可能代表在某些情況下的安全風險；例如，資訊可能包含在持續整合和持續部署 (CI/CD) 日誌中。因此，當您在日誌中包含這類輸出時，請務必檢閱 ，並在不需要時隱藏輸出。

如需保護敏感資料的其他資訊，請參閱 the section called “資料保護”。

請考慮下列最佳實務：

- 請考慮以程式設計方式從秘密存放區擷取秘密，例如 AWS Secrets Manager。
- 檢閱建置日誌的內容，以確保其中不包含敏感資訊。考慮使用管道`/dev/null`或擷取輸出做為 bash 或 PowerShell 變數等方法來抑制命令輸出。

  以下是將輸出重新導向至 但不是錯誤的 bash 範例`/dev/null`：

  ```
  $ aws s3 ls > /dev/null
  ```

  如需隱藏終端機輸出的詳細資訊，請參閱您使用之終端機的使用者文件。

- 考慮存取您的日誌，並根據您的使用案例適當調整存取範圍。

伺服器端與用戶端輸出選項

同時 AWS CLI 具有伺服器端和用戶端篩選，您可以個別或一起使用來篩選 AWS CLI 輸出。系統會先處理伺服器端篩選，並傳回用戶端篩選的輸出。服務 API 支援伺服器端篩選。用戶端使用 `--query` 參數支援 AWS CLI 用戶端篩選。

伺服器端輸出選項是 AWS 服務 API 直接支援的功能。任何篩選或分頁的資料都不會傳送到用戶端，這可以加快 HTTP 回應時間，並改善較大資料集的頻寬。

用戶端輸出選項是由 建立的功能 AWS CLI。所有資料都會傳送到用戶端，然後 AWS CLI 篩選或分頁顯示的內容。用戶端操作不會為較大的資料集節省速度或頻寬。

當伺服器端和用戶端選項一起使用時，會先完成伺服器端操作，然後傳送至用戶端以進行用戶端操作。這使用伺服器端選項的潛在速度和頻寬節省，同時使用其他 AWS CLI 功能來取得所需的輸出。

在 中設定輸出格式 AWS CLI

本主題說明 AWS Command Line Interface () 的不同輸出格式AWS CLI。 AWS CLI 支援以下輸出格式：

- **`json`** – 輸出的格式為 JSON 字串。
- **`yaml`** – 輸出的格式為 YAML 字串。
- **`yaml-stream`** – 輸出採用串流方式且格式為 YAML 字串。串流可加速處理大型資料類型。
- **`text`** – 輸出的格式是多行以 Tab 分隔的字串值。這對於將輸出傳遞給文字處理器 (如 `grep`、`sed` 或 awk) 非常有用。
- **`table`** – 輸出的格式為使用字元 +|- 形成儲存格框線的表格。它通常以「方便人類使用」的格式來呈現資訊，這種格式比其他格式更容易閱讀，但在編寫程式方面較不有用。

如何選擇輸出格式

如同組態主題所說明，您可以用三種方式來指定輸出格式：

- 使用 **`config`** 檔案具名設定檔中的 **`output`** 選項 – 下列範例將預設輸出格式設定為 `text`。

```
[default]
output=text
```

- 使用 **AWS_DEFAULT_OUTPUT** 環境變數 – 以下輸出會將此命令列工作階段中的命令格式設定為 `table`，直到變數變更或工作階段結束為止。使用此環境變數覆寫 `config` 檔案中設定的任何數值。

```
$ export AWS_DEFAULT_OUTPUT="table"
```

- 在命令列使用 **--output** 選項 – 以下範例僅將這個命令的輸出設定為 `json`。在命令列上使用此選項會覆寫任何目前設定的環境變數或 `config` 檔案中的數值。

```
$ aws swf list-domains --registration-status REGISTERED --output json
```

> **Important**
>
> 您指定的輸出類型會影響 `--query` 選項的運作方式：
>
> - 如果您指定 `--output text`，輸出會在套用`--query`篩選條件之前進行分頁，而 會在輸出的每個頁面上 AWS CLI 執行一次查詢。因此，查詢會包含每個頁面上第一個相符的要素，即會導致意外額外輸出的要素。若要額外篩選輸出，您可以使用其他命令列工具，例如 `head` 或 `tail`。
> - 如果您指定 `--output json`、 `--output yaml` 或 `--output yaml-stream`，系統就會將輸出完全處理為單一原生結構後，才套用 `--query` 篩選條件。只會針對整個結構 AWS CLI 執行一次查詢，產生篩選結果，然後輸出。

JSON 輸出格式

JSON 為 AWS CLI的預設輸出格式。大部分程式設計語言可以使用內建功能或開放使用的資料庫，來輕易解碼 JSON 字串。您可以將 JSON 輸出與 --query 選項結合，以強大的方式篩選和格式化 AWS CLI JSON 格式的輸出。

您可能無法使用 `--query` 進行更進階的篩選，您可以考慮 `jq`，JSON 處理器的命令列。您可以前往下列網址下載並取得官方教學課程 http://stedolan.github.io/jq/。

以下為 JSON 輸出的範例。

```
$ aws iam list-users --output json
```

```
{
```

```
    "Users": [
        {
            "Path": "/",
            "UserName": "Admin",
            "UserId": "AIDA1111111111EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/Admin",
            "CreateDate": "2014-10-16T16:03:09+00:00",
            "PasswordLastUsed": "2016-06-03T18:37:29+00:00"
        },
        {
            "Path": "/backup/",
            "UserName": "backup-user",
            "UserId": "AIDA2222222222EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/backup/backup-user",
            "CreateDate": "2019-09-17T19:30:40+00:00"
        },
        {
            "Path": "/",
            "UserName": "cli-user",
            "UserId": "AIDA3333333333EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/cli-user",
            "CreateDate": "2019-09-17T19:11:39+00:00"
        }
    ]
}
```

YAML 輸出格式

YAML 是使用發出或取用 YAML 格式字串的服務和工具以程式設計方式處理輸出的好選擇，例如 AWS CloudFormation 支援 YAML 格式的範本。

您可能無法使用 `--query` 進行更進階的篩選，您可以考慮 yq，YAML 處理器的命令列。您可以在 GitHub 上的 yq 存儲庫中下載 yq。

以下為 YAML 輸出的範例。

```
$ aws iam list-users --output yaml
```

```
Users:
- Arn: arn:aws:iam::123456789012:user/Admin
  CreateDate: '2014-10-16T16:03:09+00:00'
  PasswordLastUsed: '2016-06-03T18:37:29+00:00'
```

```
  Path: /
  UserId: AIDA1111111111EXAMPLE
  UserName: Admin
- Arn: arn:aws:iam::123456789012:user/backup/backup-user
  CreateDate: '2019-09-17T19:30:40+00:00'
  Path: /backup/
  UserId: AIDA2222222222EXAMPLE
  UserName: arq-45EFD6D1-CE56-459B-B39F-F9C1F78FBE19
- Arn: arn:aws:iam::123456789012:user/cli-user
  CreateDate: '2019-09-17T19:30:40+00:00'
  Path: /
  UserId: AIDA3333333333EXAMPLE
  UserName: cli-user
```

YAML 串流輸出格式

`yaml-stream` 格式會利用 YAML 格式，並同時透過將資料串流傳輸給您，提供更能回應/更快的大型資料集檢視功能。您可以先開始檢視和使用 YAML 資料，再下載整體查詢。

您可能無法使用 `--query` 進行更進階的篩選，您可以考慮 `yq`，YAML 處理器的命令列。您可以在 GitHub 上的 yq 存儲庫中下載 `yq`。

以下為 `yaml-stream` 輸出範例。

```
$ aws iam list-users --output yaml-stream
```

```
- IsTruncated: false
  Users:
  - Arn: arn:aws:iam::123456789012:user/Admin
    CreateDate: '2014-10-16T16:03:09+00:00'
    PasswordLastUsed: '2016-06-03T18:37:29+00:00'
    Path: /
    UserId: AIDA1111111111EXAMPLE
    UserName: Admin
  - Arn: arn:aws:iam::123456789012:user/backup/backup-user
    CreateDate: '2019-09-17T19:30:40+00:00'
    Path: /backup/
    UserId: AIDA2222222222EXAMPLE
    UserName: arq-45EFD6D1-CE56-459B-B39F-F9C1F78FBE19
  - Arn: arn:aws:iam::123456789012:user/cli-user
    CreateDate: '2019-09-17T19:30:40+00:00'
    Path: /
```

```
    UserId: AIDA3333333333EXAMPLE
    UserName: cli-user
```

以下是 yaml-stream 輸出搭配使用 --page-size 參數對串流的 YAML 內容進行分頁的範例。

```
$ aws iam list-users --output yaml-stream --page-size 2
```

```
- IsTruncated: true
  Marker: ab1234cdef5ghi67jk8lmo9p/
q012rs3t445uv6789w0x1y2z/345a6b78c9d00/1efgh234ij56klmno78pqrstu90vwxyx
  Users:
  - Arn: arn:aws:iam::123456789012:user/Admin
    CreateDate: '2014-10-16T16:03:09+00:00'
    PasswordLastUsed: '2016-06-03T18:37:29+00:00'
    Path: /
    UserId: AIDA1111111111EXAMPLE
    UserName: Admin
  - Arn: arn:aws:iam::123456789012:user/backup/backup-user
    CreateDate: '2019-09-17T19:30:40+00:00'
    Path: /backup/
    UserId: AIDA2222222222EXAMPLE
    UserName: arq-45EFD6D1-CE56-459B-B39F-F9C1F78FBE19
- IsTruncated: false
  Users:
  - Arn: arn:aws:iam::123456789012:user/cli-user
    CreateDate: '2019-09-17T19:30:40+00:00'
    Path: /
    UserId: AIDA3333333333EXAMPLE
    UserName: cli-user
```

文字輸出格式

text 格式會將 AWS CLI 輸出組織成以標籤分隔的行。這很適合用於傳統 Unix 文字工具 (例如 grep、sed 和 awk)，和 PowerShell 執行的文字處理。

text 輸出格式依據基本結構，如下所示。欄位依據底層 JSON 物件之對應金鑰名稱的字母排序。

```
IDENTIFIER  sorted-column1 sorted-column2
IDENTIFIER2 sorted-column1 sorted-column2
```

以下為 text 輸出範例。每個的標籤都與其他欄位獨立，空的欄位則會有額外的標籤。

```
$ aws iam list-users --output text
```

```
USERS   arn:aws:iam::123456789012:user/Admin                2014-10-16T16:03:09+00:00
 2016-06-03T18:37:29+00:00   /          AIDA1111111111EXAMPLE   Admin
USERS   arn:aws:iam::123456789012:user/backup/backup-user   2019-09-17T19:30:40+00:00
                            /backup/   AIDA2222222222EXAMPLE   backup-user
USERS   arn:aws:iam::123456789012:user/cli-user             2019-09-17T19:11:39+00:00
                            /          AIDA3333333333EXAMPLE   cli-user
```

第四欄位為 PasswordLastUsed 欄位，且最後兩個項目是空的，因為該使用者永遠不會登入 AWS Management Console。

> **Important**
>
> 我們強烈建議指定 *text* 輸出時，一律使用 *--query* 選項以確保一致的行為。
> 這是因為文字格式依字母順序依 AWS 服務傳回的基礎 JSON 物件金鑰名稱排序輸出資料欄，而類似的資源可能沒有相同的金鑰名稱。例如，Linux 型 Amazon EC2 執行個體的 JSON 表示法，可能具有 Windows 型執行個體之 JSON 表示法中不存在的元素，反之亦然。此外，資源在未來更新中可能會新增或移除鍵值元素，並改變欄位順序。--query 在此處擴大 text 輸出的功能，以讓您完全控制輸出格式。
> 在下列範例中，命令指定了要顯示的元素，並使用清單表示法 [key1, key2, ...] 來定義欄位順序。如此可讓您充分信賴預期的欄位永遠都會顯示正確的鍵值。最後，請注意 AWS CLI 輸出如何None作為不存在金鑰的值。
>
> ```
> $ aws iam list-users --output text --query 'Users[*].
> [UserName,Arn,CreateDate,PasswordLastUsed,UserId]'
> ```
>
> ```
> Admin         arn:aws:iam::123456789012:user/Admin
>  2014-10-16T16:03:09+00:00   2016-06-03T18:37:29+00:00   AIDA1111111111EXAMPLE
> backup-user   arn:aws:iam::123456789012:user/backup-user
>  2019-09-17T19:30:40+00:00   None                        AIDA2222222222EXAMPLE
> cli-user      arn:aws:iam::123456789012:user/cli-backup
>  2019-09-17T19:11:39+00:00   None                        AIDA3333333333EXAMPLE
> ```

以下範例顯示如何使用 grep 和 awk 搭配來自 aws ec2 describe-instances 命令的 text 輸出。第一個命令顯示可用區域、目前狀態，以及 text 輸出中每個執行個體的執行個體 ID。第二個命令處理該輸出以僅顯示 us-west-2a 可用區域內所有執行中執行個體的執行個體 ID。

```
$ aws ec2 describe-instances --query 'Reservations[*].Instances[*].
[Placement.AvailabilityZone, State.Name, InstanceId]' --output text
```

```
us-west-2a      running i-4b41a37c
us-west-2a      stopped i-a071c394
us-west-2b      stopped i-97a217a0
us-west-2a      running i-3045b007
us-west-2a      running i-6fc67758
```

```
$ aws ec2 describe-instances --query 'Reservations[*].Instances[*].
[Placement.AvailabilityZone, State.Name, InstanceId]' --output text | grep us-west-2a |
 grep running | awk '{print $3}'
```

```
i-4b41a37c
i-3045b007
i-6fc67758
```

以下範例更進一步，不僅顯示如何篩選輸出，也顯示如何使用該輸出來自動變更每個已停止執行個體的執行個體類型。

```
$ aws ec2 describe-instances --query 'Reservations[*].Instances[*].[State.Name,
 InstanceId]' --output text |
> grep stopped |
> awk '{print $2}' |
> while read line;
> do aws ec2 modify-instance-attribute --instance-id $line --instance-type '{"Value":
 "m1.medium"}';
> done
```

text 輸出在 PowerShell 也相當實用。由於 text 輸出中的欄以 Tab 鍵分隔，因此您可以使用 PowerShell 的 `t 分隔符號輕鬆將輸出分割為陣列。下列命令可在第一欄位 (InstanceId) 符合字串 AvailabilityZone 的情況下，顯示第三欄位中的數值 (us-west-2a)。

```
PS C:\>aws ec2 describe-instances --query 'Reservations[*].Instances[*].
[Placement.AvailabilityZone, State.Name, InstanceId]' --output text |
%{if ($_.split("`t")[0] -match "us-west-2a") { $_.split("`t")[2]; } }
```

```
-4b41a37c
i-a071c394
```

```
i-3045b007
i-6fc67758
```

請注意，雖然先前的範例已示範如何使用 `--query` 參數來剖析基礎 JSON 物件並提取所需的欄，但 PowerShell 有自己處理 JSON 的能力，當您無須擔心跨平台相容性時，即可使用。PowerShell 可讓您使用 `ConvertFrom-JSON` cmdlet 產生階層結構化物件，而非以文字形式處理輸出。然後，您可以直接從該物件存取您想要的成員。

```
(aws ec2 describe-instances --output json | ConvertFrom-
Json).Reservations.Instances.InstanceId
```

Tip

如果您輸出文字並使用 `--query` 參數將輸出篩選為單一欄位，則輸出會是單行標籤分隔值。若要將每個數值放在單獨的行上，您可以用括弧括住輸出欄位，如下列範例所示。

Tab 分隔的單行輸出：

```
$ aws iam list-groups-for-user --user-name susan  --output text --query
 "Groups[].GroupName"
```

```
HRDepartment    Developers      SpreadsheetUsers  LocalAdmins
```

將 `[GroupName]` 放在括弧中，每個數值都在單獨一行上：

```
$ aws iam list-groups-for-user --user-name susan  --output text --query
 "Groups[].[GroupName]"
```

```
HRDepartment
Developers
SpreadsheetUsers
LocalAdmins
```

表格輸出格式

`table` 格式能對複雜的 AWS CLI 輸出以表格形式產生易讀形式。

```
$ aws iam list-users --output table
```

```
-----------------------------------------------------------------------------------------------
|
 ListUsers                                                                    |
+-----------------------------------------------------------------------------------------------
+
||
 Users                                                                    ||
|+-----------------------------------------------------+---------------------------
+---------------------------+----------+----------------------+-------------+|
||                        Arn                        |       CreateDate          |
 PasswordLastUsed       |   Path   |        UserId         |   UserName  ||
|+-----------------------------------------------------+---------------------------
+---------------------------+----------+----------------------+-------------+|
||  arn:aws:iam::123456789012:user/Admin              | 2014-10-16T16:03:09+00:00 |
 2016-06-03T18:37:29+00:00 | /        | AIDA1111111111EXAMPLE | Admin       ||
||  arn:aws:iam::123456789012:user/backup/backup-user | 2019-09-17T19:30:40+00:00 |
                      | /backup/ | AIDA2222222222EXAMPLE | backup-user ||
||  arn:aws:iam::123456789012:user/cli-user           | 2019-09-17T19:11:39+00:00 |
                      | /        | AIDA3333333333EXAMPLE | cli-user    ||
+-----------------------------------------------------------------------------------------------
+
```

您可將 `--query` 選項與 `table` 格式結合使用，以顯示從列輸出中預選的元素組。請留意字典表示法和清單表示法之間的輸出差異：在第一個範例中，欄位名稱以字母順序排序；在第二個範例中，未命名欄位則根據使用者定義來排序。如需有關 `--query` 選項的詳細資訊，請參閱 在 中篩選輸出 AWS CLI。

```
$ aws ec2 describe-volumes --query 'Volumes[*].
{ID:VolumeId,InstanceId:Attachments[0].InstanceId,AZ:AvailabilityZone,Size:Size}' --
output table
```

```
-----------------------------------------------------
|                  DescribeVolumes                  |
+------------+----------------+--------------+-------+
|     AZ     |      ID        | InstanceId   | Size  |
+------------+----------------+--------------+-------+
|  us-west-2a|  vol-e11a5288  |  i-a071c394  |  30   |
|  us-west-2a|  vol-2e410a47  |  i-4b41a37c  |  8    |
+------------+----------------+--------------+-------+
```

```
$ aws ec2 describe-volumes --query 'Volumes[*].
[VolumeId,Attachments[0].InstanceId,AvailabilityZone,Size]' --output table
```

```
-------------------------------------------------
|                DescribeVolumes                |
+--------------+--------------+--------------+-----+
|  vol-e11a5288|  i-a071c394  |  us-west-2a  |  30 |
|  vol-2e410a47|  i-4b41a37c  |  us-west-2a  |  8  |
+--------------+--------------+--------------+-----+
```

在 中使用分頁選項 AWS CLI

本主題說明從 AWS Command Line Interface () 分頁輸出的不同方式AWS CLI。

主要有兩種方法來控制 AWS CLI輸出的分頁。

- 使用伺服器端的分頁參數。
- 使用您的預設輸出用戶端分頁程式。

伺服器端分頁參數先處理，然後所有輸出都傳送到用戶端分頁。

伺服器端分頁

對於傳回大量項目清單的大多數命令， AWS CLI 具有多個伺服器端選項，可在 AWS CLI 呼叫服務的 API 填入清單時控制輸出中包含的項目數量。中的伺服器端分頁由 AWS 服務 API AWS CLI 啟用，因此這些選項只有在服務 API 啟用時才有效。

大多數 AWS CLI 命令的選項包括下列項目：

- 如何使用 --no-paginate 參數
- 如何使用 --page-size 參數
- 如何使用 --max-items 參數
- 如何使用 --starting-token 參數

根據預設， AWS CLI 會使用由個別服務決定的頁面大小，並擷取所有可用的項目。例如，Amazon S3 的預設頁面大小為 1000。例如，如果您在包含 3,500 個物件的 Amazon S3 儲存貯體上執行 `aws s3api list-objects`， AWS CLI 會自動對 Amazon S3 進行 4 次呼叫、在背景為您處理服務特定的分頁邏輯，並在最終輸出中傳回全部 3,500 個物件。

如需特定命令是否具有伺服器端分頁的詳細資訊，請參閱第 AWS CLI 2 版參考指南。

如何使用 --no-paginate 參數

--no-paginate 選項在客户端停用以下分頁標記。使用命令時，根據預設， AWS CLI 會自動進行多次呼叫，以傳回所有可能的結果來建立分頁。每個頁面一個呼叫。停用分頁對於命令結果的第一頁 AWS CLI 只有一次呼叫。

例如，如果您在包含 3，500 個物件的 Amazon S3 儲存貯體aws s3api list-objects上執行 ，則 AWS CLI 只會對 Amazon S3 進行第一次呼叫，並在最終輸出中僅傳回前 1，000 個物件。

```
$ aws s3api list-objects \
    --bucket amzn-s3-demo-bucket \
    --no-paginate
{
    "Contents": [
...
```

如何使用 --page-size 參數

如果您在針對大量資源執行清單命令時發生問題，可能是預設頁面大小太高。這可能會導致對 AWS 服務的呼叫超過允許的時間上限，並產生「逾時」錯誤。您可以使用 --page-size選項來指定從每次對 AWS 服務的呼叫 AWS CLI 請求較少數量的項目。 AWS CLI 仍會擷取完整清單，但在背景執行更多數量的服務 API 呼叫，並在每次呼叫時擷取較少數量的項目。這可讓每次呼叫不逾時的成功機會更高。變更頁面大小不會影響輸出；它只會影響產生輸出所需進行的 API 呼叫數量。

```
$ aws s3api list-objects \
    --bucket amzn-s3-demo-bucket \
    --page-size 100
{
    "Contents": [
...
```

如何使用 --max-items 參數

若要一次在 AWS CLI 輸出中包含較少的項目，請使用 --max-items選項。 AWS CLI 仍會如先前所述使用 服務處理分頁，但一次只會列印您指定的項目數量。

```
$ aws s3api list-objects \
    --bucket amzn-s3-demo-bucket \
```

```
    --max-items 100
{
    "NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ==",
    "Contents": [
...
```

如何使用 --starting-token 參數

如果項目輸出數量 (--max-items) 少於基礎 API 呼叫傳回的項目總數，則輸出會包含一個 NextToken，提供您在後續命令中傳送以擷取下一組項目。以下範例說明如何使用前一範例傳回的 NextToken 值，並讓您擷取第二批的 100 個項目。

> **Note**
>
> 參數 --starting-token 不能為 null 或空白。如果先前的命令未傳回 NextToken 值，表示已沒有更多項目可傳回，您也不需要再次呼叫命令。

```
$ aws s3api list-objects \
    --bucket amzn-s3-demo-bucket \
    --max-items 100 \
    --starting-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ==
{
    "Contents": [
...
```

每次呼叫 時，指定的 AWS 服務可能不會以相同的順序傳回項目。如果您為 --page-size 和 --max-items 指定不同的數值，您可能會取得意外的結果，並發生遺漏或重複的項目。欲避免此情況，請對 --page-size 和 --max-items 使用相同數字，以同步 AWS CLI 和基礎服務的分頁。您亦可擷取整個清單，並於本機執行所需的分頁操作。

用戶端分頁程式

AWS CLI 第 2 版提供使用用戶端分頁程式進行輸出。根據預設，此功能會透過作業系統的預設分頁程式傳回所有輸出。

依據優先順序，您可以使用以下方式指定輸出分頁程式：

- 使用 default 或具名設定檔內 config 檔案中的 cli_pager 設定。
- 使用 AWS_PAGER 環境變數。

- 使用 PAGER 環境變數。

依優先順序，您可以透過下列方式停用所有對外部分頁程式的使用：

- 使用 `--no-cli-pager` 命令列選項停用單一命令使用的分頁程式。
- 將 `cli_pager` 設定或 AWS_PAGER 變數設為空白字串。

用戶端分頁程式主題：

- 如何使用 cli_pager 設定
- 如何設定 AWS_PAGER 環境變數
- 如何使用 --no-cli-pager 選項
- 如何使用分頁程式標記

如何使用 cli_pager 設定

您可以將常用的組態設定和憑證儲存在 AWS CLI維護的檔案中。具名設定檔內的設定優先於 `default` 設定檔內的設定。如需組態設定的詳細資訊，請參閱 中的組態和登入資料檔案設定 AWS CLI。

以下範例設定預設的輸出分頁程式至 `less` 程式。

```
[default]
cli_pager=less
```

下列範例會將預設值設定為停用分頁程式。

```
[default]
cli_pager=
```

如何設定 AWS_PAGER 環境變數

以下範例設定預設的輸出分頁程式至 `less` 程式。如需有關環境變數的詳細資訊，請參閱 設定 的環境變數 AWS CLI。

Linux and macOS

```
$ export AWS_PAGER="less"
```

Windows

```
C:\> setx AWS_PAGER "less"
```

如何使用 --no-cli-pager 選項

若要停用單一命令對呼叫程式的使用，請使用 `--no-cli-pager` 選項。如需命令列選項的詳細資訊，請參閱 中的命令列選項 AWS CLI。

```
$ aws s3api list-objects \
    --bucket amzn-s3-demo-bucket \
    --no-cli-pager
{
    "Contents": [
...
```

如何使用分頁程式標記

您可以指定要自動與分頁程式搭配使用的標記。標記是依您使用的分頁程式而定。以下範例是 `less` 和 `more` 的典型預設內容。

Linux and macOS

如果您不另外指定，分頁器第 2 AWS CLI 版預設使用 `less`。如果您沒有設定LESS環境變數，第 2 AWS CLI 版會使用 FRX旗標。您可以在設定 AWS CLI 分頁器時指定旗標，以結合旗標。

下列範例使用 S 標記。然後，此標誌會與預設的 FRX 標誌合併，以建立最終 FRXS 標記。

```
$ export AWS_PAGER="less -S"
```

如果您不希望有任何 FRX 標記，可將其否定。以下範例會否定 F 標記以建立最終 RX 標記。

```
$ export AWS_PAGER="less -+F"
```

如需 `less` 的詳細資訊，請參閱 manpages.org 上的 less。

Windows

如果您未另外指定，分頁器 AWS CLI 版本 2 預設會使用 `more`，沒有額外的旗標。

下列範例會使用 /c 參數。

```
C:\> setx AWS_PAGER "more /c"
```

如需more旗標的詳細資訊，請參閱 Microsoft 文件more上的 。

在 中篩選輸出 AWS CLI

AWS Command Line Interface (AWS CLI) 具有伺服器端和用戶端篩選，您可以個別或一起使用來篩選 AWS CLI 輸出。系統會先處理伺服器端篩選，並傳回用戶端篩選的輸出。

- 伺服器端篩選受 API 支援，且您通常會搭配 `--filter` 參數執行該篩選。服務僅會傳回符合的結果，即可加速大型資料集的 HTTP 回應時間之結果。
- 用戶端使用 `--query` 參數支援 AWS CLI 用戶端篩選。此參數具有伺服器端篩選可能沒有的功能。

主題

- 伺服器端篩選
- 用戶端篩選
- 結合伺服器端和用戶端篩選
- 其他資源

伺服器端篩選

中的伺服器端篩選由 AWS 服務 API AWS CLI 提供。 AWS 服務僅會傳回符合您篩選條件 HTTP 回應中的記錄，其可加速大型資料集的 HTTP 回應時間。由於伺服器端篩選是由服務 API 定義，參數名稱和函數會因服務而異。部分用於篩選的常見參數名稱有：

- `--filter` 例如 ses 和 ce。
- `--filters` 例如 ec2、autoscaling 和 rds。
- 以單字 `filter` 開頭來命名，例如對 `aws dynamodb scan` 命令而言為 `--filter-expression`。

如需特定命令是否具有伺服器端篩選和篩選規則的資訊，請參閱第 AWS CLI 2 版參考指南。

用戶端篩選

使用 `--query` 參數 AWS CLI 提供內建的 JSON 型用戶端篩選功能。`--query` 參數是功能強大的工具，可供您用來自訂輸出的內容與樣式。`--query` 參數會取得從伺服器傳回的 HTTP 回應，並在顯示結果之前先行篩選。因為整個 HTTP 回應會在篩選之前傳送至用戶端，所以在大型資料集速度方面，用戶端篩選可能會比伺服器端篩選來得慢。

查詢會使用 JMESPath 語法 語法來建立表達式，以便篩選輸出。若要學習 JMESPath 語法，請參閱 JMESPath 網站上的教學課程。

> **Important**
>
> 您指定的輸出類型會影響 `--query` 選項的運作方式：
>
> - 如果您指定 `--output text`，輸出會在套用`--query`篩選條件之前進行分頁，而 會在輸出的每個頁面上 AWS CLI 執行一次查詢。因此，查詢會包含每個頁面上第一個相符的要素，即會導致意外額外輸出的要素。若要額外篩選輸出，您可以使用其他命令列工具，例如 `head` 或 `tail`。
> - 如果您指定 `--output json`、 `--output yaml` 或 `--output yaml-stream`，系統就會將輸出完全處理為單一原生結構後，才套用 `--query` 篩選條件。只會針對整個結構 AWS CLI 執行一次查詢，產生篩選結果，然後輸出。

用戶端篩選主題

- 開始之前
- 識別碼
- 從清單中選取
- 篩選巢狀資料
- 扁平化結果
- 篩選特定值
- 輸送表達式
- 篩選多個識別碼值
- 將標籤新增至識別碼值
- 函數
- 進階 --query 範例

開始之前

> **Note**
>
> 這些篩選條件表達式範例是針對基本類似 Linux 的 Shell 撰寫。使用這些範例時，請務必針對終端機 Shell 使用正確的引號規則。終端機解譯輸入的方式可能會大幅變更傳送至 的內容 AWS CLI。終端機如何讀取單引號'、雙引號 "或反引號 ` 可以變更內容的讀取方式。
> 如需詳細資訊，請參閱the section called “具有字串的引號”。

以下 JSON 輸出會顯示如何產生 `--query` 參數之範例。輸出會描述連接到不同 Amazon EC2 執行個體的三個 Amazon EBS 磁碟區。

範例輸出

```
$ aws ec2 describe-volumes
{
  "Volumes": [
    {
      "AvailabilityZone": "us-west-2a",
      "Attachments": [
        {
          "AttachTime": "2013-09-17T00:55:03.000Z",
          "InstanceId": "i-a071c394",
          "VolumeId": "vol-e11a5288",
          "State": "attached",
          "DeleteOnTermination": true,
          "Device": "/dev/sda1"
        }
      ],
      "VolumeType": "standard",
      "VolumeId": "vol-e11a5288",
      "State": "in-use",
      "SnapshotId": "snap-f23ec1c8",
      "CreateTime": "2013-09-17T00:55:03.000Z",
      "Size": 30
    },
    {
      "AvailabilityZone": "us-west-2a",
      "Attachments": [
        {
          "AttachTime": "2013-09-18T20:26:16.000Z",
```

```
          "InstanceId": "i-4b41a37c",
          "VolumeId": "vol-2e410a47",
          "State": "attached",
          "DeleteOnTermination": true,
          "Device": "/dev/sda1"
        }
      ],
      "VolumeType": "standard",
      "VolumeId": "vol-2e410a47",
      "State": "in-use",
      "SnapshotId": "snap-708e8348",
      "CreateTime": "2013-09-18T20:26:15.000Z",
      "Size": 8
    },
    {
      "AvailabilityZone": "us-west-2a",
      "Attachments": [
        {
          "AttachTime": "2020-11-20T19:54:06.000Z",
          "InstanceId": "i-1jd73kv8",
          "VolumeId": "vol-a1b3c7nd",
          "State": "attached",
          "DeleteOnTermination": true,
          "Device": "/dev/sda1"
        }
      ],
      "VolumeType": "standard",
      "VolumeId": "vol-a1b3c7nd",
      "State": "in-use",
      "SnapshotId": "snap-234087fb",
      "CreateTime": "2020-11-20T19:54:05.000Z",
      "Size": 15
    }
  ]
}
```

識別碼

識別碼是輸出值的標籤。在建立篩選條件時，您可以使用識別碼來縮小查詢結果範圍。在下列輸出範例中，所有識別碼 (例如 Volumes、AvailabilityZone 以及 AttachTime) 皆會反白顯示。

```
$ aws ec2 describe-volumes
{
```

```
  "Volumes": [
    {
      "AvailabilityZone": "us-west-2a",
      "Attachments": [
        {
          "AttachTime": "2013-09-17T00:55:03.000Z",
          "InstanceId": "i-a071c394",
          "VolumeId": "vol-e11a5288",
          "State": "attached",
          "DeleteOnTermination": true,
          "Device": "/dev/sda1"
        }
      ],
      "VolumeType": "standard",
      "VolumeId": "vol-e11a5288",
      "State": "in-use",
      "SnapshotId": "snap-f23ec1c8",
      "CreateTime": "2013-09-17T00:55:03.000Z",
      "Size": 30
    },
    {
      "AvailabilityZone": "us-west-2a",
      "Attachments": [
        {
          "AttachTime": "2013-09-18T20:26:16.000Z",
          "InstanceId": "i-4b41a37c",
          "VolumeId": "vol-2e410a47",
          "State": "attached",
          "DeleteOnTermination": true,
          "Device": "/dev/sda1"
        }
      ],
      "VolumeType": "standard",
      "VolumeId": "vol-2e410a47",
      "State": "in-use",
      "SnapshotId": "snap-708e8348",
      "CreateTime": "2013-09-18T20:26:15.000Z",
      "Size": 8
    },
    {
      "AvailabilityZone": "us-west-2a",
      "Attachments": [
        {
          "AttachTime": "2020-11-20T19:54:06.000Z",
```

```
            "InstanceId": "i-1jd73kv8",
            "VolumeId": "vol-a1b3c7nd",
            "State": "attached",
            "DeleteOnTermination": true,
            "Device": "/dev/sda1"
          }
        ],
        "VolumeType": "standard",
        "VolumeId": "vol-a1b3c7nd",
        "State": "in-use",
        "SnapshotId": "snap-234087fb",
        "CreateTime": "2020-11-20T19:54:05.000Z",
        "Size": 15
      }
    ]
}
```

如需詳細資訊，請參閱 JMESPath 網站上的識別碼。

從清單中選取

輕單或陣列為後接方括號 [的識別碼，例如 the section called “開始之前” 中的 Volumes 和 Attachments。

語法

```
<listName>[ ]
```

若要從陣列中篩選所有輸出，您可以使用萬用表示法。萬用字元表達式為使用 * 符號來傳回要素的表達式。

以下範例會查詢所有 Volumes 內容。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*]'
[
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2013-09-17T00:55:03.000Z",
        "InstanceId": "i-a071c394",
```

```
        "VolumeId": "vol-e11a5288",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-e11a5288",
    "State": "in-use",
    "SnapshotId": "snap-f23ec1c8",
    "CreateTime": "2013-09-17T00:55:03.000Z",
    "Size": 30
  },
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2020-11-20T19:54:06.000Z",
        "InstanceId": "i-1jd73kv8",
        "VolumeId": "vol-a1b3c7nd",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-a1b3c7nd",
    "State": "in-use",
    "SnapshotId": "snap-234087fb",
    "CreateTime": "2020-11-20T19:54:05.000Z",
    "Size": 15
  }
]
```

若要依索引檢視陣列中的特定磁碟區，您可以呼叫陣列索引。例如，`Volumes` 陣列中首個項目的索引為 0，會導致 `Volumes[0]` 查詢。如需有關陣列索引的詳細資訊，請參閱 JMESPath 網站上的索引表達式。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[0]'
{
  "AvailabilityZone": "us-west-2a",
  "Attachments": [
```

```
    {
      "AttachTime": "2013-09-17T00:55:03.000Z",
      "InstanceId": "i-a071c394",
      "VolumeId": "vol-e11a5288",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  "VolumeType": "standard",
  "VolumeId": "vol-e11a5288",
  "State": "in-use",
  "SnapshotId": "snap-f23ec1c8",
  "CreateTime": "2013-09-17T00:55:03.000Z",
  "Size": 30
}
```

若要依索引檢視特定範圍的磁碟區，請搭配 `slice` 使用以下語法，其中 start 為開始陣列索引，stop 為篩選條件停止處理的索引，step 則是略過間隔。

語法

```
<arrayName>[<start>:<stop>:<step>]
```

如果從配量表達式中省略了其中任何一個，則會使用下列預設值：

- Start – 清單中的第一個索引，值為 0。
- Stop – 清單中的最後一個索引。
- Step – 未略過步驟，其數值為 1。

若只要傳回前兩個磁碟區，您可以使用 start 數值 0、stop 數值 2，以及 step 數值 1，如下列範例所示。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[0:2:1]'
[
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2013-09-17T00:55:03.000Z",
```

```
        "InstanceId": "i-a071c394",
        "VolumeId": "vol-e11a5288",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-e11a5288",
    "State": "in-use",
    "SnapshotId": "snap-f23ec1c8",
    "CreateTime": "2013-09-17T00:55:03.000Z",
    "Size": 30
  },
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2013-09-18T20:26:16.000Z",
        "InstanceId": "i-4b41a37c",
        "VolumeId": "vol-2e410a47",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-2e410a47",
    "State": "in-use",
    "SnapshotId": "snap-708e8348",
    "CreateTime": "2013-09-18T20:26:15.000Z",
    "Size": 8
  }
]
```

由於此範例中包含預設值，您可以將配量從 `Volumes[0:2:1]` 縮短至 `Volumes[:2]`。

下列範例會省略預設值，並傳回整個陣列中的每兩個磁碟區。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[::2]'
[
  {
    "AvailabilityZone": "us-west-2a",
```

```
    "Attachments": [
      {
        "AttachTime": "2013-09-17T00:55:03.000Z",
        "InstanceId": "i-a071c394",
        "VolumeId": "vol-e11a5288",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-e11a5288",
    "State": "in-use",
    "SnapshotId": "snap-f23ec1c8",
    "CreateTime": "2013-09-17T00:55:03.000Z",
    "Size": 30
  },
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2020-11-20T19:54:06.000Z",
        "InstanceId": "i-1jd73kv8",
        "VolumeId": "vol-a1b3c7nd",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-a1b3c7nd",
    "State": "in-use",
    "SnapshotId": "snap-234087fb",
    "CreateTime": "2020-11-20T19:54:05.000Z",
    "Size": 15
  }
]
```

Step 也可以使用負數並以與陣列相反的順序來篩選，如下列範例所示。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[::-2]'
[
```

```
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2020-11-20T19:54:06.000Z",
        "InstanceId": "i-1jd73kv8",
        "VolumeId": "vol-a1b3c7nd",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-a1b3c7nd",
    "State": "in-use",
    "SnapshotId": "snap-234087fb",
    "CreateTime": "2020-11-20T19:54:05.000Z",
    "Size": 15
  },
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2013-09-17T00:55:03.000Z",
        "InstanceId": "i-a071c394",
        "VolumeId": "vol-e11a5288",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-e11a5288",
    "State": "in-use",
    "SnapshotId": "snap-f23ec1c8",
    "CreateTime": "2013-09-17T00:55:03.000Z",
    "Size": 30
  }
]
```

如需詳細資訊，請參閱 JMESPath 網站上的配量。

篩選巢狀資料

若要縮小對巢狀值 Volumes[*] 的篩選範圍，您可以附加句點以及您的篩選條件標準來使用子運算式。

語法

```
<expression>.<expression>
```

以下範例顯示所有磁碟區的所以 Attachments 資訊。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments'
[
  [
    {
      "AttachTime": "2013-09-17T00:55:03.000Z",
      "InstanceId": "i-a071c394",
      "VolumeId": "vol-e11a5288",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  [
    {
      "AttachTime": "2013-09-18T20:26:16.000Z",
      "InstanceId": "i-4b41a37c",
      "VolumeId": "vol-2e410a47",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  [
    {
      "AttachTime": "2020-11-20T19:54:06.000Z",
      "InstanceId": "i-1jd73kv8",
      "VolumeId": "vol-a1b3c7nd",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
```

```
  ]
]
```

若要進一步篩選巢狀值，請附加每個巢狀識別碼的表達式。以下範例會列出所有 Volumes 的 State。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[*].State'
[
  [
    "attached"
  ],
  [
    "attached"
  ],
  [
    "attached"
  ]
]
```

扁平化結果

如需詳細資訊，請參閱 JMESPath 網站上的子運算式。

您可以移除導致 Volumes[*].Attachments[].State 查詢的萬用表示法，以展平 Volumes[*].Attachments[*].State 結果。展平化通常有助於改善結果的可讀性。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[].State'
[
  "attached",
  "attached",
  "attached"
]
```

如需詳細資訊，請參閱 JMESPath 網站上的展平化。

篩選特定值

若要篩選清單中的特定值，您可以使用篩選條件表達式，如下列語法所示。

語法

```
? <expression> <comparator> <expression>]
```

表達式比較子包括 ==、!=、<、<=、> 和 >=。下列範例會篩選 AttachedState 中所有 Volumes 的 VolumeIds。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[?State==`attached`].VolumeId'
[
  [
    "vol-e11a5288"
  ],
  [
    "vol-2e410a47"
  ],
  [
    "vol-a1b3c7nd"
  ]
]
```

然後可將其展平化，形成以下範例。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[?State==`attached`].VolumeId[]'
[
  "vol-e11a5288",
  "vol-2e410a47",
  "vol-a1b3c7nd"
]
```

下列範例會篩選所有 Volumes 中規模小於 20 的 VolumeIds。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[?Size < `20`].VolumeId'
[
  "vol-2e410a47",
  "vol-a1b3c7nd"
]
```

如需詳細資訊，請參閱 JMESPath 網站上的篩選條件表達式。

輸送表達式

您可以將篩選結果輸送至新清單，然後透過下列語法利用另一個表達式來篩選結果：

語法

```
<expression> | <expression>]
```

下列範例會採用 Volumes[*].Attachments[].InstanceId 表達式的篩選結果，並輸出陣列中的第一個結果。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[].InstanceId | [0]'
"i-a071c394"
```

此範例會先從以下表達式中建立陣列，以達成上述目標。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[].InstanceId'
"i-a071c394",
  "i-4b41a37c",
  "i-1jd73kv8"
```

接著，傳回該陣列中的第一個要素。

```
"i-a071c394"
```

如需詳細資訊，請參閱 JMESPath 網站上的輸送表達式。

篩選多個識別碼值

若要篩選多個識別碼，您可以透過下列語法來使用複選清單：

語法

```
<listName>[].[<expression>, <expression>]
```

在以下範例中，VolumeId 和 VolumeType 會在 Volumes 清單受到篩選，以產生下列表達式。

```
$ aws ec2 describe-volumes \
```

```
    --query 'Volumes[].[VolumeId, VolumeType]'
[
  [
    "vol-e11a5288",
    "standard"
  ],
  [
    "vol-2e410a47",
    "standard"
  ],
  [
    "vol-a1b3c7nd",
    "standard"
  ]
]
```

若要將巢狀資料新增至清單，請新增另一個複選清單。以下範例會透過在巢狀 Attachments 清單中篩選 InstanceId 和 State 來擴展上述範例。成果如以下表達式。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[].[VolumeId, VolumeType, Attachments[].[InstanceId, State]]'
[
  [
    "vol-e11a5288",
    "standard",
    [
      [
        "i-a071c394",
        "attached"
      ]
    ]
  ],
  [
    "vol-2e410a47",
    "standard",
    [
      [
        "i-4b41a37c",
        "attached"
      ]
    ]
  ],
  [
```

```
    "vol-a1b3c7nd",
    "standard",
    [
      [
        "i-1jd73kv8",
        "attached"
      ]
    ]
  ]
]
```

為了更具可讀性，會如下列範例所示展平表達式。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[].[VolumeId, VolumeType, Attachments[].[InstanceId, State][]][]'
[
  "vol-e11a5288",
  "standard",
  [
    "i-a071c394",
    "attached"
  ],
  "vol-2e410a47",
  "standard",
  [
    "i-4b41a37c",
    "attached"
  ],
  "vol-a1b3c7nd",
  "standard",
  [
    "i-1jd73kv8",
    "attached"
  ]
]
```

如需詳細資訊，請參閱 JMESPath 網站上的複選清單。

將標籤新增至識別碼值

為了使此輸出更容易閱讀，請使用具有以下語法的複選雜湊。

語法

```
<listName>[].{<label>: <expression>, <label>: <expression>}
```

您的識別碼標籤不需與識別碼的名稱相同。以下範例會將 VolumeType 標籤用於 VolumeType 值。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[].{VolumeType: VolumeType}'
[
  {
    "VolumeType": "standard",
  },
  {
    "VolumeType": "standard",
  },
  {
    "VolumeType": "standard",
  }
]
```

為了簡單起見，下列範例會保留每個標籤的識別碼名稱，並顯示所有磁碟區的 VolumeId、VolumeType、InstanceId 以及 State：

```
$ aws ec2 describe-volumes \
    --query 'Volumes[].{VolumeId: VolumeId, VolumeType: VolumeType, InstanceId:
 Attachments[0].InstanceId, State: Attachments[0].State}'
[
  {
    "VolumeId": "vol-e11a5288",
    "VolumeType": "standard",
    "InstanceId": "i-a071c394",
    "State": "attached"
  },
  {
    "VolumeId": "vol-2e410a47",
    "VolumeType": "standard",
    "InstanceId": "i-4b41a37c",
    "State": "attached"
  },
  {
    "VolumeId": "vol-a1b3c7nd",
    "VolumeType": "standard",
    "InstanceId": "i-1jd73kv8",
    "State": "attached"
```

```
  }
]
```

如需詳細資訊，請參閱 JMESPath 網站上的複選雜湊。

函數

JMESPath 語法包含許多可用於查詢的功能。如需 JMESPath 函數的相關資訊，請參閱 JMESPath 網站上的內建函數。

為了示範如何將函數合併至您的查詢中，下列範例會使用 `sort_by` 函數。`sort_by` 函數會使用以下語法，利用表達式作為排序索引鍵來排序陣列：

語法

```
sort_by(<listName>, <sort expression>)[].<expression>
```

下列範例會使用先前的複選雜湊範例，並藉由 `VolumeId` 來排序輸出。

```
$ aws ec2 describe-volumes \
    --query 'sort_by(Volumes, &VolumeId)[].{VolumeId: VolumeId, VolumeType: VolumeType,
 InstanceId: Attachments[0].InstanceId, State: Attachments[0].State}'
[
  {
    "VolumeId": "vol-2e410a47",
    "VolumeType": "standard",
    "InstanceId": "i-4b41a37c",
    "State": "attached"
  },
  {
    "VolumeId": "vol-a1b3c7nd",
    "VolumeType": "standard",
    "InstanceId": "i-1jd73kv8",
    "State": "attached"
  },
  {
    "VolumeId": "vol-e11a5288",
    "VolumeType": "standard",
    "InstanceId": "i-a071c394",
    "State": "attached"
  }
]
```

如需詳細資訊，請參閱 JMESPath 網站上的 sort_by。

進階 **--query** 範例

從特定項目中擷取資訊

以下範例使用 --query 參數來尋找清單中的特定項目，然後從該項目中擷取資訊。該範例會列出與指定服務端點相關的所有 AvailabilityZones。它從具有指定 ServiceName 的 ServiceDetails 清單擷取項目，然後從該選取項目輸出 AvailabilityZones 欄位。

```
$ aws --region us-east-1 ec2 describe-vpc-endpoint-services \
    --query 'ServiceDetails[?ServiceName==`com.amazonaws.us-
east-1.ecs`].AvailabilityZones'
[
    [
        "us-east-1a",
        "us-east-1b",
        "us-east-1c",
        "us-east-1d",
        "us-east-1e",
        "us-east-1f"
    ]
]
```

顯示指定建立日期之後的快照

以下範例說明如何列出在指定日期之後建立的所有快照，並只包括輸出中的幾個可用欄位。

```
$ aws ec2 describe-snapshots --owner self \
    --output json \
    --query 'Snapshots[?StartTime>=`2018-02-07`].
{Id:SnapshotId,VId:VolumeId,Size:VolumeSize}'
[
    {
        "id": "snap-0effb42b7a1b2c3d4",
        "vid": "vol-0be9bb0bf12345678",
        "Size": 8
    }
]
```

顯示最近的 AMI

以下範例會列出您建立的 5 個最新 Amazon Machine Image (AMI)，依最新到最舊排列。

```
$ aws ec2 describe-images \
    --owners self \
    --query 'reverse(sort_by(Images,&CreationDate))[:5].{id:ImageId,date:CreationDate}'
[
    {
        "id": "ami-0a1b2c3d4e5f60001",
        "date": "2018-11-28T17:16:38.000Z"
    },
    {
        "id": "ami-0a1b2c3d4e5f60002",
        "date": "2018-09-15T13:51:22.000Z"
    },
    {
        "id": "ami-0a1b2c3d4e5f60003",
        "date": "2018-08-19T10:22:45.000Z"
    },
    {
        "id": "ami-0a1b2c3d4e5f60004",
        "date": "2018-05-03T12:04:02.000Z"
    },
    {
        "id": "ami-0a1b2c3d4e5f60005",
        "date": "2017-12-13T17:16:38.000Z"
    }
]
```

顯示狀況不良的 Auto Scaling 執行個體

以下範例僅會顯示指定 Auto Scaling 群組中任何狀況不良的執行個體的 `InstanceId`。

```
$ aws autoscaling describe-auto-scaling-groups \
    --auto-scaling-group-name My-AutoScaling-Group-Name \
    --output text \
    --query 'AutoScalingGroups[*].Instances[?HealthStatus==`Unhealthy`].InstanceId'
```

包含有指定標籤的磁碟區

以下範例描述所有具備 `test` 標籤的執行個體。只要 `test` 旁邊有另一個附加至磁碟區的標籤，則磁碟區仍會傳回結果中。

下列表達式傳回陣列中具有 `test` 標籤的所有標籤。任何不是 `test` 標籤的標籤都包含 `null` 值。

```
$ aws ec2 describe-volumes \
```

```
    --query 'Volumes[*].Tags[?Value == `test`]'
```

排除有指定標籤的磁碟區

以下範例描述所有沒有 test 標籤的執行個體。使用簡易 ?Value != `test` 表達式之方式，不適用於排除磁碟區，原因是磁碟區可擁有多個標籤。只要 test 旁邊有另一個附加至磁碟區的標籤，則磁碟區仍會傳回結果中。

若要排除所有具有 test 標籤的磁碟區，請從以下表達式開始傳回陣列中所有具有 test 標籤的標籤。任何不是 test 標籤的標籤都包含 null 值。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Tags[?Value == `test`]'
```

接著，請使用 not_null 函數篩選掉所有正向 test 結果。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[?!not_null(Tags[?Value == `test`].Value)]'
```

輸送結果，以展平會導致以下查詢的結果。

```
$ aws ec2 describe-volumes \
    --query 'Volumes[?!not_null(Tags[?Value == `test`].Value)] | []'
```

結合伺服器端和用戶端篩選

您可以將伺服器端和用戶端篩選搭配著使用。先完成伺服器端篩選，再將資料傳送至 --query 參數稍後要篩選的用戶端。如果您使用的是大型資料集，則先使用伺服器端篩選可以降低每次 AWS CLI 呼叫傳送至用戶端的資料量，同時仍保留用戶端篩選提供的強大自訂功能。

以下範例會列出同時使用伺服器端和用戶端篩選的 Amazon EC2 磁碟區。此服務會篩選 us-west-2a 可用區域內所有附加磁碟區。--query 參數會將輸出進一步限制為 Size 數值大於 50 的那些磁碟區，並只顯示具有使用者定義名稱的指定欄位。

```
$ aws ec2 describe-volumes \
    --filters "Name=availability-zone,Values=us-west-2a" "Name=status,Values=attached"
 \
    --query 'Volumes[?Size > `50`].{Id:VolumeId,Size:Size,Type:VolumeType}'
[
    {
        "Id": "vol-0be9bb0bf12345678",
```

```
        "Size": 80,
        "VolumeType": "gp2"
    }
]
```

以下範例擷取符合幾個條件的影像清單。接著使用 `--query` 參數來依照 `CreationDate` 排序輸出，只選擇最新的項目。最後，顯示該映像的 `ImageId`。

```
$ aws ec2 describe-images \
    --owners amazon \
    --filters "Name=name,Values=amzn*gp2" "Name=virtualization-type,Values=hvm"
 "Name=root-device-type,Values=ebs" \
    --query "sort_by(Images, &CreationDate)[-1].ImageId" \
    --output text
ami-00ced3122871a4921
```

以下範例透過使用 `length` 計算清單中有多少項目數，顯示超過 1000 IOPS 的可用磁碟區數。

```
$ aws ec2 describe-volumes \
    --filters "Name=status,Values=available" \
    --query 'length(Volumes[?Iops > `1000`])'
3
```

下列範例會擷取在使用 AWS CloudFormation 堆疊的指定 中，使用啟動組態 AWS 區域 的 Auto Scaling 群組名稱。

```
$ aws autoscaling describe-auto-scaling-groups --region us-west-2 \
  --filters Name=tag-key,Values=aws:cloudformation:stack-name \
  --query 'AutoScalingGroups[?LaunchConfigurationName!=`null`].AutoScalingGroupName'
[
    "group-1",
    "group-2",
    "group-3"
]
```

其他資源

AWS CLI 自動提示

開始使用篩選條件表達式時，您可以使用第 2 AWS CLI 版中的自動提示功能。自動提示功能可在您按下 F5 鍵時提供預覽。如需詳細資訊，請參閱 the section called “自動提示”。

JMESPath 終端機

JMESPath 終端機為交互式終端命令，與用於用戶端篩選的 JMESPath 表達式一起進行實驗。使用 `jpterm` 命令時，終端機會在您輸入時立即顯示查詢結果。您可以直接將 AWS CLI 輸出輸送至終端機，啟用進階查詢實驗。

下列範例會直接輸送 `aws ec2 describe-volumes` 輸出至 JMESPath 終端機。

```
$ aws ec2 describe-volumes | jpterm
```

如需 JMESPath 終端機和安裝說明的詳細資訊，請參閱 GitHub 上的 JMESPath 終端機。

jq 公用程式

`jq` 公用程式會提供一個方法，讓您能將用戶端的輸出轉換為您要的輸出格式。如需 `jq` 和安裝說明的詳細資訊，請參閱 GitHub 上的 jq。

中的命令列傳回代碼 AWS CLI

傳回碼通常是在執行 a AWS Command Line Interface (AWS CLI) 命令後傳送的隱藏程式碼，描述命令的狀態。您可以使用 `echo`命令來顯示從最後一個 AWS CLI 命令傳送的程式碼，並使用這些程式碼來判斷命令是否成功或失敗，以及為什麼命令可能發生錯誤。除了傳回碼之外，您還可以使用 `--debug` 交換器來檢視有關執行命令失敗的更多詳細資訊。此交換器會產生 AWS CLI 用於處理命令的步驟的詳細報告，以及每個步驟的結果。

若要判斷 AWS CLI 命令的傳回碼，請在執行 CLI 命令後立即執行下列其中一個命令。

Linux and macOS

```
$ echo $?
0
```

Windows PowerShell

```
PS> echo $lastexitcode
0
```

Windows Command Prompt

```
C:\> echo %errorlevel%
```

```
0
```

以下是可在執行 a AWS Command Line Interface (AWS CLI) 命令結束時傳回的傳回碼值。

代碼	意義
0	服務回應的 HTTP 回應狀態碼為 200，表示 AWS CLI 和 服務未產生錯誤，請求已傳送到該 AWS 服務。
1	一或多個 Amazon S3 傳輸操作失敗。限於 S3 命令。
2	此傳回碼的意義取決於命令： • 適用於所有 AWS CLI 命令 – 無法剖析輸入的命令。剖析失敗的原因可能包括 (但不限於) 遺漏必要的子命令或引數，或使用不明命令或參數。 • 限於 S3 命令 – 在傳輸過程中略過一或多個已標示為傳輸的檔案。不過，所有其他標示為傳輸的檔案已成功傳輸。在傳輸過程中略過的檔案包括：不存在的檔案、字元特殊裝置、區塊特殊裝置、FIFO 佇列或通訊埠的檔案，以及使用者不具有讀取許可的檔案。
130	命令被 SIGINT 中斷。這便是您傳送的以 `Ctrl+C` 取消命令的訊號。
252	命令語法無效、提供了未知參數，或參數值不正確，導致命令無法執行。
253	系統環境或組態無效。雖然提供的命令在語法上可能有效，但缺少組態或憑證會導致命令無法執行。
254	已成功剖析命令，並向指定的服務發出請求，但服務傳回錯誤。這通常表示 API 使用不正確，或發生其他服務特定問題。
255	命令失敗。 AWS CLI 或 AWS 傳送請求的服務產生錯誤。

使用自訂精靈在 中執行互動式命令 AWS CLI

AWS Command Line Interface (AWS CLI) 提供對某些命令使用精靈的功能。若要貢獻或檢視可用 AWS CLI 精靈的完整清單，請參閱 GitHub 上的AWS CLI 精靈資料夾。

運作方式

與 AWS 主控台類似， AWS CLI 具有 UI 精靈，可引導您管理 AWS 資源。若要使用精靈，您可以呼叫 `wizard` 子命令，以及命令中服務名稱後的精靈名稱。命令結構如下所示：

語法:

```
$ aws <command> wizard <wizardName>
```

下列範例為呼叫精靈建立新的 dynamodb 資料表。

```
$ aws dynamodb wizard new-table
```

`aws configure` 是唯一沒有精靈名稱的精靈。在執行精靈時，請依照下面範例示範的方式來執行 `aws configure wizard` 命令：

```
$ aws configure wizard
```

在呼叫精靈之後，系統會顯示 Shell 中的表單。對於各個參數，系統會提供一個選項清單供您選取，或提示您輸入字串。若要從清單中選取，請使用向上鍵和向下鍵，然後按 ENTER。若要檢視某個選項的詳細資訊，請按向右鍵。在您完成填寫參數後，請按 ENTER。

```
$ aws configure wizard
What would you like to configure
> Static Credentials
  Assume Role
  Process Provider
  Additional CLI configuration
Enter the name of the profile:
Enter your Access Key Id:
Enter your Secret Access Key:
```

若要編輯先前的提示，請使用 SHIFT + TAB。對於某些精靈，在填入所有提示之後，您可以預覽 AWS CloudFormation 範本或填入您資訊的 AWS CLI 命令。此預覽模式有助於了解 AWS CLI、 服務 APIs，以及建立指令碼的範本。

請在檢視或最後提示結束後，按 ENTER 來執行最終命令。

```
$ aws configure wizard
What would you like to configure
```

```
Enter the name of the profile: testWizard
Enter your Access Key Id: AB1C2D3EF4GH5I678J90K
Enter your Secret Access Key: ab1c2def34gh5i67j8k90l1mnop2qr3s45tu678v90
<ENTER>
```

在 中建立和使用別名 AWS CLI

別名是您可以在 AWS Command Line Interface (AWS CLI) 中建立的捷徑，以縮短您經常使用的命令或指令碼。您可以在位於組態資料夾的 `alias` 檔案中建立別名。

主題

- 先決條件
- 步驟 1：建立別名檔案
- 步驟 2：建立別名
- 步驟 3：呼叫別名
- 別名儲存庫範例
- 資源

先決條件

若要使用別名命令，您需要完成以下事項：

- 安裝及設定 AWS CLI。如需詳細資訊，請參閱the section called “安裝/更新”及的身分驗證和存取憑證 AWS CLI。
- 使用最低 AWS CLI 版本 1.11.24 或 2.0.0。
- （選用）若要使用別名 bash AWS CLI 指令碼，您必須使用與 bash 相容的終端機。

步驟 1：建立別名檔案

若要建立 `alias` 檔案，您可以使用檔案導覽和文字編輯器，或使用您偏好的終端機並按照逐步程序來建立。若要快速建立別名檔案，請使用以下命令區塊。

Linux and macOS

```
$ mkdir -p ~/.aws/cli
$ echo '[toplevel]' > ~/.aws/cli/alias
```

Windows

```
C:\> md %USERPROFILE%\.aws\cli
C:\> echo [toplevel] > %USERPROFILE%/.aws/cli/alias
```

建立別名檔案

1. 在 AWS CLI 組態資料夾中建立名為 cli的資料夾。預設情況下，組態資料夾是 ~/.aws/ (Linux 或 macOS) 或 %USERPROFILE%\.aws\ (Windows)。您可以透過檔案導覽或使用以下命令來建立此資料夾。

   Linux and macOS

   ```
   $ mkdir -p ~/.aws/cli
   ```

   Windows

   ```
   C:\> md %USERPROFILE%\.aws\cli
   ```

   產生的 cli 資料夾預設路徑為 ~/.aws/cli/ (Linux 或 macOS) 或 %USERPROFILE%\.aws\cli (Windows)。

2. 在 cli 資料夾中，建立沒有副檔名且名為 alias 的文字檔案，並將 [toplevel] 新增至第一行。您可以透過偏好的文字編輯器或使用下列命令來建立此檔案。

   Linux and macOS

   ```
   $ echo '[toplevel]' > ~/.aws/cli/alias
   ```

   Windows

   ```
   C:\> echo [toplevel] > %USERPROFILE%/.aws/cli/alias
   ```

步驟 2：建立別名

您可以使用基本命令或 Bash 指令碼來建立別名。

建立基本命令別名

您可以在上一步建立的 alias 檔案中，使用下列語法新增命令以建立別名。

語法

```
aliasname = command [--options]
```

aliasname 是您所命名的別名。*command* 為您要呼叫的命令，此命令可包含其他別名。您可以在別名中加入選項或參數，或在呼叫別名時新增選項或參數。

以下範例使用 aws sts get-caller-identity 命令來建立名為 aws whoami 的別名。由於此別名會呼叫現有的 AWS CLI 命令，因此您編寫的命令無需添加 aws 前綴。

```
whoami = sts get-caller-identity
```

下列範例採用之前的 whoami 範例，並新增 Account 篩選條件和文字 output 選項。

```
whoami2 = sts get-caller-identity --query Account --output text
```

建立子命令別名

> **Note**
>
> 子命令別名功能需要 1.11.24 或 2.0.0 的最低 AWS CLI 版本

您可以在上一步建立的 alias 檔案中，使用下列語法新增命令以建立子命令的別名。

語法

```
[command commandGroup]
aliasname = command [--options]
```

commandGroup 是命令名稱空間，例如，命令 aws ec2 describe-regions 在 ec2 命令群組下。*aliasname* 是您所命名的別名。*command* 為您要呼叫的命令，此命令可包含其他別名。您可以在別名中加入選項或參數，或在呼叫別名時新增選項或參數。

以下範例使用 aws ec2 describe-regions 命令來建立名為 aws ec2 regions 的別名。由於此別名會呼叫 ec2 命令命名空間下現有的 AWS CLI 命令，因此您編寫的命令無需添加 aws ec2 字首。

```
[command ec2]
regions = describe-regions --query Regions[].RegionName
```

若要從命令名稱空間之外的命令建立別名，請在完整命令前加上驚嘆號。以下範例使用 [aws iam list-instance-profiles] 命令來建立名為 aws ec2 instance-profiles 的別名。

```
[command ec2]
instance-profiles = !aws iam list-instance-profiles
```

> **Note**
>
> 別名只使用現有的命令命名空間，您不能建立新的命令空間。例如，由於 johnsmith 命令名空間尚未存在，因此您無法使用 [command johnsmith] 區段建立別名。

建立 Bash 指令碼別名

> **Warning**
>
> 若要使用別名 bash AWS CLI 指令碼，您必須使用與 bash 相容的終端機

您可以透過以下語法，在更進階的流程中使用 Bash 指令碼建立別名。

語法

```
aliasname =
    !f() {
        script content
}; f
```

aliasname 是您所命名的別名，*script content* 則是您在呼叫別名時要執行的指令碼。

以下範例使用 opendns 來輸出您目前的 IP 地址。由於您可以在其他別名中使用別名，因此以下 myip 別名，可用於從其他別名中允許或撤銷 IP 地址的存取權限。

```
myip =
  !f() {
```

```
    dig +short myip.opendns.com @resolver1.opendns.com
  }; f
```

下列指令碼範例會呼叫上一個 aws myip 別名，以授予您的 IP 地址進入 Amazon EC2 安全群組的權限。

```
authorize-my-ip =
  !f() {
    ip=$(aws myip)
    aws ec2 authorize-security-group-ingress --group-id ${1} --cidr $ip/32 --protocol
 tcp --port 22
  }; f
```

在您呼叫使用 Bash 指令碼的別名時，變數一律會按照您輸入的順序傳遞。在 Bash 指令碼中，變數名稱不是考慮因素，僅會考量其顯示順序。在下列 textalert 別名範例中，--message 選項的變數為第一個，而 --phone-number 選項為第二個。

```
textalert =
  !f() {
    aws sns publish --message "${1}" --phone-number ${2}
  }; f
```

步驟 3：呼叫別名

若要執行您在 alias 檔案中建立的別名，請使用以下語法。您可以在呼叫別名時新增其他選項。

語法

```
$ aws aliasname
```

下列範例使用 aws whoami 命令別名。

```
$ aws
 whoami
{
    "UserId": "A12BCD34E5FGHI6JKLM",
    "Account": "1234567890987",
    "Arn": "arn:aws:iam::1234567890987:user/userName"
}
```

以下範例使用 aws whoami 別名以及其他選項，以便只將 Account 號碼回傳到 text 輸出中。

```
$ aws whoami --query Account --output
 text
1234567890987
```

下列範例使用 aws ec2 regions 子命令別名。

```
$ aws ec2
 regions
[
    "ap-south-1",
    "eu-north-1",
    "eu-west-3",
    "eu-west-2",
...
```

使用 Bash 指令碼變數呼叫別名

在您呼叫使用 Bash 指令碼的別名時，變數會按照您輸入的順序傳遞。在 Bash 指令碼中，變數名稱不是考慮因素，僅會考量其顯示順序。例如，在下列 textalert 別名中，--message 選項的變數為第一個，而 --phone-number 為第二個。

```
textalert =
  !f() {
    aws sns publish --message "${1}" --phone-number ${2}
  }; f
```

在呼叫 textalert 別名時，您必須按照變數在別名中的執行順序來傳遞變數。在以下範例中，我們使用變數 $message 和 $phone。$message 變數會作為 --message 選項的 ${1} 傳遞，$phone 變數則作為 --phone-number 選項的 ${2} 傳遞。如此會成功呼叫 textalert 別名來傳送訊息。

```
$ aws textalert $message
 $phone
{
    "MessageId": "1ab2cd3e4-fg56-7h89-i01j-2klmn34567"
}
```

在下列範例中，呼叫 $phone 和 $message 別名時會對調順序。$message 變數會作為 --message 選項的 ${1} 傳遞，$phone 變數則作為 --phone-number 選項的 ${2} 傳遞。由於變數順序不正

確，別名會錯誤傳遞變數。因為 $message 的內容與 --phone-number 選項的電話號碼格式需求不相符，所以會導致錯誤發生。

```
$ aws textalert $phone
 $message
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

  aws help
  aws <command> help
  aws <command> <subcommand> help

Unknown options: text
```

別名儲存庫範例

GitHub 上的AWS CLI 別名儲存庫包含 AWS CLI 開發人員團隊和社群建立的 AWS CLI 別名範例。您可以使用整個 alias 檔案範例，或者自行採用個別別名。

> **Warning**
>
> 執行本節中的命令會刪除您現有的 alias 檔案。為了避免覆寫現有別名檔案，請變更下載位置。

使用儲存庫中的別名

1. 安裝 Git。如需安裝說明，請參閱 Git 文件中的新手入門 – 安裝 Git。
2. 安裝 jp 命令。jp 命令是用於 tostring 別名。如需安裝說明，請參閱 GitHub 上的 JMESPath (jp) README.md。
3. 安裝 jq 命令。jq 命令是用於 tostring-with-jq 別名。如需安裝說明，請參閱 GitHub 上的 JSON processor (jq)。
4. 執行下列其中一項操作來下載 alias 檔案：

   - 執行以下命令，即可從儲存庫下載 alias 檔案並將它複製到組態資料夾。

     Linux and macOS

     ```
     $ git clone https://github.com/awslabs/awscli-aliases.git
     $ mkdir -p ~/.aws/cli
     ```

```
$ cp awscli-aliases/alias ~/.aws/cli/alias
```

Windows

```
C:\> git clone https://github.com/awslabs/awscli-aliases.git
C:\> md %USERPROFILE%\.aws\cli
C:\> copy awscli-aliases\alias %USERPROFILE%\.aws\cli
```

- 直接從儲存庫下載，並儲存至 AWS CLI 組態`cli`資料夾中的 資料夾。預設情況下，組態資料夾是 `~/.aws/` (Linux 或 macOS) 或 `%USERPROFILE%\.aws\` (Windows)。

5. 若要驗證別名是否正常運作，請執行以下別名。

```
$ aws whoami
```

如此會顯示與 `aws sts get-caller-identity` 命令相同的回應：

```
{
    "Account": "012345678901",
    "UserId": "AIUAINBADX2VEG2TC6HD6",
    "Arn": "arn:aws:iam::012345678901:user/myuser"
}
```

資源

- GitHub 上的AWS CLI 別名儲存庫包含 AWS CLI 開發人員團隊建立的 AWS CLI 別名範例，以及社群的 AWS CLI 貢獻。
- 來自 AWS re：Invent 2016 的別名功能公告：YouTube 上的有效 AWS CLI 使用者。 YouTube
- `aws sts get-caller-identity`
- `aws ec2 describe-instances`
- `aws sns publish`

的程式碼範例 AWS CLI

本章提供範例的集合，說明如何搭配 AWS Command Line Interface (AWS CLI) 使用 AWS 服務。

本指南中有 AWS CLI 下列類型的範例：

- 引導式命令範例 - AWS CLI 使用者指南的引導式命令範例，說明如何 AWS CLI 搭配部分 使用 AWS 服務。這些通常是比第 AWS CLI 2 版參考指南中的範例更詳細的範例。
- AWS CLI 命令範例 - 也提供參考第 AWS CLI 2 版參考指南中的開放原始碼命令範例。命令範例託管於 GitHub 上的 AWS CLI 儲存庫中。
- AWS CLI 使用 Bash 指令碼程式碼範例 - 開放原始碼 bash 指令碼範例。Bash 指令碼範例託管於 GitHub 上的 AWS 程式碼範例儲存庫中。

> **範例意見回饋**
>
> 找不到所需的內容嗎？ 使用此頁面底部的提供意見回饋連結，或在第 AWS CLI 2 版參考指南中的相關命令頁面上，請求命令範例。
> 想要貢獻嗎？ GitHub 上AWS 程式碼範例儲存庫中的 Contribute AWS CLI 命令範例。如需貢獻的詳細資訊，請參閱 GitHub 頁面上的AWS CLI 程式碼範例貢獻快速步驟。

的引導式命令範例 AWS CLI

AWS Command Line Interface (AWS CLI) 是一種開放原始碼工具，可讓您 AWS 服務 在命令列 Shell 中使用各種命令與 互動。本節提供引導式範例，示範如何利用 AWS CLI 來存取部分 AWS 服務。這包括一些自訂 AWS CLI 命令，例如高階`aws s3`命令。這些命令範例示範用於某些 的常見動作 AWS 服務 ，並提供其他資源以取得詳細資訊。

無論您是經驗豐富的 AWS 使用者還是初次使用 AWS CLI，這些引導式範例都可做為簡化 AWS 操作的資源。

如需每個 所有可用命令的完整參考 AWS 服務，請參閱第 AWS CLI 2 版參考指南。此外，您可以使用內建命令列說明來探索 中的 AWS 服務、 命令、選項和功能陣列 AWS CLI。

如需本節中無法使用的更多命令範例，請參閱AWS CLI 命令範例一節。這些是開放原始碼命令範例，也可在第 AWS CLI 2 版參考指南中找到。命令範例託管於 GitHub 上的 AWS CLI 儲存庫中。

如需開放原始碼 bash 指令碼範例，請參閱 the section called “Bash 指令碼範例”。Bash 指令碼範例託管於 GitHub 上的 AWS 程式碼範例儲存庫中。

服務

- 在 中使用 Amazon DynamoDB AWS CLI
- 在 中使用 Amazon EC2 AWS CLI
- 在 中使用 Amazon S3 Glacier AWS CLI
- 在 中使用 IAM AWS CLI
- 在 中使用 Amazon S3 AWS CLI
- 在 中存取 Amazon SNS AWS CLI

在 中使用 Amazon DynamoDB AWS CLI

AWS Command Line Interface (AWS CLI) 支援所有 AWS 資料庫服務，包括 Amazon DynamoDB。您可以使用 AWS CLI 進行臨時操作，例如建立資料表。您也可以用其在公用程式指令碼中嵌入 DynamoDB 操作。

如需 AWS CLI 搭配 DynamoDB 使用 的詳細資訊，請參閱《 AWS CLI 命令參考dynamodb》中的 。

若要列出 DynamoDB 的 AWS CLI 命令，請使用下列命令。

```
$ aws dynamodb help
```

主題

- 先決條件
- 建立和使用 DynamoDB 資料表
- 使用 DynamoDB Local
- 資源

先決條件

若要執行 dynamodb 命令，您需要：

- 安裝及設定 AWS CLI。如需詳細資訊，請參閱the section called “安裝/更新”及的身分驗證和存取憑證 AWS CLI。

建立和使用 DynamoDB 資料表

命令列格式包含 DynamoDB 命令名稱，隨後接著該命令的參數。 AWS CLI 支援參數值和完整 JSON 的 CLI 速記語法。

以下範例會建立名為 `MusicCollection` 的資料表。

```
$ aws dynamodb create-table \
    --table-name MusicCollection \
    --attribute-definitions AttributeName=Artist,AttributeType=S
 AttributeName=SongTitle,AttributeType=S \
    --key-schema AttributeName=Artist,KeyType=HASH
 AttributeName=SongTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=1,WriteCapacityUnits=1
```

您可以使用類似於以下範例中所示的命令，新增行到資料表。這些範例混合使用速記語法和 JSON。

```
$ aws dynamodb put-item \
    --table-name MusicCollection \
    --item '{
        "Artist": {"S": "No One You Know"},
        "SongTitle": {"S": "Call Me Today"} ,
        "AlbumTitle": {"S": "Somewhat Famous"}
      }' \
    --return-consumed-capacity TOTAL
{
    "ConsumedCapacity": {
        "CapacityUnits": 1.0,
        "TableName": "MusicCollection"
    }
}
```

```
$ aws dynamodb put-item \
    --table-name MusicCollection \
    --item '{
        "Artist": {"S": "Acme Band"},
        "SongTitle": {"S": "Happy Day"} ,
        "AlbumTitle": {"S": "Songs About Life"}
      }' \
    --return-consumed-capacity TOTAL
{
    "ConsumedCapacity": {
```

```
        "CapacityUnits": 1.0,
        "TableName": "MusicCollection"
    }
}
```

在單行命令中編寫有效的 JSON 可能很困難。為了簡化此作業， AWS CLI 可以讀取 JSON 檔案。例如，請試想下列 JSON 程式碼片段，其存放在名為 expression-attributes.json 的檔案中。

```
{
  ":v1": {"S": "No One You Know"},
  ":v2": {"S": "Call Me Today"}
}
```

您可以使用該檔案來使用 AWS CLI發出 query 請求。在下列範例中，expression-attributes.json 檔案的內容會作為 --expression-attribute-values 參數的數值使用。

```
$ aws dynamodb query --table-name MusicCollection \
    --key-condition-expression "Artist = :v1 AND SongTitle = :v2" \
    --expression-attribute-values file://expression-attributes.json
{
    "Count": 1,
    "Items": [
        {
            "AlbumTitle": {
                "S": "Somewhat Famous"
            },
            "SongTitle": {
                "S": "Call Me Today"
            },
            "Artist": {
                "S": "No One You Know"
            }
        }
    ],
    "ScannedCount": 1,
    "ConsumedCapacity": null
}
```

使用 DynamoDB Local

除了 DynamoDB 之外，您還可以 AWS CLI 搭配 DynamoDB Local 使用 。DynamoDB Local 是模擬 DynamoDB 服務的小型用戶端資料庫及伺服器。DynamoDB Local 可讓您編寫使用 DynamoDB API 的

應用程式，無須處理 DynamoDB Web 服務中的任何表格與資料。所有 API 動作均會重新路由至本機資料庫。如此可讓您節省佈建輸送、資料儲存和數據傳輸費。

如需 DynamoDB Local 以及如何搭配 使用的詳細資訊 AWS CLI，請參閱《Amazon DynamoDB 開發人員指南》中的下列章節：

- DynamoDB Local
- 搭配 DynamoDB Local 使用 AWS CLI

資源

AWS CLI 參考：

- `aws dynamodb`
- `aws dynamodb create-table`
- `aws dynamodb put-item`
- `aws dynamodb query`

服務參考：

- 《Amazon DynamoDB 開發人員指南》中的 DynamoDB Local
- 《Amazon DynamoDB 開發人員指南》中的搭配 DynamoDB Local 使用 AWS CLI

在 中使用 Amazon EC2 AWS CLI

Amazon Elastic Compute Cloud 簡介
Amazon EC2 簡介 - 使用 的 Elastic Cloud Server 和託管 AWS

Amazon Elastic Compute Cloud (Amazon EC2) 提供高度可擴展且靈活的虛擬運算環境。Amazon EC2 可讓您佈建和管理稱為 Amazon EC2 執行個體的虛擬伺服器，以滿足各種運算需求。

Amazon EC2 執行個體是虛擬機器，可透過 CPU、記憶體、儲存和聯網功能的各種組態進行自訂。您可以選擇各種執行個體類型，從輕量型、經濟實惠的選項到功能強大且高效能的執行個體，視您的應用程式需求而定。此彈性可讓您符合運算需求，以最佳化效能和成本效益。

此外，Amazon EC2 提供一套功能，讓您能夠有效管理運算資源。這些功能包括快速啟動新執行個體、建立自訂機器映像 (AMIs) 以進行快速部署，以及視需要擴展或縮減運算容量。

您可以使用 AWS Command Line Interface () 存取 Amazon EC2 的功能AWS CLI。若要列出 Amazon EC2 的 AWS CLI 命令，請使用下列命令。

```
aws ec2 help
```

在執行任何命令前，請先設定您的預設憑證。如需詳細資訊，請參閱設定 的設定 AWS CLI。

本主題顯示執行 Amazon EC2 常見任務的 AWS CLI 命令的簡短格式範例。

如需 AWS CLI 命令的長格式範例，請參閱 GitHub 上的AWS CLI 程式碼範例儲存庫。

主題

- 在 中建立、顯示和刪除 Amazon EC2 金鑰對 AWS CLI
- 在 中建立、設定和刪除 Amazon EC2 安全群組 AWS CLI
- 在 中啟動、列出和刪除 Amazon EC2 執行個體 AWS CLI
- 在 中使用 bash 指令碼變更 Amazon EC2 執行個體類型 AWS CLI

在 中建立、顯示和刪除 Amazon EC2 金鑰對 AWS CLI

您可以使用 AWS Command Line Interface (AWS CLI) 來建立、顯示和刪除 Amazon Elastic Compute Cloud (Amazon EC2) 的金鑰對。使用金鑰對來連線到 Amazon EC2 執行個體。您可以在建立執行個體時提供金鑰對給 Amazon EC2，然後在連線至執行個體時使用該金鑰對進行身分驗證。

> **Note**
>
> 如需其他命令範例，請參閱AWS CLI 參考指南。

主題

- 先決條件
- 建立金鑰對
- 顯示您的金鑰對
- 刪除您的金鑰對

- 參考

先決條件

若要執行 ec2 命令，您需要：

- 安裝及設定 AWS CLI。如需詳細資訊，請參閱the section called “安裝/更新”及的身分驗證和存取憑證 AWS CLI。
- 設定 IAM 許可，以允許 Amazon EC2 存取。如需 Amazon EC2 IAM 許可的詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Amazon EC2 的 IAM 政策。 Amazon EC2

建立金鑰對

若要建立金鑰對，請使用 aws ec2 create-key-pair 命令搭配 --query 選項和 --output text 選項，將您的私有金鑰直接輸送到檔案中。

```
$ aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text
 > MyKeyPair.pem
```

對於 PowerShell，> file 重新導向預設為 UTF-8 編碼，不適用於某些 SSH 用戶端。因此，您必須將輸出輸送到 out-file 命令並明確設定編碼為 ascii，以轉換輸出。

```
PS C:\>aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text
 | out-file -encoding ascii -filepath MyKeyPair.pem
```

產生的 MyKeyPair.pem 檔案看起來類似下列。

```
-----BEGIN RSA PRIVATE KEY-----
EXAMPLEKEYKCAQEAy7WZhaDsrA1W3mRlQtvhwyORRX8gnxgDAfRt/gx42kWXsT4rXE/b5CpSgie/
vBoU7jLxx92pNHoFnByP+Dc21eyyz6CvjTmWA0JwfWiW5/akH7iO5dSrvC7dQkW2duV5QuUdE0QW
Z/aNxMniGQE6XAgfwlnXVBwrerrQo+ZWQeqiUwwMkuEbLeJFLhMCvYURpUMSC1oehm449ilx9X1F
G50TCFeOzfl8dqqCP6GzbPaIjiU19xX/azOR9V+tpUOzEL+wmXnZt3/nHPQ5xvD2OJH67km6SuPW
oPzev/D8V+x4+bHthfSjR9Y7DvQFjfBVwHXigBdtZcU2/wei8D/HYwIDAQABAoIBAGZ1kaEvnrqu
/uler7vgIn5m7lN5LKw4hJLAIW6tUT/fzvtcHK0SkbQCQXuriHmQ2MQyJX/0kn2NfjLV/ufGxbL1
mb5qwMGUnEpJaZD6QSSs3kICLwWUYUiGfc0uiSbmJoap/GTLU0W5Mfcv36PaBUNy5p53V6G7hXb2
bahyWyJNfjLe4M86yd2YK3V2CmK+X/BOsShnJ36+hjrXPPWmV3N9zEmCdJjA+K15DYmhm/tJWSD9
81oGk9TopEp7CkIfatEATyyZiVqoRq6k64iuM9JkA3OzdXzMQexXVJ1TLZVEH0E7bhlY9d8O1ozR
oQs/FiZNAx2iijCWyv0lpjE73+kCgYEA9mZtyhkHkFDpwrSM1APaL8oNAbbjwEy7Z5Mqfql+lIp1
YkriL0DbLXlvRAH+yHPRit2hHOjtUNZh4Axv+cpg09qbUI3+43eEy24B7G/Uh+GTfbjsXsOxQx/x
```

```
p9otyVwc7hsQ5TA5PZb+mvkJ5OBEKzet9XcKwONBYELGhnEPe7cCgYEA06Vgov6YHleHui9kHuws
ayav0elc5zkxjF9nfHFJRry21R1trw2Vdpn+9g481URrpzWVOEihvm+xTtmaZlSp//lkq75XDwnU
WA8gkn6O3QE3fq2yN98BURsAKdJfJ5RL1HvGQvTe10HLYYXpJnEkHv+Unl2ajLivWUt5pbBrKbUC
gYBjbO+OZk0sCcpZ29sbzjYjpIddErySIyRX5gV2uNQwAjLdp9PfN295yQ+BxMBXiIycWVQiw0bH
oMo7yykABY7Ozd5wQewBQ4AdSlWSX4nGDtsiFxWiI5sKuAAeOCbTosy1s8w8fxoJ5Tz1sdoxNeGs
Arq6Wv/G16zQuAE9zK9vvwKBgF+09VI/1wJBirsDGz9whVWfFPrTkJNvJZzYt69qezxlsjgFKshy
WBhd4xHZtmCqpBPlAymEjr/TOlbxyARmXMnIOWIAnNXMGB4KGSyl1mzSVAoQ+fqR+cJ3d0dyPl1j
jjb0Ed/NY8frlNDxAVHE8BSkdsx2f6ELEyBKJSRr9snRAoGAMrTwYneXzvTskF/S5Fyu0iOegLDa
NWUH38v/nDCgEpIXD5Hn3qAEcju1IjmbwlvtW+nY2jVhv7UGd8MjwUTNGItdb6nsYqM2asrnF3qS
VRkAKKKYeGjkpUfVTrW0YFjXkfcrR/V+QFL5OndHAKJXjW7a4ejJLncTzmZSpYzwApc=
-----END RSA PRIVATE KEY-----
```

您的私有金鑰不會存放在 中，而且只能在建立時 AWS 擷取。您稍後無法進行復原。如果您遺失私有金鑰，則必須建立新的金鑰對。

如果您是從 Linux 電腦連接到執行個體，建議您使用下列命令來設定私有金鑰檔案的許可，即可確保只有您能夠讀取該檔案。

```
$ chmod 400 MyKeyPair.pem
```

顯示您的金鑰對

「指紋」是由您的金鑰對產生，您可以將其用於驗證本機擁有的私有金鑰與儲存在 AWS的公有金鑰相符。

指紋為取自私有金鑰的 DER 編碼副本的 SHA1 雜湊。此值會在建立金鑰對時擷取，並使用公 AWS 有金鑰存放在 中。您可以在 Amazon EC2 主控台或執行 AWS CLI 命令 來檢視指紋aws ec2 describe-key-pairs。

以下範例顯示 MyKeyPair 的指紋。

```
$ aws ec2 describe-key-pairs --key-name MyKeyPair
{
    "KeyPairs": [
        {
            "KeyName": "MyKeyPair",
            "KeyFingerprint":
 "1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f"
        }
    ]
}
```

如需金鑰和指紋的詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Amazon EC2 金鑰對。
Amazon EC2

刪除您的金鑰對

若要刪除金鑰對，請執行 aws ec2 delete-key-pair 命令，以您要刪除的金鑰對名稱來取代 *MyKeyPair*。

```
$ aws ec2 delete-key-pair --key-name MyKeyPair
```

參考

AWS CLI 參考：

- aws ec2
- aws ec2 create-key-pair
- aws ec2 delete-key-pair
- aws ec2 describe-key-pairs

其他參考：

- Amazon Elastic Compute Cloud 文件
- 若要檢視和貢獻 AWS SDK 和 AWS CLI 程式碼範例，請參閱 GitHub 上的AWS 程式碼範例儲存庫。

在 中建立、設定和刪除 Amazon EC2 安全群組 AWS CLI

您可以為您的 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體建立基本上作為防火牆執行的安全群組，並使用規則來判斷哪些網路流量可以進入和離開。

使用 AWS Command Line Interface (AWS CLI) 建立安全群組、將規則新增至現有安全群組，以及刪除安全群組。

> **Note**
>
> 如需其他命令範例，請參閱 AWS CLI 。

主題

- 先決條件
- 建立安全群組
- 新增規則至安全群組
- 刪除您的安全群組
- 參考

先決條件

若要執行 ec2 命令，您需要：

- 安裝及設定 AWS CLI。如需詳細資訊，請參閱the section called “安裝/更新”及的身分驗證和存取憑證 AWS CLI。
- 設定 IAM 許可，以允許 Amazon EC2 存取。如需 Amazon EC2 IAM 許可的詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Amazon EC2 的 IAM 政策。 Amazon EC2

建立安全群組

您可以建立與虛擬私有雲端 (VPC) 相關聯的安全性群組。

下列 aws ec2 create-security-group 範例顯示如何為指定的 VPC 建立安全群組。

```
$ aws ec2 create-security-group --group-name my-sg --description "My security group" --
vpc-id vpc-1a2b3c4d
{
    "GroupId": "sg-903004f8"
}
```

若要檢視安全群組的初始資訊，請執行 aws ec2 describe-security-groups 命令。您只能依 vpc-id 參考 EC2-VPC 安全群組，不能使用其名稱。

```
$ aws ec2 describe-security-groups --group-ids sg-903004f8
{
    "SecurityGroups": [
        {
            "IpPermissionsEgress": [
                {
```

```
                    "IpProtocol": "-1",
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "UserIdGroupPairs": []
                }
            ],
            "Description": "My security group"
            "IpPermissions": [],
            "GroupName": "my-sg",
            "VpcId": "vpc-1a2b3c4d",
            "OwnerId": "123456789012",
            "GroupId": "sg-903004f8"
        }
    ]
}
```

新增規則至安全群組

當您執行 Amazon EC2 執行個體，您必須啟用安全群組中的規則，以允許傳入網路流量來連線到映像。

例如，如果您正在啟動 Windows 執行個體，則通常需新增規則以允許 TCP 連接埠 3389 上的對內流量來支援遠端桌面通訊協定 (RDP)。如果您正在啟動 Linux 執行個體，則通常必須新增規則以允許 TCP 連接埠 22 上的對內流量來支援 SSH 連線。

使用 `aws ec2 authorize-security-group-ingress` 命令以新增規則至安全群組。此命令的一項必要參數為您電腦的公有 IP 地址或您電腦連接的網路 (採用地址範圍的形式)，使用 CIDR 符號。

> **Note**
>
> 我們提供下列服務 https：//https://checkip.global.api.aws/。若要尋找可協助您識別 IP 地址的其他服務，請使用瀏覽器來搜尋「我的 IP 地址是什麼」。如果您透過 ISP 或從防火牆後方使用動態 IP 地址來連線 (透過私有網路的 NAT 閘道)，則您的地址可能會定期變更。在這種情況下，您必須找出用戶端電腦所用 IP 地址的範圍。

以下範例顯示如何新增 RDP 的規則 (TCP 連接埠 3389) 到使用您的 IP 地址、ID 為 `sg-903004f8` 的 EC2-VPC 安全群組。

若要開始，請找出 IP 地址。

```
$ curl https://checkip.amazonaws.com
x.x.x.x
```

然後您就可以透過執行 aws ec2 authorize-security-group-ingress 命令來將 IP 地址新增至安全群組。

```
$ aws ec2 authorize-security-group-ingress --group-id sg-903004f8 --protocol tcp --port
 3389 --cidr x.x.x.x/x
```

以下命令會新增另一個規則，以啟用 SSH 連線到相同安全群組中的執行個體。

```
$ aws ec2 authorize-security-group-ingress --group-id sg-903004f8 --protocol tcp --port
 22 --cidr x.x.x.x/x
```

若要檢視安全群組的任何變更，請執行 aws ec2 describe-security-groups 命令。

```
$ aws ec2 describe-security-groups --group-ids sg-903004f8
{
    "SecurityGroups": [
        {
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "UserIdGroupPairs": []
                }
            ],
            "Description": "My security group"
            "IpPermissions": [
                {
                    "ToPort": 22,
                    "IpProtocol": "tcp",
                    "IpRanges": [
                        {
                            "CidrIp": "x.x.x.x/x"
```

```
                    }
                ]
                "UserIdGroupPairs": [],
                "FromPort": 22
            }
        ],
        "GroupName": "my-sg",
        "OwnerId": "123456789012",
        "GroupId": "sg-903004f8"
    }
  ]
}
```

刪除您的安全群組

若要刪除安全群組，請執行 aws ec2 delete-security-group 命令。

> **Note**
>
> 您不能刪除目前已經連接到環境的安全群組。

下列命令範例會刪除 EC2-VPC 安全群組。

```
$ aws ec2 delete-security-group --group-id sg-903004f8
```

參考

AWS CLI 參考：

- aws ec2
- aws ec2 authorize-security-group-ingress
- aws ec2 create-security-group
- aws ec2 delete-security-group
- aws ec2 describe-security-groups

其他參考：

- Amazon Elastic Compute Cloud 文件

- 若要檢視和貢獻 AWS SDK 和 AWS CLI 程式碼範例，請參閱 GitHub 上的AWS 程式碼範例儲存庫。

在 中啟動、列出和刪除 Amazon EC2 執行個體 AWS CLI

您可以使用 AWS Command Line Interface (AWS CLI) 來啟動、列出和刪除 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體。如果您啟動不在 AWS 免費方案內的執行個體，則會在您啟動執行個體之後向您收費，並按執行個體執行的時間收費，即使執行個體保持閒置狀態。

> **Note**
>
> 如需其他命令範例，請參閱AWS CLI 參考指南。

主題

- 先決條件
- 啟動您的執行個體
- 新增區塊型儲存設備至您的執行個體
- 新增標籤至您的執行個體
- 連線到您的執行個體
- 列出您的執行個體
- 刪除您的執行個體
- 參考

先決條件

若要在此主題中執行 ec2 命令，您需要：

- 安裝及設定 AWS CLI。如需詳細資訊，請參閱the section called “安裝/更新”及的身分驗證和存取憑證 AWS CLI。
- 設定 IAM 許可，以允許 Amazon EC2 存取。如需 Amazon EC2 IAM 許可的詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Amazon EC2 的 IAM 政策。 Amazon EC2
- 建立金鑰對和安全群組。
- 選取 Amazon Machine Image (AMI) 並記下 AMI ID。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的尋找合適的 AMI。

啟動您的執行個體

若要使用所選的 AMI 來啟動 Amazon EC2 執行個體，請使用 aws ec2 run-instances 命令。您可於虛擬私有雲端 (VPC) 中啟動執行個體。

一開始您的執行個體會處於 pending 狀態，但幾分鐘後就會變更為 running 狀態。

以下範例說明如何在 VPC 的指定子網路中啟動 t2.micro 執行個體。以自訂值取代##參數值。

```
$ aws ec2 run-instances --image-id ami-xxxxxxxx --count 1 --instance-type t2.micro --
key-name MyKeyPair --security-group-ids sg-903004f8 --subnet-id subnet-6e7f829e
{
    "OwnerId": "123456789012",
    "ReservationId": "r-5875ca20",
    "Groups": [
        {
            "GroupName": "my-sg",
            "GroupId": "sg-903004f8"
        }
    ],
    "Instances": [
        {
            "Monitoring": {
                "State": "disabled"
            },
            "PublicDnsName": null,
            "Platform": "windows",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "EbsOptimized": false,
            "LaunchTime": "2013-07-19T02:42:39.000Z",
            "PrivateIpAddress": "10.0.1.114",
            "ProductCodes": [],
            "VpcId": "vpc-1a2b3c4d",
            "InstanceId": "i-5203422c",
            "ImageId": "ami-173d747e",
            "PrivateDnsName": "ip-10-0-1-114.ec2.internal",
            "KeyName": "MyKeyPair",
            "SecurityGroups": [
                {
                    "GroupName": "my-sg",
                    "GroupId": "sg-903004f8"
```

```
            }
        ],
        "ClientToken": null,
        "SubnetId": "subnet-6e7f829e",
        "InstanceType": "t2.micro",
        "NetworkInterfaces": [
            {
                "Status": "in-use",
                "SourceDestCheck": true,
                "VpcId": "vpc-1a2b3c4d",
                "Description": "Primary network interface",
                "NetworkInterfaceId": "eni-a7edb1c9",
                "PrivateIpAddresses": [
                    {
                        "PrivateDnsName": "ip-10-0-1-114.ec2.internal",
                        "Primary": true,
                        "PrivateIpAddress": "10.0.1.114"
                    }
                ],
                "PrivateDnsName": "ip-10-0-1-114.ec2.internal",
                "Attachment": {
                    "Status": "attached",
                    "DeviceIndex": 0,
                    "DeleteOnTermination": true,
                    "AttachmentId": "eni-attach-52193138",
                    "AttachTime": "2013-07-19T02:42:39.000Z"
                },
                "Groups": [
                    {
                        "GroupName": "my-sg",
                        "GroupId": "sg-903004f8"
                    }
                ],
                "SubnetId": "subnet-6e7f829e",
                "OwnerId": "123456789012",
                "PrivateIpAddress": "10.0.1.114"
            }
        ],
        "SourceDestCheck": true,
        "Placement": {
            "Tenancy": "default",
            "GroupName": null,
            "AvailabilityZone": "us-west-2b"
        },
```

```
            "Hypervisor": "xen",
            "BlockDeviceMappings": [
                {
                    "DeviceName": "/dev/sda1",
                    "Ebs": {
                        "Status": "attached",
                        "DeleteOnTermination": true,
                        "VolumeId": "vol-877166c8",
                        "AttachTime": "2013-07-19T02:42:39.000Z"
                    }
                }
            ],
            "Architecture": "x86_64",
            "StateReason": {
                "Message": "pending",
                "Code": "pending"
            },
            "RootDeviceName": "/dev/sda1",
            "VirtualizationType": "hvm",
            "RootDeviceType": "ebs",
            "Tags": [
                {
                    "Value": "MyInstance",
                    "Key": "Name"
                }
            ],
            "AmiLaunchIndex": 0
        }
    ]
}
```

新增區塊型儲存設備至您的執行個體

您啟動的每個執行個體均有一個相關聯的根設備磁碟區。您可以使用區塊型設備映射，來指定執行個體啟動時要連接的其他 Amazon Elastic Block Store (Amazon EBS) 磁碟區或執行個體存放磁碟區。

若要新增區塊型儲存設備至執行個體，請在使用 `run-instances` 時指定 `--block-device-mappings` 選項。

以下範例參數佈建 20 GB 大小的標準 Amazon EBS 磁碟區，並使用識別符 `/dev/sdf` 將其映射到您的執行個體。

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdf\",\"Ebs\":{\"VolumeSize\":20,
\"DeleteOnTermination\":false}}]"
```

下列範例根據現有快照，新增映射到 /dev/sdf 的 Amazon EBS 磁碟區。快照代表載入到磁碟區的映像。當您指定快照時，不必指定磁碟區大小，它夠大到可以保存你的映像。不過，如果您指定大小，則必須等於或大於快照的大小。

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdf\",\"Ebs\":{\"SnapshotId\":\"snap-
a1b2c3d4\"}}]"
```

下列範例新增兩個磁碟區到執行個體。可供執行個體使用的磁碟區數量，根據其執行個體類型而定。

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdf\",\"VirtualName\":\"ephemeral0\"},
{\"DeviceName\":\"/dev/sdg\",\"VirtualName\":\"ephemeral1\"}]"
```

以下範例會建立對應 (/dev/sdj)，但不會為執行個體佈建磁碟區。

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdj\",\"NoDevice\":\"\"}]"
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的封鎖裝置映射。

新增標籤至您的執行個體

標籤是您指派給 AWS 資源的標籤。它可讓您新增中繼資料到您的資源，以供您用於各種用途。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的標記您的 資源。

以下範例顯示說明如何使用 aws ec2 create-tags 命令，將金鑰名稱為「Name」和金鑰值為「MyInstance」的標籤新增至指定的執行個體。

```
$ aws ec2 create-tags --resources i-5203422c --tags Key=Name,Value=MyInstance
```

連線到您的執行個體

執行個體正在執行時，您可以與其連結並加以使用，如同操作面前的電腦一般。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的連線至您的 Amazon EC2 執行個體。 Amazon EC2

列出您的執行個體

您可以使用 AWS CLI 列出您的執行個體，並檢視其相關資訊。您可以列出所有執行個體，或根據感興趣的執行個體來篩選條件結果。

下列範例顯示如何使用 aws ec2 describe-instances 命令。

下列命令能列出您所有的執行個體。

```
$ aws ec2 describe-instances
```

以下命令會將清單篩選為僅限您的 t2.micro 執行個體，並僅輸出每個相符項目的 InstanceId 值。

```
$ aws ec2 describe-instances --filters "Name=instance-type,Values=t2.micro" --query
 "Reservations[].Instances[].InstanceId"
[
    "i-05e998023d9c69f9a"
]
```

以下命令會列出具有標籤 Name=MyInstance 的任何執行個體。

```
$ aws ec2 describe-instances --filters "Name=tag:Name,Values=MyInstance"
```

下列命令列出使用下列任何 AMI 啟動的執行個體：ami-x0123456、ami-y0123456 和 ami-z0123456。

```
$ aws ec2 describe-instances --filters "Name=image-id,Values=ami-x0123456,ami-
y0123456,ami-z0123456"
```

刪除您的執行個體

終止執行個體會將其刪除。終止執行個體後，您將無法重新連接至該執行個體。

執行個體的狀態變更為 shutting-down 或 terminated 時刻起，該執行個體便停止收取費用。如果您希望稍後重新連接執行個體，請使用 stop-instances 而不是 terminate-instances。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的終止您的執行個體。

若要刪除執行個體，請使用命令 aws ec2 terminate-instances 將其刪除。

```
$ aws ec2 terminate-instances --instance-ids i-5203422c
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-5203422c",
```

```
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

參考

AWS CLI 參考：

- aws ec2
- aws ec2 create-tags
- aws ec2 describe-instances
- aws ec2 run-instances
- aws ec2 terminate-instances

其他參考：

- Amazon Elastic Compute Cloud 文件
- 若要檢視和貢獻 AWS SDK 和 AWS CLI 程式碼範例，請參閱 GitHub 上的AWS 程式碼範例儲存庫。

在 中使用 bash 指令碼變更 Amazon EC2 執行個體類型 AWS CLI

此 Amazon EC2 的 bash 指令碼範例會使用 AWS Command Line Interface () 變更 Amazon EC2 執行個體的執行個體類型AWS CLI。如果執行個體正在執行，它會停止執行個體，變更執行個體類型，然後在要求時重新啟動執行個體。Shell 指令碼是專為在命令列界面中執行而設計的程式。

> **Note**
>
> 如需其他命令範例，請參閱 AWS CLI 。

主題

- 開始之前
- 關於此範例
- 參數
- 檔案
- 參考

開始之前

在可以執行下列任何範例之前，您必須先完成下列事項。

- 安裝及設定 AWS CLI。如需詳細資訊，請參閱the section called “安裝/更新”及的身分驗證和存取憑證 AWS CLI。
- 您使用的設定檔必須具有許可，以允許範例執行 AWS 的操作。
- 您有許可能夠停止和修改帳戶中正在執行的 Amazon EC2 執行個體。如果您執行測試指令碼，它會為您啟動執行個體，測試變更類型，然後終止執行個體。
- AWS 最佳實務是授予此程式碼的最低權限，或僅授予執行任務所需的許可。如需詳細資訊，請參閱《AWS Identity and Access Management (IAM) 使用者指南》中的授予最低權限。
- 此程式碼尚未在所有 AWS 區域中進行測試。部分 AWS 服務僅適用於特定 區域。如需詳細資訊，請參閱《AWS 一般參考指南》中的服務端點和配額。
- 執行此程式碼可能會對您的帳戶產生費用 AWS 。您有責任確保在使用完該指令碼建立的所有資源後將這些資源移除。

關於此範例

這個範例被寫成 Shell 指令碼檔案 `change_ec2_instance_type.sh` 中的一個函數，您可以從另一個指令碼或從命令列中 `source`。每個指令碼檔案包含描述每個函數的註釋。待函數儲存到記憶體中，您就可以從命令列呼叫它。例如，下列命令會將指定執行個體的類型變更為 `t2.nano`：

```
$ source ./change_ec2_instance_type.sh
$ ./change_ec2_instance_type -i *instance-id* -t new-type
```

如需完整範例和可下載的指令碼檔案，請參閱 GitHub 上 AWS 程式碼範例儲存庫中的變更 Amazon EC2 執行個體類型。

參數

-i – (字串) 指定要修改的執行個體 ID。

-t – (字串) 指定要切換的 Amazon EC2 執行個體類型。

-r – (切換) 依預設，此為取消設定。如果設定為 `-r`，則在類型切換之後重新啟動執行個體。

-f – (切換) 依預設，指令碼會在進行切換之前提示使用者確認關閉執行個體。如果設定為 `-f`，函數不會在關閉執行個體進行類型切換之前提示使用者

-v – (切換) 依預設，指令碼會以無提示的方式操作，只會在發生錯誤時顯示輸出。如果設定為 `-v`，函數會在整個操作期間顯示狀態。

檔案

`change_ec2_instance_type.sh`

主指令碼檔案包含執行下列任務的 `change_ec2_instance_type()` 函數：

- 驗證指定的 Amazon EC2 執行個體是否存在。
- 除非選取 `-f`，否則會在停止執行個體之前警告使用者。
- 變更執行個體類型
- 如果設定了 `-r`，重新啟動執行個體並確認執行個體正在執行

檢視 GitHub 上的 change_ec2_instance_type.sh。

`test_change_ec2_instance_type.sh`

檔案 `test_change_ec2_instance_type.sh` 指令碼會為 `change_ec2_instance_type` 函數測試各種程式碼路徑。如果測試指令碼中的所有步驟都能正常運作，測試指令碼會移除它所建立的所有資源。

您可以使用下列參數來執行測試指令碼：

- -v – (切換) 每個測試都會在執行時顯示通過/失敗狀態。依預設，測試會以無提示的方式執行，且輸出只包含最終的整體通過/失敗狀態。
- -i – (切換) 指令碼會在每次測試後暫停，讓您能夠瀏覽每個步驟的中繼結果。可讓您使用 Amazon EC2 主控台檢查執行個體目前的狀態。在出現提示時按下 ENTER 後，指令碼繼續執行下一個步驟。

檢視 GitHub 上的 test_change_ec2_instance_type.sh。

awsdocs_general.sh

指令碼檔案 awsdocs_general.sh 會保存在各個 AWS CLI進階範例中使用的一般用途函數。

檢視 GitHub 上的 awsdocs_general.sh。

參考

AWS CLI 參考：

- aws ec2
- aws ec2 describe-instances
- aws ec2 modify-instance-attribute
- aws ec2 start-instances
- aws ec2 stop-instances
- aws ec2 wait instance-running
- aws ec2 wait instance-stopped

其他參考：

- Amazon Elastic Compute Cloud 文件
- 若要檢視和貢獻 AWS SDK 和 AWS CLI 程式碼範例，請參閱 GitHub 上的AWS 程式碼範例儲存庫。

在 中使用 Amazon S3 Glacier AWS CLI

Amazon S3 Glacier 簡介
Amazon S3 Glacier 簡介

本主題顯示執行 S3 Glacier 常見任務的 AWS CLI 命令範例。這些範例示範如何使用 AWS CLI 將大型檔案分割成較小的部分並從命令列上傳，以將其上傳至 S3 Glacier。

您可以使用 AWS Command Line Interface () 存取 Amazon S3 Glacier 功能AWS CLI。若要列出 S3 Glacier 的 AWS CLI 命令，請使用下列命令。

```
aws glacier help
```

> **Note**
>
> 如需命令參考和其他範例，請參閱 AWS CLI 命令參考中的 `aws glacier`。

主題

- 先決條件
- 建立 Amazon S3 Glacier 保存庫
- 準備上傳檔案
- 啟動分段上傳和上載檔案
- 完成上傳
- 資源

先決條件

若要執行 `glacier` 命令，您需要：

- 安裝及設定 AWS CLI。如需詳細資訊，請參閱the section called “安裝/更新”及的身分驗證和存取憑證 AWS CLI。
- 本教學課程使用數個命令列工具，通常預先安裝在類 Unix 的操作系統上，包括 Linux 和 macOS。Windows 使用者可以通過安裝 Cygwin 並從 Cygwin 終端機執行命令來使用相同的工具。我們會備註在可用的情況下執行相同功能的 Windows 原生命令和公用程式。

建立 Amazon S3 Glacier 保存庫

使用 `create-vault` 命令建立一個保存庫。

```
$ aws glacier create-vault --account-id - --vault-name myvault
{
    "location": "/123456789012/vaults/myvault"
}
```

> **Note**
>
> 所有 S3 Glacier 命令都需要一個帳戶 ID 參數。使用連字號字元 (`--account-id -`) 來使用目前的帳戶。

準備上傳檔案

為測試上載建立一個檔案。以下命令將建立名為 *largefile* 的檔案，其中包含 3 MiB 的隨機數據。

Linux 或 macOS

```
$ dd if=/dev/urandom of=largefile bs=3145728 count=1
1+0 records in
1+0 records out
3145728 bytes (3.1 MB) copied, 0.205813 s, 15.3 MB/s
```

dd 是一個實用程式，可將輸入檔案中的大量位元組複製到輸出檔案。上述範例使用系統裝置檔案 `/dev/urandom` 作為隨機資料的來源。`fsutil` 在 Windows 執行類似的功能。

Windows

```
C:\> fsutil file createnew largefile 3145728
File C:\temp\largefile is created
```

接著，使用檔案分割器將檔案分割成 1 MiB (1，048，576 位元組） 區塊。

```
$ split -b 1048576 --verbose largefile chunk
creating file `chunkaa'
creating file `chunkab'
creating file `chunkac'
```

啟動分段上傳和上載檔案

使用 initiate-multipart-upload 命令，在 Amazon S3 Glacier 中建立分段上傳。

```
$ aws glacier initiate-multipart-upload --account-id - --archive-description "multipart
 upload test" --part-size 1048576 --vault-name myvault
{
    "uploadId": "19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ",
```

```
    "location": "/123456789012/vaults/myvault/multipart-
uploads/19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ"
}
```

S3 Glacier 要求每個部分的大小以位元組為單位 (本範例中為 1 MiB)、您的保存庫名稱和帳戶 ID，以便設定分段上傳。操作完成時， 會 AWS CLI 輸出上傳 ID。將上傳 ID 儲存到 Shell 變數以供稍後使用。

Linux 或 macOS

```
$ UPLOADID="19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ"
```

Windows

```
C:\> set UPLOADID="19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ"
```

接下來，使用 upload-multipart-part 命令來上傳這三個部分。

```
$ aws glacier upload-multipart-part --upload-id $UPLOADID --body chunkaa --range 'bytes
 0-1048575/*' --account-id - --vault-name myvault
{
    "checksum": "e1f2a7cd6e047fa606fe2f0280350f69b9f8cfa602097a9a026360a7edc1f553"
}
$ aws glacier upload-multipart-part --upload-id $UPLOADID --body chunkab --range 'bytes
 1048576-2097151/*' --account-id - --vault-name myvault
{
    "checksum": "e1f2a7cd6e047fa606fe2f0280350f69b9f8cfa602097a9a026360a7edc1f553"
}
$ aws glacier upload-multipart-part --upload-id $UPLOADID --body chunkac --range 'bytes
 2097152-3145727/*' --account-id - --vault-name myvault
{
    "checksum": "e1f2a7cd6e047fa606fe2f0280350f69b9f8cfa602097a9a026360a7edc1f553"
}
```

> **Note**
>
> 上面的範例使用貨幣符號 ($) 來參考 Linux 上的 UPLOADID Shell 變數內容。在 Windows 命令列，請在變數名稱 (例如 %UPLOADID%) 任一側使用百分比符號 (%)。

您必須在上傳時指定每個部分的位元組範圍，以便可以透過 S3 Glacier 按照正確的順序重新組合。每一部分是 1048576 bytes，所以第一部分的位元組是 0-1048575，第二部分是 1048576-2097151，第三部分是 2097152-3145727。

完成上傳

Amazon S3 Glacier 需要原始檔案的樹雜湊，以確認所有上傳的片段都完 AWS 好無損。

若要計算樹雜湊值，您必須將檔案拆分為 1 MiB 部分，並計算每個部分的二進位 SHA-256 雜湊值。然後，將雜湊值清單分為幾對，在每一對中合併兩個二進位雜湊值，並取結果的雜湊值。重複此過程，直到只剩下一個雜湊值。如果在任何階層都有奇數的雜湊值，則將其推進到下一階層而不進行修改。

使用命令列公用程式正確計算樹雜湊函數的關鍵是將每個雜湊值儲存為二進位格式，並且只在最後一步轉換為十六進位。組合或雜湊樹中任何雜湊的十六進位版本將導致錯誤的結果。

> **Note**
>
> Windows 使用者可以使用 `type` 命令代替`cat`。位於 OpenSSL.org的 OpenSSL 適用於 Windows。

計算樹雜湊函數

1. 如果尚未這麼做，請將原始檔案拆分為 1 MiB 部分。

```
$ split --bytes=1048576 --verbose largefile chunk
creating file `chunkaa'
creating file `chunkab'
creating file `chunkac'
```

2. 計算並存放每個區塊的二進位 SHA-256 雜湊。

```
$ openssl dgst -sha256 -binary chunkaa > hash1
$ openssl dgst -sha256 -binary chunkab > hash2
$ openssl dgst -sha256 -binary chunkac > hash3
```

3. 合併前兩個雜湊，並取結果的二進位雜湊值。

```
$ cat hash1 hash2 > hash12
$ openssl dgst -sha256 -binary hash12 > hash12hash
```

4. 以與區塊 ac 的雜湊值和雜湊結果將區塊 aa 和 ab 的父系雜湊值作結合，此時會輸出十六進位。將結果存放在 Shell 變數中。

```
$ cat hash12hash hash3 > hash123
$ openssl dgst -sha256 hash123
SHA256(hash123)= 9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67
$ TREEHASH=9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67
```

最後，用 complete-multipart-upload 命令完成上傳。此命令採用原始檔案的大小 (以位元組為單位)、最終的樹雜湊值 (十六進位) 以及您的帳戶 ID 和保存庫名稱。

```
$ aws glacier complete-multipart-upload --checksum $TREEHASH --archive-size 3145728 --
upload-id $UPLOADID --account-id - --vault-name myvault
{
    "archiveId": "d3AbWhE0YE1m6f_fI1jPG82F8xzbMEEZmrAlLGAAONJAzo5QdP-
N83MKqd96Unspoa5H5lItWX-sK8-QS0ZhwsyGiu9-R-
kwWUyS1dSBlmgPPWkEbeFfqDSav053rU7FvVLHfRc6hg",
    "checksum": "9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
    "location": "/123456789012/vaults/myvault/archives/
d3AbWhE0YE1m6f_fI1jPG82F8xzbMEEZmrAlLGAAONJAzo5QdP-N83MKqd96Unspoa5H5lItWX-sK8-
QS0ZhwsyGiu9-R-kwWUyS1dSBlmgPPWkEbeFfqDSav053rU7FvVLHfRc6hg"
}
```

您還可以使用 describe-vault 命令來檢查保存庫的狀態。

```
$ aws glacier describe-vault --account-id - --vault-name myvault
{
    "SizeInBytes": 3178496,
    "VaultARN": "arn:aws:glacier:us-west-2:123456789012:vaults/myvault",
    "LastInventoryDate": "2018-12-07T00:26:19.028Z",
    "NumberOfArchives": 1,
    "CreationDate": "2018-12-06T21:23:45.708Z",
    "VaultName": "myvault"
}
```

> **Note**
>
> 每天更新一次保存庫狀態。請參閱使用保存庫了解更多資訊。

現在可以安全地移除您先前建立的區塊和雜湊檔案。

```
$ rm chunk* hash*
```

如需分段上傳的詳細資訊，請參閱《Amazon S3 Glacier 開發人員指南》中的分段上傳大型封存和運算檢查總和。

資源

AWS CLI 參考：

- aws glacier
- aws glacier complete-multipart-upload
- aws glacier create-vault
- aws glacier describe-vault
- aws glacier initiate-multipart-upload

服務參考：

- Amazon S3 Glacier 開發人員指南
- Amazon S3 Glacier 開發人員指南中的以部分形式上傳大型封存
- Amazon S3 Glacier 開發人員指南中的運算檢查總和
- Amazon S3 Glacier 開發人員指南中的使用文件庫

在 中使用 IAM AWS CLI

簡介 AWS Identity and Access Management
簡介 AWS Identity and Access Management

您可以使用 AWS Identity and Access Management () 存取 AWS Command Line Interface (IAM) 的功能AWS CLI。若要列出 IAM 的 AWS CLI 命令，請使用下列命令。

```
aws iam help
```

本主題顯示執行 IAM 常見任務的 AWS CLI 命令範例。

在執行任何命令前，請先設定您的預設憑證。如需詳細資訊，請參閱 設定 的設定 AWS CLI。

如需 IAM 服務的詳細資訊，請參閱《AWS Identity and Access Management 使用者指南》。

主題

- 建立 IAM 使用者與群組
- 將 IAM 受管政策連接至使用者
- 為 IAM 使用者設定初始密碼
- 為 IAM 使用者建立存取金鑰

建立 IAM 使用者與群組

建立群組並在其中新增使用者

1. 使用 `create-group` 命令來建立群組。

```
$ aws iam create-group --group-name MyIamGroup
{
    "Group": {
        "GroupName": "MyIamGroup",
        "CreateDate": "2018-12-14T03:03:52.834Z",
        "GroupId": "AGPAJNUJ2W4IJVEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:group/MyIamGroup",
        "Path": "/"
    }
}
```

2. 使用 `create-user` 命令來建立使用者。

```
$ aws iam create-user --user-name MyUser
{
    "User": {
        "UserName": "MyUser",
        "Path": "/",
        "CreateDate": "2018-12-14T03:13:02.581Z",
        "UserId": "AIDAJY2PE5XUZ4EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:user/MyUser"
    }
```

```
}
```

3. 使用 add-user-to-group 命令將使用者新增到該群組。

```
$ aws iam add-user-to-group --user-name MyUser --group-name MyIamGroup
```

4. 如果要確認 MyIamGroup 群組是否包含 MyUser，請使用 get-group 命令。

```
$ aws iam get-group --group-name MyIamGroup
{
    "Group": {
        "GroupName": "MyIamGroup",
        "CreateDate": "2018-12-14T03:03:52Z",
        "GroupId": "AGPAJNUJ2W4IJVEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:group/MyIamGroup",
        "Path": "/"
    },
    "Users": [
        {
            "UserName": "MyUser",
            "Path": "/",
            "CreateDate": "2018-12-14T03:13:02Z",
            "UserId": "AIDAJY2PE5XUZ4EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/MyUser"
        }
    ],
    "IsTruncated": "false"
}
```

將 IAM 受管政策連接至使用者

此範例中的政策可提供使用者「進階使用者存取」。

將 IAM 受管政策連接至使用者

1. 決定要連接到政策的 Amazon 資源名稱 (ARN)。以下命令使用 list-policies 來尋找名稱為 PowerUserAccess 之政策的 ARN。接著，將 ARN 存放在環境變數中。

```
$ export POLICYARN=$(aws iam list-policies --query 'Policies[?
PolicyName==`PowerUserAccess`].{ARN:Arn}' --output text)       ~
$ echo $POLICYARN
```

```
arn:aws:iam::aws:policy/PowerUserAccess
```

2. 若要連接政策，請使用 attach-user-policy 命令並參考保留政策 ARN 的環境變數。

```
$ aws iam attach-user-policy --user-name MyUser --policy-arn $POLICYARN
```

3. 執行 list-attached-user-policies 命令，確認該政策已連接至使用者。

```
$ aws iam list-attached-user-policies --user-name MyUser
{
    "AttachedPolicies": [
        {
            "PolicyName": "PowerUserAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/PowerUserAccess"
        }
    ]
}
```

如需詳細資訊，請參閱存取管理資源。此主題提供許可與政策概觀的連結，以及用於存取 Amazon S3、Amazon EC2 和其他服務的政策範例連結。

為 IAM 使用者設定初始密碼

以下命令使用 create-login-profile 來為指定的使用者設定初始密碼。當使用者首次登入時，使用者需要將密碼變更為只有該使用者知道的內容。

```
$ aws iam create-login-profile --user-name MyUser --password My!User1Login8P@ssword --
password-reset-required
{
    "LoginProfile": {
        "UserName": "MyUser",
        "CreateDate": "2018-12-14T17:27:18Z",
        "PasswordResetRequired": true
    }
}
```

您可使用 update-login-profile 命令來變更使用者的密碼。

```
$ aws iam update-login-profile --user-name MyUser --password My!User1ADifferentP@ssword
```

為 IAM 使用者建立存取金鑰

您可以使用 create-access-key 命令來為使用者建立存取金鑰。存取金鑰是一組安全憑證，包含存取金鑰 ID 和私密金鑰。

一位使用者一次只能建立兩個存取金鑰。若您嘗試建立第三組，命令會傳回 LimitExceeded 錯誤。

```
$ aws iam create-access-key --user-name MyUser
{
    "AccessKey": {
        "UserName": "MyUser",
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "Status": "Active",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "CreateDate": "2018-12-14T17:34:16Z"
    }
}
```

使用 delete-access-key 命令來為使用者刪除存取金鑰。使用存取金鑰 ID 來指定要刪除哪一個存取金鑰。

```
$ aws iam delete-access-key --user-name MyUser --access-key-id AKIAIOSFODNN7EXAMPLE
```

在 中使用 Amazon S3 AWS CLI

Amazon Simple Storage Service (Amazon S3) 簡介

Amazon Simple Storage Service (Amazon S3) 簡介 – AWS雲端儲存

您可以使用 AWS Command Line Interface () 存取 Amazon Simple Storage Service (Amazon S3) 的功能AWS CLI。Amazon S3 是一種高度可擴展且耐用的物件儲存服務。Amazon S3 旨在提供幾乎無限制的儲存容量，使其成為滿足各種資料儲存和管理需求的理想解決方案。

Amazon S3 可讓您以物件形式存放和擷取任意數量的資料，從小型檔案到大型資料集。每個物件都存放在稱為儲存貯體的容器中，可透過 AWS Management Console 或以程式設計方式透過 AWS SDKs、工具和 存取和管理 AWS CLI。

Amazon S3 也提供基本儲存，包括生命週期管理、版本控制、可擴展性和安全性。這些與其他 AWS 服務 整合，可讓您建置雲端型解決方案來擴展您的需求。

AWS CLI 提供兩種存取 Amazon S3 的命令層：

- s3 – 專門為 發出的自訂高階命令 AWS CLI ，可簡化常見的任務，例如建立、操作、刪除和同步物件和儲存貯體。
- s3api— 公開對所有 Amazon S3 API 操作的直接存取，使您能夠執行進階操作。

本指南的主題：

- 在 中使用高階 (s3) 命令 AWS CLI
- 在 中使用 API 層級 (s3api) 命令 AWS CLI
- 中的 Amazon S3 儲存貯體生命週期指令碼範例 AWS CLI

在 中使用高階 (s3) 命令 AWS CLI

本主題說明如何使用 AWS CLI中的 `aws s3` 命令來管理 Amazon S3 儲存貯體和物件。有關本主題中未涉及的命令和其他命令範例，請參閱 AWS CLI 參考中的 `aws s3` 命令。

高階 `aws s3` 命令可簡化 Amazon S3 物件的管理作業。這些命令可讓您管理 Amazon S3 內部的內容以及本機目錄的內容。

主題

- 先決條件
- 開始之前
- 建立 儲存貯體
- 列出儲存貯體和物件
- 刪除儲存貯體
- 刪除物件
- 移動物件
- 複製物件
- 同步物件
- s3 命令的常用選項
- 資源

先決條件

若要執行 s3 命令，您需要：

- 安裝及設定 AWS CLI。如需詳細資訊，請參閱the section called “安裝/更新”及的身分驗證和存取憑證 AWS CLI。
- 您使用的設定檔必須具有許可，以允許範例執行 AWS 的操作。
- 了解這些 Amazon S3 術語：
  - 儲存貯體 – 頂層 Amazon S3 資料夾。
  - 前綴 – 儲存貯體中的 Amazon S3 資料夾。
  - 物件 – 託管於 Amazon S3 儲存貯體中的任一個項目。

開始之前

本節說明在使用 aws s3 命令之前應注意的幾個事項。

大型物件上傳

當您使用 aws s3 命令將大型物件上傳至 Amazon S3 儲存貯體時， AWS CLI 會自動執行分段上傳。使用這些 aws s3 命令時，您無法恢復失敗的上傳。

如果分段上傳因逾時而失敗，或者如果您在 中手動取消 AWS CLI，則 會 AWS CLI 停止上傳並清除任何已建立的檔案。此程序需要幾分鐘的時間。

如果分段上傳或清理程序因為 Kill 命令或系統故障而取消，則建立的檔案會保留在 Amazon S3 儲存貯體中。若要清理分段上傳，請使用 s3api abort-multipart-upload 命令。

分段副本中的檔案屬性和標籤

當您使用 aws s3 命名空間中的 AWS CLI 第 1 版命令，將檔案從一個 Amazon S3 儲存貯體位置複製到另一個 Amazon S3 儲存貯體位置，且該操作使用分段複製時，不會將來源物件中的檔案屬性複製到目的地物件。

根據預設，執行分段複製的s3命名空間中的第 2 content-disposition AWS CLI 版命令會將所有標籤和下列屬性集從來源傳輸到目的地複本：content-type、content-language、content-encoding、cache-control、、 expires和 metadata。

如果您使用第 1 AWS CLI 版，這可能會導致對 Amazon S3 端點進行額外的 AWS API 呼叫。這些可能包括：HeadObject、GetObjectTagging、和 PutObjectTagging。

如果您需要變更第 2 AWS CLI 版命令中的此預設行為，請使用 `--copy-props` 參數指定下列其中一個選項：

- 預設 – 預設值。指定副本包括連接至來源物件的所有標籤，以及用於非多段副本的 `--metadata-directive` 參數所包含的屬性：`content-type`、`content-language`、`content-encoding`、`content-disposition`、`cache-control`、`expires` 和 `metadata`。
- metadata-directive – 指定副本只包含用於非分段副本的 `--metadata-directive` 參數所包含的屬性。它不會複製任何標籤。
- none – 指定副本不包含來源物件的任何屬性。

建立 儲存貯體

使用 `s3 mb` 命令來建立儲存貯體。儲存貯體名稱必須是全域唯一 (在所有 Amazon S3 中都為唯一)，且應符合 DNS 標準。

儲存貯體名稱可包含小寫字母、數字、連字號和句號。儲存貯體名稱的開頭和結尾只能使用字母或數字，連字號或句號旁邊不能使用句號。

語法

```
$ aws s3 mb <target> [--options]
```

s3 mb 範例

下列範例會建立 `s3://amzn-s3-demo-bucket` 儲存貯體。

```
$ aws s3 mb s3://amzn-s3-demo-bucket
```

列出儲存貯體和物件

若要列出儲存貯體、資料夾或物件，請使用 `s3 ls` 命令。使用沒有目標或選項的命令會列出所有儲存貯體。

語法

```
$ aws s3 ls <target> [--options]
```

如需搭配此命令使用的幾個常見選項和範例，請參閱 s3 命令的常用選項。如需可用選項的完整清單，請參閱 AWS CLI 命令參考中的 `s3 ls`。

s3 ls 範例

以下範例列出您的所有 Amazon S3 儲存貯體。

```
$ aws s3 ls
2018-12-11 17:08:50 amzn-s3-demo-bucket1
2018-12-14 14:55:44 amzn-s3-demo-bucket2
```

下列命令列出儲存貯體中的所有物件和前綴。在此範例輸出中，前綴 example/ 有一個名為 MyFile1.txt 的檔案。

```
$ aws s3 ls s3://amzn-s3-demo-bucket
                           PRE example/
2018-12-04 19:05:48          3 MyFile1.txt
```

您可以將特定的前綴包含在命令中，將輸出篩選為該前綴。下列命令列出 *bucket-name/example/* 中的物件 (也就是使用前綴 *example/* 篩選出 *bucket-name* 中的物件)。

```
$ aws s3 ls s3://amzn-s3-demo-bucket/example/
2018-12-06 18:59:32          3 MyFile1.txt
```

若要僅顯示特定區域中的儲存貯體和物件，請使用 --region選項

```
$ aws s3 ls --region us-east-2
2018-12-06 18:59:32          3 MyFile1.txt
```

如果您有大量的儲存貯體和物件清單，您可以使用 --max-items或 --page-size選項分頁結果。--max-items 選項會限制通話中傳回的總儲存貯體和物件數量，而--page-size選項會限制頁面上列出的儲存貯體和物件數量。

```
$ aws s3 ls --max-items 100 --page-size 10
```

如需分頁的詳細資訊，請參閱 the section called “--page-size”和 the section called “--max-items”。

刪除儲存貯體

若要刪除儲存貯體，請使用 s3 rb 命令。

語法

```
$ aws s3 rb <target> [--options]
```

s3 rb 範例

以下範例會移除 s3://amzn-s3-demo-bucket 儲存貯體。

```
$ aws s3 rb s3://amzn-s3-demo-bucket
```

根據預設，儲存貯體必須為空才能成功操作。若要移除不是空的儲存貯體，您需要包含 --force 選項。如果您使用受版本控制的儲存貯體 (其包含先前已刪除但仍保留的物件)，則此命令不會允許您移除該儲存貯體。您必須先刪除所有內容。

以下範例命令會在儲存貯體中刪除所有物件和前綴，然後刪除該儲存貯體。

```
$ aws s3 rb s3://amzn-s3-demo-bucket --force
```

刪除物件

若要刪除儲存貯體或本機目錄中的物件，請使用 s3 rm 命令。

語法

```
$ aws s3 rm  <target> [--options]
```

如需搭配此命令使用的幾個常見選項和範例，請參閱 s3 命令的常用選項。如需選項的完整清單，請參閱 AWS CLI 命令參考中的 s3 rm。

s3 rm 範例

下列範例刪除來自 s3://amzn-s3-demo-bucket/example 的 filename.txt。

```
$ aws s3 rm s3://amzn-s3-demo-bucket/example/filename.txt
```

以下範例使用 --recursive 選項刪除所有來自 s3://amzn-s3-demo-bucket/example 的物件。

```
$ aws s3 rm s3://amzn-s3-demo-bucket/example --recursive
```

移動物件

使用 s3 mv 命令從儲存貯體或本機目錄移動物件。s3 mv 命令會將來源物件或檔案複製到指定的目的地，然後刪除來源物件或檔案。

語法

```
$ aws s3 mv <source> <target> [--options]
```

如需搭配此命令使用的幾個常見選項和範例，請參閱 s3 命令的常用選項。如需可用選項的完整清單，請參閱 AWS CLI 命令參考中的 s3 mv。

> **Warning**
>
> 如果您在 Amazon S3 來源或目的地 URIs 中使用任何類型的存取點 ARNs 或存取點別名，您必須特別注意來源和目的地 Amazon S3 URIs解析為不同的基礎儲存貯體。如果來源和目的地儲存貯體相同，可以將來源檔案或物件移至其本身，這可能會導致意外刪除您的來源檔案或物件。若要驗證來源和目的地儲存貯體是否不同，請使用 `--validate-same-s3-paths` 參數，或將環境變數設定為 `AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS` `true`。

s3 mv 範例

下列範例會將所有物件從 `s3://amzn-s3-demo-bucket/example` 移動至 `s3://amzn-s3-demo-bucket/`。

```
$ aws s3 mv s3://amzn-s3-demo-bucket/example s3://amzn-s3-demo-bucket/
```

以下範例會使用 `s3 mv` 命令將本機檔案從目前的工作目錄移動至 Amazon S3 儲存貯體。

```
$ aws s3 mv filename.txt s3://amzn-s3-demo-bucket
```

以下範例將檔案從 Amazon S3 儲存貯體移動至目前的工作目錄，其中 `./` 指定您目前使用中的目錄。

```
$ aws s3 mv s3://amzn-s3-demo-bucket/filename.txt ./
```

複製物件

使用 s3 cp 命令從儲存貯體或本機目錄複製物件。

語法

```
$ aws s3 cp <source> <target> [--options]
```

您可以使用破折號參數將檔案串流傳輸至標準輸入 (`stdin`) 或標準輸出 (`stdout`)。

> **Warning**
>
> 如果您使用的是 PowerShell，Shell 可能會改變 CRLF 的編碼，或者將 CRLF 新增至管道輸入或輸出，或者重新導向的輸出。

此 `s3 cp` 命令使用以下語法將檔案串流從 `stdin` 上傳至指定的儲存貯體。

語法

```
$ aws s3 cp - <target> [--options]
```

此 `s3 cp` 命令會使用下列語法來為 `stdout` 下載 Amazon S3 檔案串流。

語法

```
$ aws s3 cp <target> [--options] -
```

如需搭配此命令使用的幾個常見選項和範例，請參閱 s3 命令的常用選項。如需選項的完整清單，請參閱 AWS CLI 命令參考中的 `s3 cp`。

`s3 cp` 範例

下列範例會將來自 `s3://amzn-s3-demo-bucket/example` 的物件複製到 `s3://amzn-s3-demo-bucket/`。

```
$ aws s3 cp s3://amzn-s3-demo-bucket/example s3://amzn-s3-demo-bucket/
```

以下範例會使用 `s3 cp` 命令將本機檔案從目前的工作目錄複製至 Amazon S3 儲存貯體。

```
$ aws s3 cp filename.txt s3://amzn-s3-demo-bucket
```

以下範例將檔案從您的 Amazon S3 儲存貯體複製至目前的工作目錄，其中 `./` 指定您目前使用中的目錄。

```
$ aws s3 cp s3://amzn-s3-demo-bucket/filename.txt ./
```

以下範例會使用 Echo 將文字「Hello World」串流至 `s3://bucket-name/filename.txt` 檔案。

```
$ echo "hello world" | aws s3 cp - s3://amzn-s3-demo-bucket/filename.txt
```

下列範例會串流 s3://amzn-s3-demo-bucket/filename.txt 檔案至 stdout 並將內容列印至主控台。

```
$ aws s3 cp s3://amzn-s3-demo-bucket/filename.txt -
hello world
```

下列範例會串流 s3://bucket-name/pre 內容至 stdout，使用 bzip2 命令來壓縮檔案，再上傳名為 key.bz2 的壓縮檔至 s3://bucket-name。

```
$ aws s3 cp s3://amzn-s3-demo-bucket/pre - | bzip2 --best | aws s3 cp - s3://amzn-s3-demo-bucket/key.bz2
```

同步物件

s3 sync 命令可同步儲存貯體和目錄的內容，或同步兩個儲存貯體的內容。一般來說，s3 sync 會複製遺失或過期的檔案，或是來源與目標之間的物件。但是您也可以提供 --delete 選項，來從目標中移除沒有出現在來源中的檔案或物件。

語法

```
$ aws s3 sync <source> <target> [--options]
```

如需搭配此命令使用的幾個常見選項和範例，請參閱 s3 命令的常用選項。如需選項的完整清單，請參閱 AWS CLI 命令參考中的 s3 sync。

s3 同步範例

下列範例會將名為 amzn-s3-demo-bucket 之儲存貯體中名為 路徑的 Amazon S3 字首內容與目前的工作目錄同步。

s3 sync 會更新任何與目的地檔案具有相同名字，但是檔案大小或修改時間不同的檔案。輸出顯示在同步期間所執行的特定操作。請注意，操作會使用 s3://amzn-s3-demo-bucket/path/MySubdirectory 遞迴同步處理子目錄 MySubdirectory 及其內容。

```
$ aws s3 sync . s3://amzn-s3-demo-bucket/path
upload: MySubdirectory\MyFile3.txt to s3://amzn-s3-demo-bucket/path/MySubdirectory/MyFile3.txt
upload: MyFile2.txt to s3://amzn-s3-demo-bucket/path/MyFile2.txt
upload: MyFile1.txt to s3://amzn-s3-demo-bucket/path/MyFile1.txt
```

下列範例 (為前一個範例的延伸) 示範如何使用 --delete 選項。

```
// Delete local file
$ rm ./MyFile1.txt

// Attempt sync without --delete option - nothing happens
$ aws s3 sync . s3://amzn-s3-demo-bucket/path

// Sync with deletion - object is deleted from bucket
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --delete
delete: s3://amzn-s3-demo-bucket/path/MyFile1.txt

// Delete object from bucket
$ aws s3 rm s3://amzn-s3-demo-bucket/path/MySubdirectory/MyFile3.txt
delete: s3://amzn-s3-demo-bucket/path/MySubdirectory/MyFile3.txt

// Sync with deletion - local file is deleted
$ aws s3 sync s3://amzn-s3-demo-bucket/path . --delete
delete: MySubdirectory\MyFile3.txt

// Sync with Infrequent Access storage class
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --storage-class STANDARD_IA
```

在使用 --delete 選項時，--exclude 和 --include 選項可以在 s3 sync 操作期間篩選要刪除的檔案或物件。在此情況下，參數字串必須針對目標目錄或儲存貯體，指定檔案要排除或包含在刪除操作中。下列顯示一個範例。

```
Assume local directory and s3://amzn-s3-demo-bucket/path currently in sync and each
 contains 3 files:
MyFile1.txt
MyFile2.rtf
MyFile88.txt
'''

// Sync with delete, excluding files that match a pattern. MyFile88.txt is deleted,
 while remote MyFile1.txt is not.
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --delete --exclude "path/MyFile?.txt"
delete: s3://amzn-s3-demo-bucket/path/MyFile88.txt
'''

// Sync with delete, excluding MyFile2.rtf - local file is NOT deleted
$ aws s3 sync s3://amzn-s3-demo-bucket/path . --delete --exclude "./MyFile2.rtf"
download: s3://amzn-s3-demo-bucket/path/MyFile1.txt to MyFile1.txt
```

```
'''

// Sync with delete, local copy of MyFile2.rtf is deleted
$ aws s3 sync s3://amzn-s3-demo-bucket/path . --delete
delete: MyFile2.rtf
```

s3 命令的常用選項

本主題中描述的指令經常使用下列選項。如需可在命令上使用的選項完整清單，請參閱第 AWS CLI 2 版參考指南中的特定命令。

acl

s3 sync 和 s3 cp 可以使用 --acl 選項。這可讓您為複製至 Amazon S3 的檔案設定存取許可。--acl 選項接受 private、public-read 和 public-read-write 值。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的固定 ACL。

```
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --acl public-read
```

排除

在使用 s3 cp、s3 mv、s3 sync 或 s3 rm 命令時，您可以他透過 --exclude 或 --include 選項來篩選結果。--exclude 選項會將規則設定為僅從命令中排除物件，而選項會依指定的順序套用。如以下範例所示。

```
Local directory contains 3 files:
MyFile1.txt
MyFile2.rtf
MyFile88.txt

// Exclude all .txt files, resulting in only MyFile2.rtf being copied
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --exclude "*.txt"

// Exclude all .txt files but include all files with the "MyFile*.txt" format,
 resulting in, MyFile1.txt, MyFile2.rtf, MyFile88.txt being copied
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --exclude "*.txt" --include
 "MyFile*.txt"

// Exclude all .txt files, but include all files with the "MyFile*.txt" format,
 but exclude all files with the "MyFile?.txt" format resulting in, MyFile2.rtf and
 MyFile88.txt being copied
```

```
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --exclude "*.txt" --include
 "MyFile*.txt" --exclude "MyFile?.txt"
```

包含

在使用 s3 cp、s3 mv、s3 sync 或 s3 rm 命令時，您可以透過 --exclude 或 --include 選項來篩選結果。--include 選項會將規則設定為僅包含命令指定的物件，而選項會依指定的順序套用。如以下範例所示。

```
Local directory contains 3 files:
MyFile1.txt
MyFile2.rtf
MyFile88.txt

// Include all .txt files, resulting in MyFile1.txt and MyFile88.txt being copied
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "*.txt"

// Include all .txt files but exclude all files with the "MyFile*.txt" format,
 resulting in no files being copied
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "*.txt" --exclude
 "MyFile*.txt"

// Include all .txt files, but exclude all files with the "MyFile*.txt" format, but
 include all files with the "MyFile?.txt" format resulting in MyFile1.txt being
 copied

$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "*.txt" --exclude
 "MyFile*.txt" --include "MyFile?.txt"
```

授予

s3 cp、s3 mv 和 s3 sync 命令包含 --grants 選項，您可用來向指定使用者或群組授予對物件的許可。使用下列語法將 --grants 選項設定為許可清單。用您的數值取代 Permission、Grantee_Type 和 Grantee_ID。

語法

```
--grants Permission=Grantee_Type=Grantee_ID
         [Permission=Grantee_Type=Grantee_ID ...]
```

每個數值包含下列元素：

- *Permission* – 指定授予的許可。可設定為 read、readacl、writeacl 或 full。

- *Grantee_Type* – 指定識別被授予者的方式。可設定為 uri、emailaddress 或 id。
- *Grantee_ID* – 根據 *Grantee_Type* 指定被授予者。
  - uri – 群組的 URI。如需詳細資訊，請參閱被授予者是什麼？
  - emailaddress – 帳戶的電子郵件地址。
  - id – 帳戶的正式 ID。

如需有關 Amazon S3 存取控制的詳細資訊，請參閱存取控制。

下列範例將物件複製到儲存貯體中。其授予所有人對該物件的 read 許可，並授予和 full 相關聯的帳戶對該物件的 read 取可 (readacl、writeacl 和 user@example.com)。

```
$ aws s3 cp file.txt s3://amzn-s3-demo-bucket/ --grants read=uri=http://
acs.amazonaws.com/groups/global/AllUsers full=emailaddress=user@example.com
```

您也可以為您上傳到 Amazon S3 的物件，指定非預設儲存方案 (REDUCED_REDUNDANCY 或 STANDARD_IA)。若要這麼做，請使用 --storage-class 選項。

```
$ aws s3 cp file.txt s3://amzn-s3-demo-bucket/ --storage-class REDUCED_REDUNDANCY
```

recursive

當您使用此選項時，會對指定目錄或前綴下的所有檔案或物件執行該命令。以下範例刪除 s3://amzn-s3-demo-bucket/path 及其所有內容。

```
$ aws s3 rm s3://amzn-s3-demo-bucket/path --recursive
```

資源

AWS CLI 參考：

- aws s3
- aws s3 cp
- aws s3 mb
- aws s3 mv
- aws s3 ls
- aws s3 rb
- aws s3 rm

- `aws s3 sync`

服務參考：

- 《Amazon S3 使用者指南》中的使用 Amazon S3 儲存貯體 Amazon S3
- 《Amazon S3 使用者指南》中的使用 Amazon S3 物件 Amazon S3
- 《Amazon S3 使用者指南》中的使用字首和分隔符號以階層方式列出金鑰
- 使用《Amazon S3 使用者指南》中的 適用於 .NET 的 AWS SDK （低階） 中止分段上傳至 S3 儲存貯體 Amazon S3

在 中使用 API 層級 (s3api) 命令 AWS CLI

API 層級命令 (包含在 `s3api` 命令集中) 提供對 Amazon Simple Storage Service (Amazon S3) API 的直接存取，並啟用在高階 `s3` 命令中未公開的部分操作。這些命令等同於對服務功能提供 API 層級存取的其他 AWS 服務。如需這些 `s3` 命令的詳細資訊，請參閱 在 中使用高階 (s3) 命令 AWS CLI

本主題提供範例，示範如何使用映射到 Amazon S3 API 的較低層級命令。此外，您可以在第 AWS CLI 2 版參考指南的 `s3api`區段中找到每個 S3 API 命令的範例。

主題

- 先決條件
- 套用自訂 ACL
- 設定記錄政策
- 資源

先決條件

若要執行 `s3api` 命令，您需要：

- 安裝及設定 AWS CLI。如需詳細資訊，請參閱the section called “安裝/更新”及的身分驗證和存取憑證 AWS CLI。
- 您使用的設定檔必須具有許可，以允許範例執行 AWS 的操作。
- 了解這些 Amazon S3 術語：
  - 儲存貯體 – 頂層 Amazon S3 資料夾。
  - 前綴 – 儲存貯體中的 Amazon S3 資料夾。

- 物件 – 託管於 Amazon S3 儲存貯體中的任一個項目。

套用自訂 ACL

透過高階命令，您可使用 `--acl` 選項在 Amazon S3 物件上套用預先定義的存取控制清單 (ACL)。但不能使用該命令來設定整個儲存貯體的 ACL。不過，您可以使用 `put-bucket-acl` API 層級命令做到這一點。

下列範例顯示如何將完全控制權授予兩個 AWS 使用者 (user1@example.com 和 user2@example.com)，以及授予每個人讀取許可。"everyone" 的識別符來自您當做參數傳遞的特殊 URI。

```
$ aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --grant-full-control
 'emailaddress="user1@example.com",emailaddress="user2@example.com"' --grant-read
 'uri="http://acs.amazonaws.com/groups/global/AllUsers"'
```

如需有關如何建構 ACL 的詳細資訊，請參閱《Amazon Simple Storage Service API 參考》中的 PUT 儲存貯體 acl。CLI 中的 `s3api` ACL 命令 (如 `put-bucket-acl`) 使用相同的速記參數表示法。

設定記錄政策

API 命令 `put-bucket-logging` 設定儲存貯體記錄政策。

在下列範例中， AWS 使用者 user@example.com 被授予對日誌檔案的完全控制權，且所有使用者都可以讀取這些檔案。請注意，授予 Amazon S3 記錄傳送系統 (由 URI 指定) 將日誌讀取和寫入至儲存貯體的必要許可時，也需要 `put-bucket-acl` 命令。

```
$ aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --grant-read-acp 'URI="http://
acs.amazonaws.com/groups/s3/LogDelivery"' --grant-write 'URI="http://acs.amazonaws.com/
groups/s3/LogDelivery"'
$ aws s3api put-bucket-logging --bucket amzn-s3-demo-bucket --bucket-logging-status
 file://logging.json
```

上一個命令中的 `logging.json` 檔案包含下列內容。

```
{
  "LoggingEnabled": {
    "TargetBucket": "amzn-s3-demo-bucket",
    "TargetPrefix": "amzn-s3-demo-bucketLogs/",
    "TargetGrants": [
```

```
      {
        "Grantee": {
          "Type": "AmazonCustomerByEmail",
          "EmailAddress": "user@example.com"
        },
        "Permission": "FULL_CONTROL"
      },
      {
        "Grantee": {
          "Type": "Group",
          "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
        },
        "Permission": "READ"
      }
    ]
  }
}
```

資源

AWS CLI 參考：

- aws s3api
- aws s3api put-bucket-acl
- aws s3api put-bucket-logging

服務參考：

- 《Amazon S3 使用者指南》中的使用 Amazon S3 儲存貯體 Amazon S3
- 《Amazon S3 使用者指南》中的使用 Amazon S3 物件
- 《Amazon S3 使用者指南》中的使用字首和分隔符號以階層方式列出金鑰
- 使用《Amazon S3 使用者指南》中的 適用於 .NET 的 AWS SDK （低階） 中止分段上傳至 S3 儲存貯體 Amazon S3

中的 Amazon S3 儲存貯體生命週期指令碼範例 AWS CLI

本主題使用 Bash 指令碼範例來說明使用 AWS Command Line Interface (AWS CLI) 的 Amazon S3 儲存貯體生命週期操作。此指令碼範例會使用 aws s3api 命令組。Shell 指令碼是專為在命令列界面中執行而設計的程式。

主題

- 開始之前
- 關於此範例
- 檔案
- 參考

開始之前

在可以執行下列任何範例之前，您必須先完成下列事項。

- 安裝及設定 AWS CLI。如需詳細資訊，請參閱the section called “安裝/更新”及的身分驗證和存取憑證 AWS CLI。
- 您使用的設定檔必須具有許可，以允許範例執行 AWS 的操作。

- AWS 最佳實務是授予此程式碼的最低權限，或僅授予執行任務所需的許可。如需詳細資訊，請參閱《IAM 使用者指南》中的授予最低權限。
- 此程式碼尚未在所有區域中進行測試 AWS 。某些 AWS 服務僅適用於特定 區域。如需詳細資訊，請參閱《AWS 一般參考指南》中的服務端點和配額。
- 執行此程式碼可能會導致 AWS 您的帳戶產生費用。您有責任確保在使用完該指令碼建立的所有資源後將這些資源移除。

Amazon S3 服務使用以下術語：

- 儲存貯體 – 頂層 Amazon S3 資料夾。
- 前綴 – 儲存貯體中的 Amazon S3 資料夾。
- 物件 – 託管於 Amazon S3 儲存貯體中的任一個項目。

關於此範例

此範例示範如何使用 Shell 指令碼檔案中的一組函數與某些基本 Amazon S3 操作互動。這些函數位於名為 `bucket-operations.sh` 的 Shell 指令碼檔案中。您可以在另一個檔案中呼叫這些函數。每個指令碼檔案包含描述每個函數的註釋。

若要查看每個步驟的中繼結果，請執行具有 `-i` 參數的指令碼。您可以使用 Amazon S3 主控台檢視儲存貯體的目前狀態或其內容。在出現提示時按下 ENTER 後，指令碼才會繼續執行下一個步驟。

如需完整範例和可下載的指令碼檔案，請參閱 GitHub 上 AWS 程式碼範例儲存庫中的 Amazon S3 儲存貯體生命週期操作。

檔案

本範例內含下列檔案：

bucket-operations.sh

此主指令碼檔案可從另一個檔案獲取。它包含執行以下任務的函數：

- 建立一個儲存貯體並驗證它是否存在
- 將檔案從本機電腦複製至儲存貯體
- 將檔案從一個儲存貯體位置複製到其他儲存貯體位置
- 列出儲存貯體的內容
- 從儲存貯體刪除檔案
- 刪除儲存貯體

檢視 GitHub 上的 bucket-operations.sh。

test-bucket-operations.sh

Shell 指令碼檔案 test-bucket-operations.sh 示範了如何透過獲取 bucket-operations.sh 檔案並呼叫每個函數來呼叫函數。呼叫函數後，測試指令碼會刪除它所建立的所有資源。

檢視 GitHub 上的 test-bucket-operations.sh。

awsdocs-general.sh

指令碼檔案 awsdocs-general.sh 會保存在各個 AWS CLI進階程式碼範例中使用的一般用途函數。

檢視 GitHub 上的 awsdocs-general.sh。

參考

AWS CLI 參考：

- aws s3api
- aws s3api create-bucket
- aws s3api copy-object
- aws s3api delete-bucket
- aws s3api delete-object
- aws s3api head-bucket
- aws s3api list-objects
- aws s3api put-object

其他參考：

- 《Amazon S3 使用者指南》中的使用 Amazon S3 儲存貯體 Amazon S3
- 《Amazon S3 使用者指南》中的使用 Amazon S3 物件
- 若要檢視和貢獻 AWS SDK 和 AWS CLI 程式碼範例，請參閱 GitHub 上的AWS 程式碼範例儲存庫。

在 中存取 Amazon SNS AWS CLI

您可以使用 AWS Command Line Interface () 存取 Amazon Simple Notification Service (Amazon SNS) 的功能AWS CLI。若要列出 Amazon SNS 的 AWS CLI 命令，請使用下列命令。

```
aws sns help
```

在執行任何命令前，請先設定您的預設憑證。如需詳細資訊，請參閱設定 的設定 AWS CLI。

本主題顯示執行 Amazon SNS 常見任務的 AWS CLI 命令範例。

主題

- 建立主題
- 訂閱主題
- 發布到主題
- 取消訂閱主題
- 刪除主題

建立主題

若要建立主題，請使用 sns create-topic 命令，並指定要指派給主題的名稱。

```
$ aws sns create-topic --name my-topic
{
    "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
}
```

記下回應的 TopicArn，稍後您將使用它來發佈訊息。

訂閱主題

若要訂閱主題，請使用 sns subscribe 命令。

以下範例指定 email 通訊協定和 notification-endpoint 的電子郵件地址。

```
$ aws sns subscribe --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic --
protocol email --notification-endpoint saanvi@example.com
{
    "SubscriptionArn": "pending confirmation"
}
```

AWS 立即透過電子郵件將確認訊息傳送至您在 subscribe命令中指定的地址。電子郵件訊息會有以下文字。

```
You have chosen to subscribe to the topic:
arn:aws:sns:us-west-2:123456789012:my-topic
To confirm this subscription, click or visit the following link (If this was in error
 no action is necessary):
Confirm subscription
```

收件人按一下 Confirm subscription (確認訂閱) 連結之後，收件人的瀏覽器會顯示一個通知訊息，其中包含以下資訊。

```
Subscription confirmed!

You have subscribed saanvi@example.com to the topic:my-topic.

Your subscription's id is:
```

```
arn:aws:sns:us-west-2:123456789012:my-topic:1328f057-de93-4c15-512e-8bb22EXAMPLE

If it was not your intention to subscribe, click here to unsubscribe.
```

發布到主題

若要將訊息傳送給一個主題的所有訂閱者，請使用 sns publish 命令。

以下範例會傳送「Hello World!」訊息 給指定主題的所有訂閱者。

```
$ aws sns publish --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic --
message "Hello World!"
{
    "MessageId": "4e41661d-5eec-5ddf-8dab-2c867EXAMPLE"
}
```

在此範例中， AWS 會傳送文字為「Hello World！」的電子郵件訊息 至 saanvi@example.com。

取消訂閱主題

若要取消訂閱主題並停止接收發佈給該主題的訊息，請使用 sns unsubscribe 命令並指定您想要取消訂閱的主題 ARN。

```
$ aws sns unsubscribe --subscription-arn arn:aws:sns:us-west-2:123456789012:my-
topic:1328f057-de93-4c15-512e-8bb22EXAMPLE
```

若要驗證是否成功取消訂閱，請使用 sns list-subscriptions 命令來確認 ARN 不再顯示於清單中。

```
$ aws sns list-subscriptions
```

刪除主題

若要刪除主題，請使用 sns delete-topic 命令。

```
$ aws sns delete-topic --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic
```

若要驗證 是否 AWS 已成功刪除主題，請使用 sns list-topics命令來確認主題不再出現在清單中。

```
$ aws sns list-topics
```

AWS CLI 命令範例

本主題中的程式碼範例說明如何 AWS Command Line Interface 搭配 使用 AWS。

基本概念是程式碼範例，這些範例說明如何在服務內執行基本操作。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

案例是向您展示如何呼叫服務中的多個函數或與其他 AWS 服務組合來完成特定任務的程式碼範例。

有些服務包含其他範例類別，示範如何利用服務特定的程式庫或函數。

服務

- 使用 的 ACM 範例 AWS CLI
- 使用 的 API Gateway 範例 AWS CLI
- 使用 的 API Gateway HTTP 和 WebSocket API 範例 AWS CLI
- 使用 的 API Gateway Management API 範例 AWS CLI
- 使用 的 App Mesh 範例 AWS CLI
- 使用 的 App Runner 範例 AWS CLI
- AWS AppConfig 使用 的範例 AWS CLI
- 使用 的 Application Auto Scaling 範例 AWS CLI
- 使用 的應用程式探索服務範例 AWS CLI
- 使用 的 AppRegistry 範例 AWS CLI
- 使用 的 Athena 範例 AWS CLI
- 使用 的 Auto Scaling 範例 AWS CLI
- 使用 的 Auto Scaling Plans 範例 AWS CLI
- AWS Backup 使用 的範例 AWS CLI
- AWS Batch 使用 的範例 AWS CLI
- AWS Budgets 使用 的範例 AWS CLI
- 使用 的 Amazon Chime 範例 AWS CLI

- 使用 的雲端控制 API 範例 AWS CLI
- AWS Cloud Map 使用 的範例 AWS CLI
- AWS Cloud9 使用 的範例 AWS CLI
- AWS CloudFormation 使用 的範例 AWS CLI
- 使用 的 CloudFront 範例 AWS CLI
- 使用 的 Amazon CloudSearch 範例 AWS CLI
- 使用 的 CloudTrail 範例 AWS CLI
- 使用 的 CloudWatch 範例 AWS CLI
- 使用 的 CloudWatch Logs 範例 AWS CLI
- 使用 的 CloudWatch Network Monitoring 範例 AWS CLI
- 使用 的 CloudWatch 可觀測性存取監視器範例 AWS CLI
- 使用 的 CloudWatch 可觀測性管理員範例 AWS CLI
- 使用 的 CloudWatch Synthetics 範例 AWS CLI
- 使用 的 CodeArtifact 範例 AWS CLI
- 使用 的 CodeBuild 範例 AWS CLI
- 使用 的 CodeCommit 範例 AWS CLI
- 使用 的 CodeDeploy 範例 AWS CLI
- 使用 的 CodeGuru Reviewer 範例 AWS CLI
- 使用 的 CodePipeline 範例 AWS CLI
- AWS CodeStar 使用 的通知範例 AWS CLI
- 使用 的 CodeConnections 範例 AWS CLI
- 使用 的 Amazon Cognito 身分範例 AWS CLI
- 使用 的 Amazon Cognito 身分提供者範例 AWS CLI
- 使用 的 Amazon Comprehend 範例 AWS CLI
- 使用 的 Amazon Comprehend Medical 範例 AWS CLI
- AWS Config 使用 的範例 AWS CLI
- 使用 的 Amazon Connect 範例 AWS CLI
- AWS Cost and Usage Report 使用 的範例 AWS CLI
- 使用 的 Cost Explorer Service 範例 AWS CLI
- 使用 的 Firehose 範例 AWS CLI

- 使用 的 Amazon Data Lifecycle Manager 範例 AWS CLI
- AWS Data Pipeline 使用 的範例 AWS CLI
- 使用 的 DataSync 範例 AWS CLI
- 使用 的 DAX 範例 AWS CLI
- 使用 的偵測範例 AWS CLI
- 使用 的 Device Farm 範例 AWS CLI
- AWS Direct Connect 使用 的範例 AWS CLI
- AWS Directory Service 使用 的範例 AWS CLI
- AWS Directory Service 使用 的資料範例 AWS CLI
- AWS DMS 使用 的範例 AWS CLI
- 使用 的 Amazon DocumentDB 範例 AWS CLI
- 使用 的 DynamoDB 範例 AWS CLI
- 使用 的 DynamoDB Streams 範例 AWS CLI
- 使用 的 Amazon EC2 範例 AWS CLI
- 使用 的 Amazon EC2 Instance Connect 範例 AWS CLI
- 使用 的 Amazon ECR 範例 AWS CLI
- 使用 的 Amazon ECR 公有範例 AWS CLI
- 使用 的 Amazon ECS 範例 AWS CLI
- 使用 的 Amazon EFS 範例 AWS CLI
- 使用 的 Amazon EKS 範例 AWS CLI
- 使用 的 Elastic Beanstalk 範例 AWS CLI
- Elastic Load Balancing - 使用 的第 1 版範例 AWS CLI
- Elastic Load Balancing - 使用 的第 2 版範例 AWS CLI
- 使用 的 Elastic Transcoder 範例 AWS CLI
- 使用 的 ElastiCache 範例 AWS CLI
- 使用 的 MediaStore 範例 AWS CLI
- 使用 的 Amazon EMR 範例 AWS CLI
- 使用 的 Amazon EMR on EKS 範例 AWS CLI
- 使用 的 EventBridge 範例 AWS CLI
- 使用 的 EventBridge 管道範例 AWS CLI

- 使用 的 Firewall Manager 範例 AWS CLI
- AWS FIS 使用 的範例 AWS CLI
- 使用 的 Amazon GameLift 伺服器範例 AWS CLI
- 使用 的 Global Accelerator 範例 AWS CLI
- AWS Glue 使用 的範例 AWS CLI
- 使用 的 GuardDuty 範例 AWS CLI
- AWS Health 使用 的範例 AWS CLI
- HealthImaging 範例使用 AWS CLI
- 使用 的 HealthLake 範例 AWS CLI
- 使用 的 HealthOmics 範例 AWS CLI
- 使用 的 IAM 範例 AWS CLI
- 使用 的 IAM Access Analyzer 範例 AWS CLI
- 使用 的映像建置器範例 AWS CLI
- 使用 的 Incident Manager 範例 AWS CLI
- 使用 的 Incident Manager Contacts 範例 AWS CLI
- 使用 的 Amazon Inspector 範例 AWS CLI
- AWS IoT 使用 的範例 AWS CLI
- AWS IoT Analytics 使用 的範例 AWS CLI
- 使用 的 Device Advisor 範例 AWS CLI
- AWS IoT data 使用 的範例 AWS CLI
- AWS IoT Events 使用 的範例 AWS CLI
- AWS IoT Events-Data 使用 的範例 AWS CLI
- AWS IoT Greengrass 使用 的範例 AWS CLI
- AWS IoT Greengrass V2 使用 的範例 AWS CLI
- AWS IoT Jobs SDK release 使用 的範例 AWS CLI
- AWS IoT SiteWise 使用 的範例 AWS CLI
- AWS IoT Things Graph 使用 的範例 AWS CLI
- AWS IoT Wireless 使用 的範例 AWS CLI
- 使用 的 Amazon IVS 範例 AWS CLI
- 使用 的 Amazon IVS 聊天功能範例 AWS CLI

- 使用 的 Amazon IVS 即時串流範例 AWS CLI
- 使用 的 Amazon Kendra 範例 AWS CLI
- 使用 的 Kinesis 範例 AWS CLI
- AWS KMS 使用 的範例 AWS CLI
- 使用 的 Lake Formation 範例 AWS CLI
- 使用 的 Lambda 範例 AWS CLI
- 使用 的授權管理員範例 AWS CLI
- Lightsail 使用 的範例 AWS CLI
- 使用 的 Macie 範例 AWS CLI
- 使用 的 Amazon Managed Grafana 範例 AWS CLI
- 使用 的 MediaConnect 範例 AWS CLI
- 使用 的 MediaConvert 範例 AWS CLI
- 使用 的 MediaLive 範例 AWS CLI
- 使用 的 MediaPackage 範例 AWS CLI
- 使用 的 MediaPackage VOD 範例 AWS CLI
- 使用 的 MediaStore 資料平面範例 AWS CLI
- 使用 的 MediaTailor 範例 AWS CLI
- 使用 的 MemoryDB 範例 AWS CLI
- 使用 的 Amazon MSK 範例 AWS CLI
- 使用 的網路流量監控範例 AWS CLI
- 使用 的 Network Manager 範例 AWS CLI
- 使用 的 OpenSearch Service 範例 AWS CLI
- OpsWorks 使用 的範例 AWS CLI
- AWS OpsWorks CM 使用 的範例 AWS CLI
- 使用 的組織範例 AWS CLI
- AWS Outposts 使用 的範例 AWS CLI
- AWS Payment Cryptography 使用 的範例 AWS CLI
- AWS Payment Cryptography 使用 的資料平面範例 AWS CLI
- 使用 的 Amazon Pinpoint 範例 AWS CLI
- 使用 的 Amazon Polly 範例 AWS CLI

- AWS 價格表 使用 的範例 AWS CLI
- AWS Private CA 使用 的範例 AWS CLI
- AWS Proton 使用 的範例 AWS CLI
- 使用 的 QLDB 範例 AWS CLI
- 使用 的 Amazon RDS 範例 AWS CLI
- 使用 的 Amazon RDS Data Service 範例 AWS CLI
- 使用 的 Amazon RDS Performance Insights 範例 AWS CLI
- 使用 的 Amazon Redshift 範例 AWS CLI
- 使用 的 Amazon Rekognition 範例 AWS CLI
- AWS RAM 使用 的範例 AWS CLI
- 使用 的資源總管範例 AWS CLI
- 使用 的資源群組範例 AWS CLI
- 資源群組使用 標記 API 範例 AWS CLI
- AWS RoboMaker 使用 的範例 AWS CLI
- 使用 的 Route 53 範例 AWS CLI
- 使用 的 Route 53 網域註冊範例 AWS CLI
- 使用 的 Route 53 Profiles 範例 AWS CLI
- 使用 的 Route 53 Resolver 範例 AWS CLI
- 使用 的 Amazon S3 範例 AWS CLI
- 使用 的 Amazon S3 控制範例 AWS CLI
- 使用 的 S3 Glacier 範例 AWS CLI
- 使用 的 Secrets Manager 範例 AWS CLI
- 使用 的 Security Hub 範例 AWS CLI
- 使用 的 Security Lake 範例 AWS CLI
- AWS Serverless Application Repository 使用 的範例 AWS CLI
- 使用 的 Service Catalog 範例 AWS CLI
- 使用 Service Quotas範例 AWS CLI
- 使用 的 Amazon SES 範例 AWS CLI
- 使用 的 Shield 範例 AWS CLI
- 使用 的簽署者範例 AWS CLI

- 使用 的 Snowball Edge 範例 AWS CLI
- 使用 的 Amazon SNS 範例 AWS CLI
- 使用 的 Amazon SQS 範例 AWS CLI
- 使用 的 Storage Gateway 範例 AWS CLI
- AWS STS 使用 的範例 AWS CLI
- 支援 使用 的範例 AWS CLI
- 使用 的 Amazon SWF 範例 AWS CLI
- 使用 的 Systems Manager 範例 AWS CLI
- 使用 的 Amazon Textract 範例 AWS CLI
- 使用 的 Amazon Transcribe 範例 AWS CLI
- 使用 的 Amazon Translate 範例 AWS CLI
- Trusted Advisor 使用 的範例 AWS CLI
- 使用 的已驗證許可範例 AWS CLI
- 使用 的 VPC Lattice 範例 AWS CLI
- AWS WAF Classic 使用 的範例 AWS CLI
- AWS WAF Classic 區域性 使用 的範例 AWS CLI
- AWS WAFV2 使用 的範例 AWS CLI
- 使用 的 WorkDocs 範例 AWS CLI
- 使用 的 Amazon WorkMail 範例 AWS CLI
- 使用 的 Amazon WorkMail 訊息流程範例 AWS CLI
- 使用 的 WorkSpaces 範例 AWS CLI
- 使用 的 X-Ray 範例 AWS CLI

使用 的 ACM 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 ACM 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

add-tags-to-certificate

以下程式碼範例顯示如何使用 add-tags-to-certificate。

AWS CLI

將標籤新增至現有的 ACM 憑證

下列add-tags-to-certificate命令會將兩個標籤新增至指定的憑證。使用空格分隔多個標籤：

```
aws acm add-tags-to-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --
tags Key=Admin,Value=Alice Key=Purpose,Value=Website
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddTagsToCertificate。

delete-certificate

以下程式碼範例顯示如何使用 delete-certificate。

AWS CLI

從您的帳戶刪除 ACM 憑證

下列delete-certificate命令會刪除具有指定 ARN 的憑證：

```
aws acm delete-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCertificate。

describe-certificate

以下程式碼範例顯示如何使用 describe-certificate。

AWS CLI

擷取 ACM 憑證中包含的欄位

下列`describe-certificate`命令會使用指定的 ARN 擷取憑證的所有欄位：

```
aws acm describe-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

隨即顯示類似以下的輸出：

```
{
  "Certificate": {
    "CertificateArn":
 "arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012",
    "CreatedAt": 1446835267.0,
    "DomainName": "www.example.com",
    "DomainValidationOptions": [
      {
        "DomainName": "www.example.com",
        "ValidationDomain": "www.example.com",
        "ValidationEmails": [
          "hostmaster@example.com",
          "admin@example.com",
          "owner@example.com.whoisprivacyservice.org",
          "tech@example.com.whoisprivacyservice.org",
          "admin@example.com.whoisprivacyservice.org",
          "postmaster@example.com",
          "webmaster@example.com",
          "administrator@example.com"
        ]
      },
      {
        "DomainName": "www.example.net",
        "ValidationDomain": "www.example.net",
        "ValidationEmails": [
          "postmaster@example.net",
          "admin@example.net",
          "owner@example.net.whoisprivacyservice.org",
          "tech@example.net.whoisprivacyservice.org",
          "admin@example.net.whoisprivacyservice.org",
          "hostmaster@example.net",
          "administrator@example.net",
```

```
            "webmaster@example.net"
          ]
        }
      ],
      "InUseBy": [],
      "IssuedAt": 1446835815.0,
      "Issuer": "Amazon",
      "KeyAlgorithm": "RSA-2048",
      "NotAfter": 1478433600.0,
      "NotBefore": 1446768000.0,
      "Serial": "0f:ac:b0:a3:8d:ea:65:52:2d:7d:01:3a:39:36:db:d6",
      "SignatureAlgorithm": "SHA256WITHRSA",
      "Status": "ISSUED",
      "Subject": "CN=www.example.com",
      "SubjectAlternativeNames": [
        "www.example.com",
        "www.example.net"
      ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCertificate。

export-certificate

以下程式碼範例顯示如何使用 export-certificate。

AWS CLI

匯出私有 CA 發行的私有憑證。

下列export-certificate命令會將私有憑證、憑證鏈和私有金鑰匯出到您的顯示器：

```
aws acm export-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --
passphrase file://path-to-passphrase-file
```

若要將憑證、鏈結和私有金鑰匯出至本機檔案，請使用下列命令：

```
aws acm export-certificate --certificate-
arn arn:aws:acm:region:sccount:certificate/12345678-1234-1234-1234-123456789012 --
passphrase file://path-to-passphrase-file > c:\temp\export.txt
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ExportCertificate。

get-certificate

以下程式碼範例顯示如何使用 get-certificate。

AWS CLI

擷取 ACM 憑證

下列get-certificate命令會擷取指定 ARN 的憑證和憑證鏈：

```
aws acm get-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

隨即顯示類似以下的輸出：

```
{
  "Certificate": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",

  "CertificateChain": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
```

```
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
"-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
"-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCertificate。

import-certificate

以下程式碼範例顯示如何使用 import-certificate。

AWS CLI

將憑證匯入 ACM。

下列import-certificate命令會將憑證匯入 ACM。將檔案名稱取代為您自己的檔案名稱：

```
aws acm import-certificate --certificate file://Certificate.pem --certificate-
chain file://CertificateChain.pem --private-key file://PrivateKey.pem
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ImportCertificate。

list-certificates

以下程式碼範例顯示如何使用 list-certificates。

AWS CLI

列出 AWS 帳戶的 ACM 憑證

下列list-certificates命令會列出您帳戶中憑證ARNs：

```
aws acm list-certificates
```

上述命令會產生類似下列的輸出：

```
{
    "CertificateSummaryList": [
        {
            "CertificateArn":
 "arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012",
            "DomainName": "www.example.com"
        },
        {
            "CertificateArn": "arn:aws:acm:region:account:certificate/aaaaaaaa-bbbb-
cccc-dddd-eeeeeeeeeeee",
            "DomainName": "www.example.net"
        }
    ]
```

```
}
```

您可以決定每次呼叫 時要顯示的憑證數量list-certificates。例如，如果您有四個憑證，並且希望一次顯示不超過兩個憑證，請將max-items引數設定為 2，如下列範例所示：

```
aws acm list-certificates --max-items 2
```

將顯示兩個憑證 ARNs和一個NextToken值：

```
"CertificateSummaryList": [
  {
    "CertificateArn": "arn:aws:acm:region:account: \
            certificate/12345678-1234-1234-1234-123456789012",
    "DomainName": "www.example.com"
  },
  {
    "CertificateArn": "arn:aws:acm:region:account: \
             certificate/aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "DomainName": "www.example.net"
  }
  ],
    "NextToken": "9f4d9f69-275a-41fe-b58e-2b837bd9ba48"
```

若要顯示您帳戶中接下來的兩個憑證，請在下一次呼叫中設定此NextToken值：

```
aws acm list-certificates --max-items 2 --next-token 9f4d9f69-275a-41fe-
b58e-2b837bd9ba48
```

您可以使用 certificate-statuses引數篩選輸出。下列命令會顯示具有 PENDING_VALIDATION 狀態的憑證：

```
aws acm list-certificates --certificate-statuses PENDING_VALIDATION
```

您也可以使用 includes引數篩選輸出。下列命令會顯示依下列屬性篩選的憑證。要顯示的憑證：

```
- Specify that the RSA algorithm and a 2048 bit key are used to generate key pairs.
- Contain a Key Usage extension that specifies that the certificates can be used to
 create digital signatures.
- Contain an Extended Key Usage extension that specifies that the certificates can
 be used for code signing.
```

```
aws acm list-certificates --max-items 10 --includes
 extendedKeyUsage=CODE_SIGNING,keyUsage=DIGITAL_SIGNATURE,keyTypes=RSA_2048
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCertificates。

list-tags-for-certificate

以下程式碼範例顯示如何使用 list-tags-for-certificate。

AWS CLI

列出套用至 ACM 憑證的標籤

下列list-tags-for-certificate命令會列出套用至您帳戶中憑證的標籤：

```
aws acm list-tags-for-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

上述命令會產生類似下列的輸出：

```
{
  "Tags": [
      {
          "Value": "Website",
          "Key": "Purpose"
      },
      {
          "Value": "Alice",
          "Key": "Admin"
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTagsForCertificate。

remove-tags-from-certificate

以下程式碼範例顯示如何使用 remove-tags-from-certificate。

AWS CLI

從 ACM 憑證移除標籤

下列remove-tags-from-certificate命令會從指定的憑證中移除兩個標籤。使用空格分隔多個標籤：

```
aws acm remove-tags-from-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --
tags Key=Admin,Value=Alice Key=Purpose,Value=Website
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveTagsFromCertificate。

request-certificate

以下程式碼範例顯示如何使用 request-certificate。

AWS CLI

請求新的 ACM 憑證

下列request-certificate命令會使用 DNS 驗證請求 www.example.com 網域的新憑證：

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS
```

您可以輸入冪等字符來區分對 的呼叫request-certificate：

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS --
idempotency-token 91adc45q
```

您可以輸入一或多個主體替代名稱，以請求可保護多個頂點網域的憑證：

```
aws acm request-certificate --domain-name example.com --validation-method DNS --
idempotency-token 91adc45q --subject-alternative-names www.example.net
```

您可以輸入替代名稱，也可以用來連接您的網站：

```
aws acm request-certificate --domain-name example.com --validation-method DNS --
idempotency-token 91adc45q --subject-alternative-names www.example.com
```

您可以使用星號 (*) 做為萬用字元，為相同網域中的多個子網域建立憑證：

```
aws acm request-certificate --domain-name example.com --validation-method DNS --
idempotency-token 91adc45q --subject-alternative-names *.example.com
```

您也可以輸入多個替代名稱：

```
aws acm request-certificate --domain-name example.com --validation-method DNS --
subject-alternative-names b.example.com c.example.com d.example.com
```

如果您使用電子郵件進行驗證，您可以輸入網域驗證選項，以指定要傳送驗證電子郵件的網域：

```
aws acm request-certificate --domain-name example.com --validation-
method EMAIL --subject-alternative-names www.example.com --domain-validation-
options DomainName=example.com,ValidationDomain=example.com
```

當您請求新憑證時，下列命令會選擇退出憑證透明度記錄：

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS --
options CertificateTransparencyLoggingPreference=DISABLED --idempotency-token 184627
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RequestCertificate。

resend-validation-email

以下程式碼範例顯示如何使用 resend-validation-email。

AWS CLI

為您的 ACM 憑證請求重新傳送驗證電子郵件

下列resend-validation-email命令會告知 Amazon 憑證授權機構將驗證電子郵件傳送至適當的地址：

```
aws acm resend-validation-email --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --
domain www.example.com --validation-domain example.com
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResendValidationEmail。

update-certificate-options

以下程式碼範例顯示如何使用 update-certificate-options。

AWS CLI

更新憑證選項

下列update-certificate-options命令選擇退出憑證透明度記錄：

```
aws acm update-certificate-options --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --
options CertificateTransparencyLoggingPreference=DISABLED
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateCertificateOptions。

使用 的 API Gateway 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 API Gateway 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-api-key`

以下程式碼範例顯示如何使用 `create-api-key`。

AWS CLI

建立為現有 API 和階段啟用的 API 金鑰

命令：

```
aws apigateway create-api-key --name 'Dev API Key' --description 'Used for
 development' --enabled --stage-keys restApiId='a1b2c3d4e5',stageName='dev'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateApiKey。

create-authorizer

以下程式碼範例顯示如何使用 create-authorizer。

AWS CLI

範例 1：建立 API 的字符型 API Gateway 自訂授權方

下列create-authorizer範例會建立字符型授權方。

```
aws apigateway create-authorizer \
    --rest-api-id 1234123412 \
    --name 'First_Token_Custom_Authorizer' \
    --type TOKEN \
    --authorizer-uri 'arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:customAuthFunction/invocations' \
    --identity-source 'method.request.header.Authorization' \
    --authorizer-result-ttl-in-seconds 300
```

輸出：

```
{
    "authType": "custom",
    "name": "First_Token_Custom_Authorizer",
    "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:customAuthFunction/invocations",
    "authorizerResultTtlInSeconds": 300,
    "identitySource": "method.request.header.Authorization",
    "type": "TOKEN",
    "id": "z40xj0"
}
```

範例 2：為 API 建立以 Cognito 使用者集區為基礎的 API Gateway 自訂授權方

下列create-authorizer範例會建立以 Cognito 使用者集區為基礎的 API Gateway 自訂授權方。

```
aws apigateway create-authorizer \
    --rest-api-id 1234123412 \
    --name 'First_Cognito_Custom_Authorizer' \
    --type COGNITO_USER_POOLS \
    --provider-arns 'arn:aws:cognito-idp:us-east-1:123412341234:userpool/us-
east-1_aWcZeQbuD' \
```

```
    --identity-source 'method.request.header.Authorization'
```

輸出：

```
{
    "authType": "cognito_user_pools",
    "identitySource": "method.request.header.Authorization",
    "name": "First_Cognito_Custom_Authorizer",
    "providerARNs": [
        "arn:aws:cognito-idp:us-east-1:342398297714:userpool/us-east-1_qWbZzQhzE"
    ],
    "type": "COGNITO_USER_POOLS",
    "id": "5yid1t"
}
```

範例 3：為 API 建立以請求為基礎的 API Gateway 自訂授權方

下列create-authorizer範例會建立以請求為基礎的授權方。

```
aws apigateway create-authorizer \
    --rest-api-id 1234123412 \
    --name 'First_Request_Custom_Authorizer' \
    --type REQUEST \
    --authorizer-uri 'arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:customAuthFunction/invocations' \
    --identity-source 'method.request.header.Authorization,context.accountId' \
    --authorizer-result-ttl-in-seconds 300
```

輸出：

```
{
    "id": "z40xj0",
    "name": "First_Request_Custom_Authorizer",
    "type": "REQUEST",
    "authType": "custom",
    "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:customAuthFunction/invocations",
    "identitySource": "method.request.header.Authorization,context.accountId",
    "authorizerResultTtlInSeconds": 300
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAuthorizer。

create-base-path-mapping

以下程式碼範例顯示如何使用 create-base-path-mapping。

AWS CLI

建立自訂網域名稱的基本路徑映射

命令：

```
aws apigateway create-base-path-mapping --domain-name subdomain.domain.tld --rest-api-id 1234123412 --stage prod --base-path v1
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateBasePathMapping。

create-deployment

以下程式碼範例顯示如何使用 create-deployment。

AWS CLI

將 API 的已設定資源部署至新的階段

命令：

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --stage-description 'Development Stage' --description 'First deployment to the dev stage'
```

將 API 的已設定資源部署至現有階段

命令：

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --description 'Second deployment to the dev stage'
```

將 API 的已設定資源部署至具有階段變數的現有階段

aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --description 'Third deployment to the dev stage' --variables key='value'，otherKey='otherValue'

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDeployment。

create-domain-name-access-association

以下程式碼範例顯示如何使用 create-domain-name-access-association。

AWS CLI

建立網域名稱存取關聯

下列create-domain-name-access-association範例會在私有自訂網域名稱和 VPC 端點之間建立網域名稱存取關聯。

```
aws apigateway create-domain-name-access-association \
    --domain-name-arn arn:aws:apigateway:us-west-2:111122223333:/domainnames/
my.private.domain.tld+abcd1234 \
    --access-association-source vpce-abcd1234efg \
    --access-association-source-type VPCE
```

輸出：

```
{
    "domainNameAccessAssociationArn": "arn:aws:apigateway:us-west-2:012345678910:/
domainnameaccessassociations/domainname/my.private.domain.tld/vpcesource/vpce-
abcd1234efg
    "accessAssociationSource": "vpce-abcd1234efg",
    "accessAssociationSourceType": "VPCE",
    "domainNameArn" : "arn:aws:apigateway:us-west-2:111122223333:/domainnames/
private.example.com+abcd1234"
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的 API Gateway 中私有 API 的自訂網域名稱。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDomainNameAccessAssociation。

create-domain-name

以下程式碼範例顯示如何使用 create-domain-name。

AWS CLI

範例 1：建立公有自訂網域名稱

下列create-domain-name範例會建立公有自訂網域名稱。

```
aws apigateway create-domain-name \
    --domain-name 'my.domain.tld' \
    --certificate-name 'my.domain.tld cert'\
    --certificate-arn 'arn:aws:acm:us-east-1:012345678910:certificate/fb1b9770-a305-495d-aefb-27e5e101ff3'
```

輸出：

```
{
    "domainName": "my.domain.tld",
    "certificateName": "my.domain.tld cert",
    "certificateArn": "arn:aws:acm:us-east-1:012345678910:certificate/fb1b9770-a305-495d-aefb-27e5e101ff3",
    "certificateUploadDate": "2024-10-08T11:29:49-07:00",
    "distributionDomainName": "abcd1234.cloudfront.net",
    "distributionHostedZoneId": "Z2FDTNDATAQYW2",
    "endpointConfiguration": {
        "types": [
            "EDGE"
        ]
    },
    "domainNameStatus": "AVAILABLE",
    "securityPolicy": "TLS_1_2"
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的 API Gateway 中公有 REST API 的自訂網域名稱。 Amazon API Gateway

範例 2：建立私有自訂網域名稱

下列create-domain-name範例會建立私有自訂網域名稱。

```
aws apigateway create-domain-name \
    --domain-name 'my.private.domain.tld' \
    --certificate-name 'my.domain.tld cert' \
    --certificate-arn 'arn:aws:acm:us-east-1:012345678910:certificate/fb1b9770-a305-495d-aefb-27e5e101ff3' \
    --endpoint-configuration '{"types": ["PRIVATE"]}' \
    --security-policy 'TLS_1_2' \
    --policy file://policy.json
```

policy.json 的內容：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "execute-api:Invoke",
            "Resource": [
                "execute-api:/*"
            ]
        },
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "execute-api:Invoke",
            "Resource": [
                "execute-api:/*"
            ],
            "Condition" : {
                "StringNotEquals": {
                    "aws:SourceVpce": "vpce-abcd1234efg"
                }
            }
        }
    ]
}
```

輸出：

```
{
    "domainName": "my.private.domain.tld",
    "domainNameId": "abcd1234",
    "domainNameArn": "arn:aws:apigateway:us-east-1:012345678910:/domainnames/
my.private.domain.tld+abcd1234",
    "certificateArn": "arn:aws:acm:us-east-1:012345678910:certificate/fb1b9770-
a305-495d-aefb-27e5e101ff3",
    "certificateUploadDate": "2024-09-10T10:31:20-07:00",
    "endpointConfiguration": {
        "types": [
            "PRIVATE"
        ]
```

```
    },
    "domainNameStatus": "AVAILABLE",
    "securityPolicy": "TLS_1_2",
    "policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Principal\":\"*\",\"Action\":\"execute-api:Invoke\",\"Resource\":
\"arn:aws:execute-api:us-east-1:012345678910:/domainnames/my.private.domain.tld
+abcd1234\"},{\"Effect\":\"Deny\",\"Principal\":\"*\",\"Action\":\"execute-
api:Invoke\",\"Resource\":\"arn:aws:execute-api:us-east-1:012345678910:/domainnames/
my.private.domain.tld+abcd1234\",\"Condition\":{\"StringNotEquals\":{\"aws:SourceVpc
\":\"vpc-1a2b3c4d\"}}}]}"
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的 API Gateway 中公有 REST API 的自訂網域名稱。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDomainName。

`create-model`

以下程式碼範例顯示如何使用 `create-model`。

AWS CLI

為 API 建立模型

命令：

```
aws apigateway create-model --rest-api-id 1234123412 --name 'firstModel' --
description 'The First Model' --content-type 'application/json'  --schema
 '{ "$schema": "http://json-schema.org/draft-04/schema#", "title": "firstModel",
 "type": "object", "properties": { "firstProperty" : { "type": "object",
 "properties": { "key": { "type": "string" } } } } }'
```

輸出：

```
{
    "contentType": "application/json",
    "description": "The First Model",
    "name": "firstModel",
    "id": "2rzg0l",
    "schema": "{ \"$schema\": \"http://json-schema.org/draft-04/schema#\", \"title
\": \"firstModel\", \"type\": \"object\", \"properties\": { \"firstProperty
```

```
\" : { \"type\": \"object\", \"properties\": { \"key\": { \"type\": \"string
\" } } } } }"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateModel。

create-resource

以下程式碼範例顯示如何使用 create-resource。

AWS CLI

在 API 中建立資源

命令：

```
aws apigateway create-resource --rest-api-id 1234123412 --parent-id a1b2c3 --path-
part 'new-resource'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateResource。

create-rest-api

以下程式碼範例顯示如何使用 create-rest-api。

AWS CLI

建立 API

命令：

```
aws apigateway create-rest-api --name 'My First API' --description 'This is my first
 API'
```

從現有的 API 建立重複的 API

命令：

```
aws apigateway create-rest-api --name 'Copy of My First API' --description 'This is
 a copy of my first API' --clone-from 1234123412
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRestApi。

create-stage

以下程式碼範例顯示如何使用 create-stage。

AWS CLI

在將包含現有部署的 API 中建立階段

命令：

```
aws apigateway create-stage --rest-api-id 1234123412 --stage-name 'dev' --
description 'Development stage' --deployment-id a1b2c3
```

在 API 中建立階段，其中包含現有的部署和自訂階段變數

命令：

```
aws apigateway create-stage --rest-api-id 1234123412 --stage-name 'dev'
 --description 'Development stage' --deployment-id a1b2c3 --variables
 key='value',otherKey='otherValue'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateStage。

create-usage-plan-key

以下程式碼範例顯示如何使用 create-usage-plan-key。

AWS CLI

將現有的 API 金鑰與用量計劃建立關聯

命令：

```
aws apigateway create-usage-plan-key --usage-plan-id a1b2c3 --key-type "API_KEY" --
key-id 4vq3yryqm5
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateUsagePlanKey。

create-usage-plan

以下程式碼範例顯示如何使用 create-usage-plan。

AWS CLI

建立具有調節和配額限制的用量計劃，該限制會在當月開始時重設

命令：

```
aws apigateway create-usage-plan --name "New Usage Plan" --description "A new usage
 plan" --throttle burstLimit=10,rateLimit=5 --quota limit=500,offset=0,period=MONTH
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateUsagePlan。

delete-api-key

以下程式碼範例顯示如何使用 delete-api-key。

AWS CLI

刪除 API 金鑰

命令：

```
aws apigateway delete-api-key --api-key 8bklk8bl1k3sB38D9B3l0enyWT8c09B30lkq0blk
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteApiKey。

delete-authorizer

以下程式碼範例顯示如何使用 delete-authorizer。

AWS CLI

在 API 中刪除自訂授權方

命令：

```
aws apigateway delete-authorizer --rest-api-id 1234123412 --authorizer-id 7gkfbo
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAuthorizer。

delete-base-path-mapping

以下程式碼範例顯示如何使用 delete-base-path-mapping。

AWS CLI

刪除自訂網域名稱的基本路徑映射

命令：

```
aws apigateway delete-base-path-mapping --domain-name 'api.domain.tld' --base-path
 'dev'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBasePathMapping。

delete-client-certificate

以下程式碼範例顯示如何使用 delete-client-certificate。

AWS CLI

刪除用戶端憑證

命令：

```
aws apigateway delete-client-certificate --client-certificate-id a1b2c3
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteClientCertificate。

delete-deployment

以下程式碼範例顯示如何使用 delete-deployment。

AWS CLI

刪除 API 中的部署

命令：

```
aws apigateway delete-deployment --rest-api-id 1234123412 --deployment-id a1b2c3
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDeployment。

delete-domain-name-access-association

以下程式碼範例顯示如何使用 delete-domain-name-access-association。

AWS CLI

刪除網域名稱存取關聯

下列delete-domain-name-access-association範例會刪除私有自訂網域名稱和 VPC 端點之間的網域名稱存取關聯。

```
aws apigateway delete-domain-name-access-association \
    --domain-name-access-association-arn arn:aws:apigateway:us-west-2:012345678910:/domainnameaccessassociations/domainname/my.private.domain.tld/vpcesource/vpce-abcd1234efg
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的 API Gateway 中私有 API 的自訂網域名稱。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDomainNameAccessAssociation。

delete-domain-name

以下程式碼範例顯示如何使用 delete-domain-name。

AWS CLI

刪除自訂網域名稱

命令：

```
aws apigateway delete-domain-name --domain-name 'api.domain.tld'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDomainName。

delete-integration-response

以下程式碼範例顯示如何使用 delete-integration-response。

AWS CLI

在 API 中刪除指定資源、方法和狀態碼的整合回應

命令：

```
aws apigateway delete-integration-response --rest-api-id 1234123412 --resource-
id a1b2c3 --http-method GET --status-code 200
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteIntegrationResponse。

delete-integration

以下程式碼範例顯示如何使用 delete-integration。

AWS CLI

在 API 中刪除指定資源和方法的整合

命令：

```
aws apigateway delete-integration --rest-api-id 1234123412 --resource-id a1b2c3 --
http-method GET
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteIntegration。

delete-method-response

以下程式碼範例顯示如何使用 delete-method-response。

AWS CLI

在 API 中刪除指定資源、方法和狀態碼的方法回應

命令：

```
aws apigateway delete-method-response --rest-api-id 1234123412 --resource-id a1b2c3
 --http-method GET --status-code 200
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteMethodResponse。

delete-method

以下程式碼範例顯示如何使用 delete-method。

AWS CLI

刪除 API 中指定資源的方法

命令：

```
aws apigateway delete-method --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteMethod。

delete-model

以下程式碼範例顯示如何使用 delete-model。

AWS CLI

在指定的 API 中刪除模型

命令：

```
aws apigateway delete-model --rest-api-id 1234123412 --model-name 'customModel'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteModel。

delete-resource

以下程式碼範例顯示如何使用 delete-resource。

AWS CLI

刪除 API 中的資源

命令：

```
aws apigateway delete-resource --rest-api-id 1234123412 --resource-id a1b2c3
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteResource。

delete-rest-api

以下程式碼範例顯示如何使用 delete-rest-api。

AWS CLI

刪除 API

命令：

```
aws apigateway delete-rest-api --rest-api-id 1234123412
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRestApi。

delete-stage

以下程式碼範例顯示如何使用 delete-stage。

AWS CLI

刪除 API 中的階段

命令：

```
aws apigateway delete-stage --rest-api-id 1234123412 --stage-name 'dev'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteStage。

delete-usage-plan-key

以下程式碼範例顯示如何使用 delete-usage-plan-key。

AWS CLI

從用量計劃中移除 API 金鑰

命令：

```
aws apigateway delete-usage-plan-key --usage-plan-id a1b2c3 --key-
id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteUsagePlanKey。

delete-usage-plan

以下程式碼範例顯示如何使用 delete-usage-plan。

AWS CLI

刪除用量計劃

命令：

```
aws apigateway delete-usage-plan --usage-plan-id a1b2c3
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteUsagePlan。

flush-stage-authorizers-cache

以下程式碼範例顯示如何使用 flush-stage-authorizers-cache。

AWS CLI

清除階段上的所有授權方快取項目

命令：

```
aws apigateway flush-stage-authorizers-cache --rest-api-id 1234123412 --stage-
name dev
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 FlushStageAuthorizersCache。

flush-stage-cache

以下程式碼範例顯示如何使用 flush-stage-cache。

AWS CLI

排清 API 階段的快取

下列flush-stage-cache範例會排清階段的快取。

```
aws apigateway flush-stage-cache \
    --rest-api-id 1234123412 \
    --stage-name dev
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的排清 API Gateway 中的 API 階段快取。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 FlushStageCache。

generate-client-certificate

以下程式碼範例顯示如何使用 generate-client-certificate。

AWS CLI

建立用戶端 SSL 憑證

命令：

```
aws apigateway generate-client-certificate --description 'My First Client
 Certificate'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GenerateClientCertificate。

get-account

以下程式碼範例顯示如何使用 get-account。

AWS CLI

取得 API Gateway 帳戶設定

命令：

```
aws apigateway get-account
```

輸出：

```
{
    "cloudwatchRoleArn": "arn:aws:iam::123412341234:role/
APIGatewayToCloudWatchLogsRole",
    "throttleSettings": {
        "rateLimit": 500.0,
        "burstLimit": 1000
    }
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAccount。

get-api-key

以下程式碼範例顯示如何使用 get-api-key。

AWS CLI

取得特定 API 金鑰的相關資訊

命令：

```
aws apigateway get-api-key --api-key 8bklk8bl1k3sB38D9B3l0enyWT8c09B30lkq0blk
```

輸出：

```
{
    "description": "My first key",
    "enabled": true,
    "stageKeys": [
        "a1b2c3d4e5/dev",
        "e5d4c3b2a1/dev"
    ],
    "lastUpdatedDate": 1456184515,
    "createdDate": 1456184452,
    "id": "8bklk8bl1k3sB38D9B3l0enyWT8c09B30lkq0blk",
    "name": "My key"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetApiKey。

get-api-keys

以下程式碼範例顯示如何使用 get-api-keys。

AWS CLI

取得 API 金鑰清單

命令：

```
aws apigateway get-api-keys
```

輸出：

```
{
    "items": [
        {
            "description": "My first key",
            "enabled": true,
            "stageKeys": [
                "a1b2c3d4e5/dev",
                "e5d4c3b2a1/dev"
            ],
            "lastUpdatedDate": 1456184515,
            "createdDate": 1456184452,
            "id": "8bklk8bl1k3sB38D9B3l0enyWT8c09B30lkq0blk",
            "name": "My key"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetApiKeys。

get-authorizer

以下程式碼範例顯示如何使用 get-authorizer。

AWS CLI

取得每個 API 授權方的 API Gateway 設定

命令：

```
aws apigateway get-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3
```

輸出：

```
{
    "authorizerResultTtlInSeconds": 300,
    "name": "MyAuthorizer",
    "type": "TOKEN",
```

```
    "identitySource": "method.request.header.Authorization",
    "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:authorizer_function/invocations",
    "id": "gfi4n3"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAuthorizer。

get-authorizers

以下程式碼範例顯示如何使用 get-authorizers。

AWS CLI

取得 REST API 的授權方清單

命令：

```
aws apigateway get-authorizers --rest-api-id 1234123412
```

輸出：

```
{
    "items": [
        {
            "name": "MyAuthorizer",
            "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/
functions/arn:aws:lambda:us-west-2:123412341234:function:My_Authorizer_Function/
invocations",
            "authorizerResultTtlInSeconds": 300,
            "identitySource": "method.request.header.Authorization",
            "type": "TOKEN",
            "id": "gfi4n3"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAuthorizers。

get-base-path-mapping

以下程式碼範例顯示如何使用 get-base-path-mapping。

AWS CLI

取得自訂網域名稱的基本路徑映射

命令：

```
aws apigateway get-base-path-mapping --domain-name subdomain.domain.tld --base-path v1
```

輸出：

```
{
    "basePath": "v1",
    "restApiId": "1234w4321e",
    "stage": "api"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBasePathMapping。

get-base-path-mappings

以下程式碼範例顯示如何使用 get-base-path-mappings。

AWS CLI

取得自訂網域名稱的基本路徑映射

命令：

```
aws apigateway get-base-path-mappings --domain-name subdomain.domain.tld
```

輸出：

```
{
    "items": [
        {
            "basePath": "(none)",
            "restApiId": "1234w4321e",
            "stage": "dev"
        },
        {
            "basePath": "v1",
```

```
            "restApiId": "1234w4321e",
            "stage": "api"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBasePathMappings。

get-client-certificate

以下程式碼範例顯示如何使用 get-client-certificate。

AWS CLI

取得用戶端憑證

命令：

```
aws apigateway get-client-certificate --client-certificate-id a1b2c3
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetClientCertificate。

get-client-certificates

以下程式碼範例顯示如何使用 get-client-certificates。

AWS CLI

取得用戶端憑證清單

命令：

```
aws apigateway get-client-certificates
```

輸出：

```
{
    "items": [
        {
            "pemEncodedCertificate": "-----BEGIN CERTIFICATE----- <certificate
 content> -----END CERTIFICATE-----",
```

```
            "clientCertificateId": "a1b2c3",
            "expirationDate": 1483556561,
            "description": "My Client Certificate",
            "createdDate": 1452020561
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetClientCertificates。

get-deployment

以下程式碼範例顯示如何使用 get-deployment。

AWS CLI

取得部署的相關資訊

命令：

```
aws apigateway get-deployment --rest-api-id 1234123412 --deployment-id ztt4m2
```

輸出：

```
{
    "description": "myDeployment",
    "id": "ztt4m2",
    "createdDate": 1455218022
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDeployment。

get-deployments

以下程式碼範例顯示如何使用 get-deployments。

AWS CLI

若要取得 REST API 的部署清單

命令：

```
aws apigateway get-deployments --rest-api-id 1234123412
```

輸出：

```
{
    "items": [
        {
            "createdDate": 1453797217,
            "id": "0a2b4c",
            "description": "Deployed my API for the first time"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDeployments。

get-domain-name-access-associations

以下程式碼範例顯示如何使用 get-domain-name-access-associations。

AWS CLI

範例 1：列出所有網域名稱存取關聯

下列get-domain-name-access-associations範例列出所有網域名稱存取關聯。

```
aws apigateway get-domain-name-access-associations
```

輸出：

```
{
    "items": [
        {
        "domainNameAccessAssociationArn": "arn:aws:apigateway:us-
west-2:012345678910:/domainnameaccessassociations/domainname/my.private.domain.tld/
vpcesource/vpce-abcd1234efg
        "accessAssociationSource": "vpce-abcd1234efg",
        "accessAssociationSourceType": "VPCE",
        "domainNameArn" : "arn:aws:apigateway:us-west-2:111122223333:/domainnames/
private.example.com+abcd1234"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的 API Gateway 中私有 API 的自訂網域名稱。 Amazon API Gateway

範例 2：列出此 AWS 帳戶擁有的所有網域名稱存取關聯

下列get-domain-name-access-associations範例列出目前 AWS 帳戶擁有的所有網域名稱存取關聯。

```
aws apigateway get-domain-name-access-associations \
    --resource-owner SELF
```

輸出：

```
{
    "items": [
        {
        "domainNameAccessAssociationArn": "arn:aws:apigateway:us-
west-2:012345678910:/domainnameaccessassociations/domainname/my.private.domain.tld/
vpcesource/vpce-abcd1234efg
        "accessAssociationSource": "vpce-abcd1234efg",
        "accessAssociationSourceType": "VPCE",
        "domainNameArn" : "arn:aws:apigateway:us-west-2:111122223333:/domainnames/
private.example.com+abcd1234"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的 API Gateway 中私有 API 的自訂網域名稱。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDomainNameAccessAssociations。

get-domain-name

以下程式碼範例顯示如何使用 get-domain-name。

AWS CLI

範例 1：取得公有自訂網域名稱的相關資訊

下列get-domain-name範例取得公有自訂網域名稱的相關資訊。

```
aws apigateway get-domain-name \
    --domain-name api.domain.tld
```

輸出：

```
{
    "domainName": "api.domain.tld",
    "distributionDomainName": "d1a2f3a4c5o6d.cloudfront.net",
    "certificateName": "uploadedCertificate",
    "certificateUploadDate": 1462565487
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的 API Gateway 中公有 REST API 的自訂網域名稱。 Amazon API Gateway

範例 2：取得私有自訂網域名稱的相關資訊

下列get-domain-name範例取得私有自訂網域名稱的相關資訊。

```
aws apigateway get-domain-name \
    --domain-name api.private.domain.tld \
    --domain-name-id abcd1234
```

輸出：

```
{
    "domainName": "my.private.domain.tld",
    "domainNameId": "abcd1234",
    "domainNameArn": "arn:aws:apigateway:us-east-1:012345678910:/domainnames/
my.private.domain.tld+abcd1234",
    "certificateArn": "arn:aws:acm:us-east-1:012345678910:certificate/fb1b9770-
a305-495d-aefb-27e5e101ff3",
    "certificateUploadDate": "2024-09-10T10:31:20-07:00",
    "endpointConfiguration": {
        "types": [
            "PRIVATE"
        ]
    },
    "domainNameStatus": "AVAILABLE",
```

```
    "securityPolicy": "TLS_1_2",
    "policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Principal\":\"*\",\"Action\":\"execute-api:Invoke\",\"Resource\":
\"arn:aws:execute-api:us-east-1:012345678910:/domainnames/my.private.domain.tld
+abcd1234\"},{\"Effect\":\"Deny\",\"Principal\":\"*\",\"Action\":\"execute-
api:Invoke\",\"Resource\":\"arn:aws:execute-api:us-east-1:012345678910:/domainnames/
my.private.domain.tld+abcd1234\",\"Condition\":{\"StringNotEquals\":{\"aws:SourceVpc
\":\"vpc-1a2b3c4d\"}}}]}"
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的 API Gateway 中公有 REST API 的自訂網域名稱。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDomainName。

get-domain-names

以下程式碼範例顯示如何使用 get-domain-names。

AWS CLI

範例 1：取得自訂網域名稱的清單

下列get-domain-names命令會取得網域名稱的清單。

```
aws apigateway get-domain-names
```

輸出：

```
{
    "items": [
        {
            "distributionDomainName": "d9511k3l09bkd.cloudfront.net",
            "certificateUploadDate": 1452812505,
            "certificateName": "my_custom_domain-certificate",
            "domainName": "subdomain.domain.tld"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的 API Gateway 中私有 API 的自訂網域名稱。 Amazon API Gateway

範例 2：取得此 AWS 帳戶擁有的自訂網域名稱清單

下列get-domain-names命令會取得此 AWS 帳戶擁有的網域名稱清單。

```
aws apigateway get-domain-names \
    --resource-owner SELF
```

輸出：

```
{
    "items": [
        {
            "domainName": "my.domain.tld",
            "domainNameArn": "arn:aws:apigateway:us-east-1::/domainnames/
my.private.domain.tld",
            "certificateUploadDate": "2024-08-15T17:02:55-07:00",
            "regionalDomainName": "d-abcd1234.execute-api.us-east-1.amazonaws.com",
            "regionalHostedZoneId": "Z1UJRXOUMOOFQ8",
            "regionalCertificateArn": "arn:aws:acm:us-
east-1:012345678910:certificate/fb1b9770-a305-495d-aefb-27e5e101ff3",
            "endpointConfiguration": {
                "types": [
                    "REGIONAL"
                ]
            },
            "domainNameStatus": "AVAILABLE",
            "securityPolicy": "TLS_1_2"
        },
        {
            "domainName": "my.private.domain.tld",
            "domainNameId": "abcd1234",
            "domainNameArn": "arn:aws:apigateway:us-east-1:012345678910:/
domainnames/my.private.domain.tld+abcd1234",
            "certificateArn": "arn:aws:acm:us-east-1:012345678910:certificate/
fb1b9770-a305-495d-aefb-27e5e101ff3",
            "certificateUploadDate": "2024-11-26T11:44:40-08:00",
            "endpointConfiguration": {
                "types": [
                    "PRIVATE"
                ]
            },
            "domainNameStatus": "AVAILABLE",
            "securityPolicy": "TLS_1_2"
```

```
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的 API Gateway 中私有 API 的自訂網域名稱。 Amazon API Gateway

範例 3：取得您可以建立網域名稱存取關聯的其他 AWS 帳戶所擁有的自訂網域名稱清單。

下列get-domain-names命令會取得您具有存取權的其他 AWS 帳戶所擁有的網域名稱清單，以建立網域名稱存取關聯。

```
aws apigateway get-domain-names \
    --resource-owner OTHER_ACCOUNTS
```

輸出：

```
{
    "items": [
        {
            "domainName": "my.private.domain.tld",
            "domainNameId": "abcd1234",
            "domainNameArn": "arn:aws:apigateway:us-east-1:012345678910:/
domainnames/my.private.domain.tld+abcd1234"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的 API Gateway 中私有 API 的自訂網域名稱。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDomainNames。

get-export

以下程式碼範例顯示如何使用 get-export。

AWS CLI

取得階段的 JSON Swagger 範本

命令：

```
aws apigateway get-export --rest-api-id a1b2c3d4e5 --stage-name dev --export-
type swagger /path/to/filename.json
```

取得階段的 JSON Swagger 範本 + API Gateway Extentions

命令：

```
aws apigateway get-export --parameters extensions='integrations' --rest-api-
id a1b2c3d4e5 --stage-name dev --export-type swagger /path/to/filename.json
```

取得階段的 JSON Swagger 範本 + Postman 延伸

命令：

```
aws apigateway get-export --parameters extensions='postman' --rest-api-id a1b2c3d4e5
 --stage-name dev --export-type swagger /path/to/filename.json
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetExport。

get-integration-response

以下程式碼範例顯示如何使用 get-integration-response。

AWS CLI

取得 REST API 資源下定義之 HTTP 方法的整合回應組態

命令：

```
aws apigateway get-integration-response --rest-api-id 1234123412 --resource-
id y9h6rt --http-method GET --status-code 200
```

輸出：

```
{
    "statusCode": "200",
    "responseTemplates": {
        "application/json": null
    }
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIntegrationResponse。

get-integration

以下程式碼範例顯示如何使用 get-integration。

AWS CLI

取得 REST API 資源下定義之 HTTP 方法的整合組態

命令：

```
aws apigateway get-integration --rest-api-id 1234123412 --resource-id y9h6rt --http-
method GET
```

輸出：

```
{
    "httpMethod": "POST",
    "integrationResponses": {
        "200": {
            "responseTemplates": {
                "application/json": null
            },
            "statusCode": "200"
        }
    },
    "cacheKeyParameters": [],
    "type": "AWS",
    "uri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:My_Function/invocations",
    "cacheNamespace": "y9h6rt"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIntegration。

get-method-response

以下程式碼範例顯示如何使用 get-method-response。

AWS CLI

取得 REST API 資源下定義之 HTTP 方法的方法回應資源組態

命令：

```
aws apigateway get-method-response --rest-api-id 1234123412 --resource-id y9h6rt --
http-method GET --status-code 200
```

輸出：

```
{
    "responseModels": {
        "application/json": "Empty"
    },
    "statusCode": "200"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMethodResponse。

get-method

以下程式碼範例顯示如何使用 get-method。

AWS CLI

取得 REST API 資源下定義之 HTTP 方法的方法資源組態

命令：

```
aws apigateway get-method --rest-api-id 1234123412 --resource-id y9h6rt --http-
method GET
```

輸出：

```
{
    "apiKeyRequired": false,
    "httpMethod": "GET",
    "methodIntegration": {
        "integrationResponses": {
            "200": {
```

```
                "responseTemplates": {
                    "application/json": null
                },
                "statusCode": "200"
            }
        },
        "cacheKeyParameters": [],
        "uri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:My_Function/invocations",
        "httpMethod": "POST",
        "cacheNamespace": "y9h6rt",
        "type": "AWS"
    },
    "requestParameters": {},
    "methodResponses": {
        "200": {
            "responseModels": {
                "application/json": "Empty"
            },
            "statusCode": "200"
        }
    },
    "authorizationType": "NONE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMethod。

get-model-template

以下程式碼範例顯示如何使用 get-model-template。

AWS CLI

取得 REST API 下定義的模型映射範本

命令：

```
aws apigateway get-model-template --rest-api-id 1234123412 --model-name Empty
```

輸出：

```
{
```

```
    "value": "#set($inputRoot = $input.path('$'))\n{ }"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetModelTemplate。

get-model

以下程式碼範例顯示如何使用 get-model。

AWS CLI

若要取得 REST API 下定義的模型組態

命令：

```
aws apigateway get-model --rest-api-id 1234123412 --model-name Empty
```

輸出：

```
{
    "contentType": "application/json",
    "description": "This is a default empty schema model",
    "name": "Empty",
    "id": "etd5w5",
    "schema": "{\n  \"$schema\": \"http://json-schema.org/draft-04/schema#\",\n
 \"title\" : \"Empty Schema\",\n  \"type\" : \"object\"\n}"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetModel。

get-models

以下程式碼範例顯示如何使用 get-models。

AWS CLI

取得 REST API 的模型清單

命令：

```
aws apigateway get-models --rest-api-id 1234123412
```

輸出：

```
{
    "items": [
        {
            "description": "This is a default error schema model",
            "schema": "{\n  \"$schema\" : \"http://json-schema.org/draft-04/schema#
\",\n  \"title\" : \"Error Schema\",\n  \"type\" : \"object\",\n  \"properties\" :
 {\n    \"message\" : { \"type\" : \"string\" }\n  }\n}",
            "contentType": "application/json",
            "id": "7tpbze",
            "name": "Error"
        },
        {
            "description": "This is a default empty schema model",
            "schema": "{\n  \"$schema\": \"http://json-schema.org/draft-04/schema#
\",\n  \"title\" : \"Empty Schema\",\n  \"type\" : \"object\"\n}",
            "contentType": "application/json",
            "id": "etd5w5",
            "name": "Empty"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetModels。

get-resource

以下程式碼範例顯示如何使用 get-resource。

AWS CLI

取得資源的相關資訊

命令：

```
aws apigateway get-resource --rest-api-id 1234123412 --resource-id zwo0y3
```

輸出：

```
{
```

```
    "path": "/path",
    "pathPart": "path",
    "id": "zwo0y3",
    "parentId": "uyokt6ij2g"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResource。

get-resources

以下程式碼範例顯示如何使用 get-resources。

AWS CLI

取得 REST API 的資源清單

命令：

```
aws apigateway get-resources --rest-api-id 1234123412
```

輸出：

```
{
    "items": [
        {
            "path": "/resource/subresource",
            "resourceMethods": {
                "POST": {}
            },
            "id": "024ace",
            "pathPart": "subresource",
            "parentId": "ai5b02"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResources。

get-rest-api

以下程式碼範例顯示如何使用 get-rest-api。

AWS CLI

取得 API 的相關資訊

命令：

```
aws apigateway get-rest-api --rest-api-id 1234123412
```

輸出：

```
{
    "name": "myAPI",
    "id": "o1y243m4f5",
    "createdDate": 1453416433
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRestApi。

get-rest-apis

以下程式碼範例顯示如何使用 get-rest-apis。

AWS CLI

取得 REST APIs的清單

命令：

```
aws apigateway get-rest-apis
```

輸出：

```
{
    "items": [
        {
            "createdDate": 1438884790,
            "id": "12s44z21rb",
            "name": "My First API"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRestApis。

get-sdk

以下程式碼範例顯示如何使用 get-sdk。

AWS CLI

取得 REST API 階段的 Android 開發套件

命令：

```
aws apigateway get-sdk --rest-api-id 1234123412 --stage-name dev --sdk-type android
 --parameters
 groupId='com.mycompany',invokerPackage='com.mycompany.clientsdk',artifactId='Mycompany-
client',artifactVersion='1.0.0' /path/to/android_sdk.zip
```

輸出：

```
{
    "contentType": "application/octet-stream",
    "contentDisposition": "attachment; filename=\"android_2016-02-22_23-52Z.zip\""
}
```

取得 REST API 階段的 IOS 開發套件

命令：

```
aws apigateway get-sdk --rest-api-id 1234123412 --stage-name dev --sdk-
type objectivec --parameters classPrefix='myprefix' /path/to/iOS_sdk.zip
```

輸出：

```
{
    "contentType": "application/octet-stream",
    "contentDisposition": "attachment; filename=\"objectivec_2016-02-22_23-52Z.zip
\""
}
```

取得 REST API 階段的 Javascript SDK

命令：

```
aws apigateway get-sdk --rest-api-id 1234123412 --stage-name dev --sdk-
type javascript /path/to/javascript_sdk.zip
```

輸出：

```
{
    "contentType": "application/octet-stream",
    "contentDisposition": "attachment; filename=\"javascript_2016-02-22_23-52Z.zip
\""
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSdk。

get-stage

以下程式碼範例顯示如何使用 get-stage。

AWS CLI

取得 API 階段的相關資訊

命令：

```
aws apigateway get-stage --rest-api-id 1234123412 --stage-name dev
```

輸出：

```
{
    "stageName": "dev",
    "cacheClusterSize": "0.5",
    "cacheClusterEnabled": false,
    "cacheClusterStatus": "NOT_AVAILABLE",
    "deploymentId": "rbh1fj",
    "lastUpdatedDate": 1466802961,
    "createdDate": 1460682074,
    "methodSettings": {
        "*/*": {
            "cacheTtlInSeconds": 300,
```

```
            "loggingLevel": "INFO",
            "dataTraceEnabled": false,
            "metricsEnabled": true,
            "unauthorizedCacheControlHeaderStrategy":
 "SUCCEED_WITH_RESPONSE_HEADER",
            "throttlingRateLimit": 500.0,
            "cacheDataEncrypted": false,
            "cachingEnabled": false,
            "throttlingBurstLimit": 1000,
            "requireAuthorizationForCacheControl": true
        },
        "~1resource/GET": {
            "cacheTtlInSeconds": 300,
            "loggingLevel": "INFO",
            "dataTraceEnabled": false,
            "metricsEnabled": true,
            "unauthorizedCacheControlHeaderStrategy":
 "SUCCEED_WITH_RESPONSE_HEADER",
            "throttlingRateLimit": 500.0,
            "cacheDataEncrypted": false,
            "cachingEnabled": false,
            "throttlingBurstLimit": 1000,
            "requireAuthorizationForCacheControl": true
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetStage。

get-stages

以下程式碼範例顯示如何使用 get-stages。

AWS CLI

取得 REST API 的階段清單

命令：

```
aws apigateway get-stages --rest-api-id 1234123412
```

輸出：

```
{
    "item": [
        {
            "stageName": "dev",
            "cacheClusterSize": "0.5",
            "cacheClusterEnabled": true,
            "cacheClusterStatus": "AVAILABLE",
            "deploymentId": "123h64",
            "lastUpdatedDate": 1456185138,
            "createdDate": 1453589092,
            "methodSettings": {
                "~1resource~1subresource/POST": {
                    "cacheTtlInSeconds": 300,
                    "loggingLevel": "INFO",
                    "dataTraceEnabled": true,
                    "metricsEnabled": true,
                    "throttlingRateLimit": 500.0,
                    "cacheDataEncrypted": false,
                    "cachingEnabled": false,
                    "throttlingBurstLimit": 1000
                }
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetStages。

get-usage-plan-key

以下程式碼範例顯示如何使用 get-usage-plan-key。

AWS CLI

取得與用量計劃相關聯之 API 金鑰的詳細資訊

命令：

```
aws apigateway get-usage-plan-key --usage-plan-id a1b2c3 --key-
id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetUsagePlanKey。

get-usage-plan-keys

以下程式碼範例顯示如何使用 get-usage-plan-keys。

AWS CLI

取得與用量計劃相關聯的 API 金鑰清單

命令：

```
aws apigateway get-usage-plan-keys --usage-plan-id a1b2c3
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetUsagePlanKeys。

get-usage-plan

以下程式碼範例顯示如何使用 get-usage-plan。

AWS CLI

取得用量計劃的詳細資訊

命令：

```
aws apigateway get-usage-plan --usage-plan-id a1b2c3
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetUsagePlan。

get-usage-plans

以下程式碼範例顯示如何使用 get-usage-plans。

AWS CLI

取得所有用量計劃的詳細資訊

命令：

```
aws apigateway get-usage-plans
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetUsagePlans。

get-usage

以下程式碼範例顯示如何使用 get-usage。

AWS CLI

取得用量計劃的用量詳細資訊

命令：

```
aws apigateway get-usage --usage-plan-id a1b2c3 --start-date "2016-08-16" --end-
date "2016-08-17"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetUsage。

import-rest-api

以下程式碼範例顯示如何使用 import-rest-api。

AWS CLI

匯入 Swagger 範本並建立 API

命令：

```
aws apigateway import-rest-api --body 'file:///path/to/API_Swagger_template.json'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ImportRestApi。

put-integration-response

以下程式碼範例顯示如何使用 put-integration-response。

AWS CLI

建立整合回應做為預設回應，並定義映射範本

命令：

```
aws apigateway put-integration-response --rest-api-id 1234123412 --resource-
id a1b2c3 --http-method GET --status-code 200 --selection-pattern "" --response-
templates '{"application/json": "{\"json\": \"template\"}"}'
```

建立具有 400 regex 和靜態定義標頭值的整合回應

命令：

```
aws apigateway put-integration-response --rest-api-id 1234123412 --resource-
id a1b2c3 --http-method GET --status-code 400 --selection-pattern 400 --response-
parameters '{"method.response.header.custom-header": "'"'"'custom-value'"'"'"}'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutIntegrationResponse。

put-integration

以下程式碼範例顯示如何使用 put-integration。

AWS CLI

建立 MOCK 整合請求

命令：

```
aws apigateway put-integration --rest-api-id 1234123412 --resource-id a1b2c3 --http-
method GET --type MOCK --request-templates '{ "application/json": "{\"statusCode\":
 200}" }'
```

建立 HTTP 整合請求

命令：

```
aws apigateway put-integration --rest-api-id 1234123412 --resource-id a1b2c3 --http-
method GET --type HTTP --integration-http-method GET --uri 'https://domain.tld/path'
```

使用 Lambda 函數端點建立 AWS 整合請求

命令：

```
aws apigateway put-integration --rest-api-id 1234123412 --resource-
id a1b2c3 --http-method GET --type AWS --integration-http-method POST --uri
 'arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/arn:aws:lambda:us-
west-2:123412341234:function:function_name/invocations'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutIntegration。

put-method-response

以下程式碼範例顯示如何使用 put-method-response。

AWS CLI

在具有自訂方法回應標頭的指定狀態碼下建立方法回應

命令：

```
aws apigateway put-method-response --rest-api-id 1234123412 --
resource-id a1b2c3 --http-method GET --status-code 400 --response-
parameters "method.response.header.custom-header=false"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutMethodResponse。

put-method

以下程式碼範例顯示如何使用 put-method。

AWS CLI

在無授權、無 API 金鑰和自訂方法請求標頭的 API 中建立資源的方法

命令：

```
aws apigateway put-method --rest-api-id 1234123412 --resource-id a1b2c3 --
http-method PUT --authorization-type "NONE" --no-api-key-required --request-
parameters "method.request.header.custom-header=false"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutMethod。

put-rest-api

以下程式碼範例顯示如何使用 put-rest-api。

AWS CLI

使用 Swagger 範本覆寫現有的 API

命令：

```
aws apigateway put-rest-api --rest-api-id 1234123412 --mode overwrite --body
 'fileb:///path/to/API_Swagger_template.json'
```

將 Swagger 範本合併到現有的 API

命令：

```
aws apigateway put-rest-api --rest-api-id 1234123412 --mode merge --body 'fileb:///
path/to/API_Swagger_template.json'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutRestApi。

reject-domain-name-access-association

以下程式碼範例顯示如何使用 reject-domain-name-access-association。

AWS CLI

拒絕網域名稱存取關聯

下列reject-domain-name-access-association範例會拒絕私有自訂網域名稱和 VPC 端點之間的網域名稱存取關聯。

```
aws apigateway reject-domain-name-access-association \
    --domain-name-access-association-arn arn:aws:apigateway:us-west-2:012345678910:/
domainnameaccessassociations/domainname/my.private.domain.tld/vpcesource/vpce-
abcd1234efg \
    --domain-name-arn arn:aws:apigateway:us-east-1:012345678910:/domainnames/
my.private.domain.tld+abcd1234
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的 API Gateway 中私有 API 的自訂網域名稱。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RejectDomainNameAccessAssociation。

test-invoke-authorizer

以下程式碼範例顯示如何使用 test-invoke-authorizer。

AWS CLI

測試呼叫請求給自訂授權方，包括所需的標頭和值

命令：

```
aws apigateway test-invoke-authorizer --rest-api-id 1234123412 --authorizer-
id 5yid1t --headers Authorization='Value'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TestInvokeAuthorizer。

test-invoke-method

以下程式碼範例顯示如何使用 test-invoke-method。

AWS CLI

透過提出 GET 請求，測試在 API 中叫用根資源

命令：

```
aws apigateway test-invoke-method --rest-api-id 1234123412 --resource-id avl5sg8fw8
 --http-method GET --path-with-query-string '/'
```

使用指定的路徑參數值提出 GET 請求，以測試在 API 中叫用子資源

命令：

```
aws apigateway test-invoke-method --rest-api-id 1234123412 --resource-id 3gapai --
http-method GET --path-with-query-string '/pets/1'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TestInvokeMethod。

update-account

以下程式碼範例顯示如何使用 update-account。

AWS CLI

變更用於記錄至 CloudWatch Logs 的 IAM 角色 ARN

命令：

```
aws apigateway update-account --patch-operations op='replace',path='/
cloudwatchRoleArn',value='arn:aws:iam::123412341234:role/APIGatewayToCloudWatchLogs'
```

輸出：

```
{
    "cloudwatchRoleArn": "arn:aws:iam::123412341234:role/
APIGatewayToCloudWatchLogs",
    "throttleSettings": {
        "rateLimit": 1000.0,
        "burstLimit": 2000
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAccount。

update-api-key

以下程式碼範例顯示如何使用 update-api-key。

AWS CLI

變更 API 金鑰的名稱

命令：

```
aws apigateway update-api-key --api-key sNvjQDMReA1eEQPNAW8r37XsU2rDD7fc7m2SiMnu --
patch-operations op='replace',path='/name',value='newName'
```

輸出：

```
{
    "description": "currentDescription",
    "enabled": true,
    "stageKeys": [
        "41t2j324r5/dev"
    ],
    "lastUpdatedDate": 1470086052,
    "createdDate": 1445460347,
    "id": "sNvjQDMReA1vEQPNzW8r3dXsU2rrD7fcjm2SiMnu",
    "name": "newName"
```

```
}
```

停用 API 金鑰

命令：

```
aws apigateway update-api-key --api-key sNvjQDMReA1eEQPNAW8r37XsU2rDD7fc7m2SiMnu --
patch-operations op='replace',path='/enabled',value='false'
```

輸出：

```
{
    "description": "currentDescription",
    "enabled": false,
    "stageKeys": [
        "41t2j324r5/dev"
    ],
    "lastUpdatedDate": 1470086052,
    "createdDate": 1445460347,
    "id": "sNvjQDMReA1vEQPNzW8r3dXsU2rrD7fcjm2SiMnu",
    "name": "newName"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateApiKey。

update-authorizer

以下程式碼範例顯示如何使用 update-authorizer。

AWS CLI

變更自訂授權方的名稱

命令：

```
aws apigateway update-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3 --
patch-operations op='replace',path='/name',value='testAuthorizer'
```

輸出：

```
{
```

```
    "authType": "custom",
    "name": "testAuthorizer",
    "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:customAuthorizer/invocations",
    "authorizerResultTtlInSeconds": 300,
    "identitySource": "method.request.header.Authorization",
    "type": "TOKEN",
    "id": "gfi4n3"
}
```

變更自訂授權方調用的 Lambda 函數

命令：

```
aws apigateway update-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3 --
patch-operations op='replace',path='/authorizerUri',value='arn:aws:apigateway:us-
west-2:lambda:path/2015-03-31/functions/arn:aws:lambda:us-
west-2:123412341234:function:newAuthorizer/invocations'
```

輸出：

```
{
    "authType": "custom",
    "name": "testAuthorizer",
    "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:newAuthorizer/invocations",
    "authorizerResultTtlInSeconds": 300,
    "identitySource": "method.request.header.Authorization",
    "type": "TOKEN",
    "id": "gfi4n3"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAuthorizer。

update-base-path-mapping

以下程式碼範例顯示如何使用 update-base-path-mapping。

AWS CLI

變更自訂網域名稱的基本路徑

命令：

```
aws apigateway update-base-path-mapping --domain-name api.domain.tld --base-
path prod --patch-operations op='replace',path='/basePath',value='v1'
```

輸出：

```
{
    "basePath": "v1",
    "restApiId": "1234123412",
    "stage": "api"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateBasePathMapping。

update-client-certificate

以下程式碼範例顯示如何使用 update-client-certificate。

AWS CLI

更新用戶端憑證的描述

命令：

```
aws apigateway update-client-certificate --client-certificate-id a1b2c3 --patch-
operations op='replace',path='/description',value='My new description'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateClientCertificate。

update-deployment

以下程式碼範例顯示如何使用 update-deployment。

AWS CLI

變更部署的描述

命令：

```
aws apigateway update-deployment --rest-api-id 1234123412 --deployment-id ztt4m2 --
patch-operations op='replace',path='/description',value='newDescription'
```

輸出：

```
{
    "description": "newDescription",
    "id": "ztt4m2",
    "createdDate": 1455218022
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDeployment。

update-domain-name

以下程式碼範例顯示如何使用 update-domain-name。

AWS CLI

變更自訂網域名稱的憑證名稱

下列update-domain-name範例會變更自訂網域的憑證名稱。

```
aws apigateway update-domain-name \
    --domain-name api.domain.tld \
    --patch-operations op='replace',path='/certificateArn',value='arn:aws:acm:us-
west-2:111122223333:certificate/CERTEXAMPLE123EXAMPLE'
```

輸出：

```
{
    "domainName": "api.domain.tld",
    "distributionDomainName": "d123456789012.cloudfront.net",
    "certificateArn": "arn:aws:acm:us-west-2:111122223333:certificate/
CERTEXAMPLE123EXAMPLE",
    "certificateUploadDate": 1462565487
}
```

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的在 API Gateway 中設定 API 的自訂網域名稱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDomainName。

update-integration-response

以下程式碼範例顯示如何使用 update-integration-response。

AWS CLI

將整合回應標頭變更為具有 '*' 的靜態映射

命令：

```
aws apigateway update-integration-response --rest-api-id 1234123412 --
resource-id 3gapai --http-method GET --status-code 200 --patch-operations
 op='replace',path='/responseParameters/method.response.header.Access-Control-Allow-
Origin',value='"'"'*'"'"'
```

輸出：

```
{
    "statusCode": "200",
    "responseParameters": {
        "method.response.header.Access-Control-Allow-Origin": "'*'"
    }
}
```

移除整合回應標頭

命令：

```
aws apigateway update-integration-response --rest-api-id 1234123412 --resource-
id 3gapai --http-method GET --status-code 200 --patch-operations op='remove',path='/
responseParameters/method.response.header.Access-Control-Allow-Origin'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateIntegrationResponse。

update-integration

以下程式碼範例顯示如何使用 update-integration。

AWS CLI

新增以輸入傳遞設定的「內容類型：application/json」映射範本

命令：

```
aws apigateway update-integration \
    --rest-api-id a1b2c3d4e5 \
    --resource-id a1b2c3 \
    --http-method POST \
    --patch-operations "op='add',path='/requestTemplates/application~1json'"
```

更新（取代）以自訂範本設定的「內容類型：application/json」映射範本

命令：

```
aws apigateway update-integration \
    --rest-api-id a1b2c3d4e5 \
    --resource-id a1b2c3 \
    --http-method POST \
    --patch-operations "op='replace',path='/requestTemplates/
application~1json',value='{"example": "json"}'"
```

使用輸入傳遞更新（取代）與 'Content-Type： application/json' 相關聯的自訂範本

命令：

```
aws apigateway update-integration \
    --rest-api-id a1b2c3d4e5 \
    --resource-id a1b2c3 \
    --http-method POST \
    --patch-operations "op='replace',path='requestTemplates/application~1json'"
```

移除「Content-Type：application/json」映射範本

命令：

```
aws apigateway update-integration \
    --rest-api-id a1b2c3d4e5 \
    --resource-id a1b2c3 \
    --http-method POST \
    --patch-operations "op='remove',path='/requestTemplates/application~1json'"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateIntegration。

update-method-response

以下程式碼範例顯示如何使用 update-method-response。

AWS CLI

在方法中為 200 個回應建立新的方法回應標頭，並將其定義為不需要 （ 預設 ）

命令：

```
aws apigateway update-method-response --rest-api-id 1234123412 --resource-
id a1b2c3 --http-method GET --status-code 200 --patch-operations op="add",path="/
responseParameters/method.response.header.custom-header",value="false"
```

刪除方法中 200 回應的回應模型

命令：

```
aws apigateway update-method-response --rest-api-id 1234123412 --resource-id a1b2c3
 --http-method GET --status-code 200 --patch-operations op="remove",path="/
responseModels/application~1json"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateMethodResponse。

update-method

以下程式碼範例顯示如何使用 update-method。

AWS CLI

範例 1：修改方法以要求 API 金鑰

下列update-method範例會將 方法修改為需要 API 金鑰。

```
aws apigateway update-method \
    --rest-api-id 1234123412 \
    --resource-id a1b2c3 \
    --http-method GET \
```

```
    --patch-operations op="replace",path="/apiKeyRequired",value="true"
```

輸出：

```
{
    "httpMethod": "GET",
    "authorizationType": "NONE",
    "apiKeyRequired": true,
    "methodResponses": {
        "200": {
            "statusCode": "200",
            "responseModels": {}
        }
    },
    "methodIntegration": {
        "type": "AWS",
        "httpMethod": "POST",
        "uri": "arn:aws:apigateway:us-east-1:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-east-1:123456789111:function:hello-world/invocations",
        "passthroughBehavior": "WHEN_NO_MATCH",
        "contentHandling": "CONVERT_TO_TEXT",
        "timeoutInMillis": 29000,
        "cacheNamespace": "h7i8j9",
        "cacheKeyParameters": [],
        "integrationResponses": {
            "200": {
                "statusCode": "200",
                "responseTemplates": {}
            }
        }
    }
}
```

範例 2：修改方法以要求 IAM 授權

下列update-method範例會將 方法修改為需要 IAM 授權。

```
aws apigateway update-method \
    --rest-api-id 1234123412 \
    --resource-id a1b2c3 \
    --http-method GET \
    --patch-operations op="replace",path="/authorizationType",value="AWS_IAM"
```

輸出：

```
 {
    "httpMethod": "GET",
    "authorizationType": "AWS_IAM",
    "apiKeyRequired": false,
    "methodResponses": {
        "200": {
            "statusCode": "200",
            "responseModels": {}
        }
    },
    "methodIntegration": {
        "type": "AWS",
        "httpMethod": "POST",
        "uri": "arn:aws:apigateway:us-east-1:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-east-1:123456789111:function:hello-world/invocations",
        "passthroughBehavior": "WHEN_NO_MATCH",
        "contentHandling": "CONVERT_TO_TEXT",
        "timeoutInMillis": 29000,
        "cacheNamespace": "h7i8j9",
        "cacheKeyParameters": [],
        "integrationResponses": {
            "200": {
                "statusCode": "200",
                "responseTemplates": {}
            }
        }
    }
}
```

範例 3：修改方法以要求 Lambda 授權

下列update-method範例會將 方法修改為所需的 Lambda 授權。

```
aws apigateway update-method --rest-api-id 1234123412 \
    --resource-id a1b2c3 \
    --http-method GET \
    --patch-operations op="replace",path="/authorizationType",value="CUSTOM"
 op="replace",path="/authorizerId",value="e4f5g6"
```

輸出：

```
{
    "httpMethod": "GET",
    "authorizationType": "CUSTOM",
    "authorizerId" : "e4f5g6",
    "apiKeyRequired": false,
    "methodResponses": {
        "200": {
            "statusCode": "200",
            "responseModels": {}
        }
    },
    "methodIntegration": {
        "type": "AWS",
        "httpMethod": "POST",
        "uri": "arn:aws:apigateway:us-east-1:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-east-1:123456789111:function:hello-world/invocations",
        "passthroughBehavior": "WHEN_NO_MATCH",
        "contentHandling": "CONVERT_TO_TEXT",
        "timeoutInMillis": 29000,
        "cacheNamespace": "h7i8j9",
        "cacheKeyParameters": [],
        "integrationResponses": {
            "200": {
                "statusCode": "200",
                "responseTemplates": {}
            }
        }
    }
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的使用 API Gateway CLI 和 REST API 建立、設定和測試用量計劃，以及控制和管理 API Gateway 中 REST API 的存取。Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateMethod。

update-model

以下程式碼範例顯示如何使用 update-model。

AWS CLI

若要變更 API 中模型的描述

命令：

```
aws apigateway update-model --rest-api-id 1234123412 --model-name 'Empty' --patch-operations op=replace,path=/description,value='New Description'
```

在 API 中變更模型的結構描述

命令：

```
aws apigateway update-model --rest-api-id 1234123412 --model-name 'Empty' --patch-operations op=replace,path=/schema,value='"{ \"$schema\": \"http://json-schema.org/draft-04/schema#\", \"title\" : \"Empty Schema\", \"type\" : \"object\" }"'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateModel。

update-resource

以下程式碼範例顯示如何使用 update-resource。

AWS CLI

移動資源並將其放置在 API 中的不同父資源下

命令：

```
aws apigateway update-resource --rest-api-id 1234123412 --resource-id 1a2b3c --patch-operations op=replace,path=/parentId,value='3c2b1a'
```

輸出：

```
{
    "path": "/resource",
    "pathPart": "resource",
    "id": "1a2b3c",
    "parentId": "3c2b1a"
}
```

重新命名 API 中的資源 (pathPart)

命令：

```
aws apigateway update-resource --rest-api-id 1234123412 --resource-id 1a2b3c --
patch-operations op=replace,path=/pathPart,value=newresourcename
```

輸出：

```
{
    "path": "/newresourcename",
    "pathPart": "newresourcename",
    "id": "1a2b3c",
    "parentId": "3c2b1a"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateResource。

update-rest-api

以下程式碼範例顯示如何使用 update-rest-api。

AWS CLI

變更 API 的名稱

命令：

```
aws apigateway update-rest-api --rest-api-id 1234123412 --patch-operations
 op=replace,path=/name,value='New Name'
```

變更 API 的描述

命令：

```
aws apigateway update-rest-api --rest-api-id 1234123412 --patch-operations
 op=replace,path=/description,value='New Description'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRestApi。

update-stage

以下程式碼範例顯示如何使用 update-stage。

AWS CLI

範例 1：覆寫資源和方法的階段設定

下列update-stage範例會覆寫階段設定，並關閉特定資源和方法的完整請求/回應記錄。

```
aws apigateway update-stage \
    --rest-api-id 1234123412 \
    --stage-name 'dev' \
    --patch-operations op=replace,path=/~1resourceName/GET/logging/
dataTrace,value=false
```

輸出：

```
{
    "deploymentId": "5ubd17",
    "stageName": "dev",
    "cacheClusterEnabled": false,
    "cacheClusterStatus": "NOT_AVAILABLE",
    "methodSettings": {
        "~1resourceName/GET": {
            "metricsEnabled": false,
            "dataTraceEnabled": false,
            "throttlingBurstLimit": 5000,
            "throttlingRateLimit": 10000.0,
            "cachingEnabled": false,
            "cacheTtlInSeconds": 300,
            "cacheDataEncrypted": false,
            "requireAuthorizationForCacheControl": true,
            "unauthorizedCacheControlHeaderStrategy": "SUCCEED_WITH_RESPONSE_HEADER"
        }
    },
    "tracingEnabled": false,
    "createdDate": "2022-07-18T10:11:18-07:00",
    "lastUpdatedDate": "2022-07-18T10:19:04-07:00"
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的設定 REST API 的階段。

範例 2：更新 API 階段之所有資源和方法的階段設定

下列update-stage範例會開啟 API 階段所有資源和方法的完整請求/回應記錄。

```
aws apigateway update-stage \
    --rest-api-id 1234123412 \
    --stage-name 'dev' \
    --patch-operations 'op=replace,path=/*/*/logging/dataTrace,value=true'
```

輸出：

```
{
    "deploymentId": "5ubd17",
    "stageName": "dev",
    "cacheClusterEnabled": false,
    "cacheClusterStatus": "NOT_AVAILABLE",
    "methodSettings": {
        "*/*": {
            "metricsEnabled": false,
            "dataTraceEnabled": true,
            "throttlingBurstLimit": 5000,
            "throttlingRateLimit": 10000.0,
            "cachingEnabled": false,
            "cacheTtlInSeconds": 300,
            "cacheDataEncrypted": false,
            "requireAuthorizationForCacheControl": true,
            "unauthorizedCacheControlHeaderStrategy": "SUCCEED_WITH_RESPONSE_HEADER"
        }
    },
    "tracingEnabled": false,
    "createdDate": "2022-07-18T10:11:18-07:00",
    "lastUpdatedDate": "2022-07-18T10:31:04-07:00"
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的設定 REST API 的階段。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateStage。

update-usage-plan

以下程式碼範例顯示如何使用 update-usage-plan。

AWS CLI

變更用量計劃中定義的期間

命令：

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations
 op="replace",path="/quota/period",value="MONTH"
```

變更用量計劃中定義的配額限制

命令：

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations
 op="replace",path="/quota/limit",value="500"
```

若要變更用量計劃中定義的調節速率限制

命令：

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations
 op="replace",path="/throttle/rateLimit",value="10"
```

若要變更用量計劃中定義的調節高載限制

命令：

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations
 op="replace",path="/throttle/burstLimit",value="20"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateUsagePlan。

update-usage

以下程式碼範例顯示如何使用 update-usage。

AWS CLI

暫時修改用量計劃中所定義目前期間的 API 金鑰配額

命令：

```
aws apigateway update-usage --usage-plan-id a1b2c3 --key-
id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu --patch-operations op="replace",path="/
remaining",value="50"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateUsage。

使用 的 API Gateway HTTP 和 WebSocket API 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 API Gateway HTTP 和 WebSocket API 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-api-mapping`

以下程式碼範例顯示如何使用 `create-api-mapping`。

AWS CLI

為 API 建立 API 映射

下列`create-api-mapping`範例會將 API 的`test`階段映射至`regional.example.com`自訂網域名稱的`/myApi`路徑。

```
aws apigatewayv2 create-api-mapping \
    --domain-name regional.example.com \
    --api-mapping-key myApi \
    --api-id a1b2c3d4 \
    --stage test
```

輸出：

```
{
    "ApiId": "a1b2c3d4",
    "ApiMappingId": "0qzs2sy7bh",
    "ApiMappingKey": "myApi"
```

```
    "Stage": "test"
}
```

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的在 API Gateway 中設定區域性自訂網域名稱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateApiMapping。

create-api

以下程式碼範例顯示如何使用 create-api。

AWS CLI

建立 HTTP API

下列create-api範例使用快速建立來建立 HTTP API。您可以使用快速建立來建立具有 AWS Lambda 或 HTTP 整合的 API、預設全部擷取路由，以及設定為自動部署變更的預設階段。下列命令使用快速建立來建立與 Lambda 函數整合的 HTTP API。

```
aws apigatewayv2 create-api \
    --name my-http-api \
    --protocol-type HTTP \
    --target arn:aws:lambda:us-west-2:123456789012:function:my-lambda-function
```

輸出：

```
{
    "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
    "ApiId": "a1b2c3d4",
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "CreatedDate": "2020-04-08T19:05:45+00:00",
    "Name": "my-http-api",
    "ProtocolType": "HTTP",
    "RouteSelectionExpression": "$request.method $request.path"
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的在 API Gateway 中開發 HTTP API。 Amazon API Gateway

建立 WebSocket API

下列create-api範例會建立具有指定名稱的 WebSocket API。

```
aws apigatewayv2 create-api \
    --name "myWebSocketApi" \
    --protocol-type WEBSOCKET \
    --route-selection-expression '$request.body.action'
```

輸出：

```
{
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "Name": "myWebSocketApi",
    "CreatedDate": "2018-11-15T06:23:51Z",
    "ProtocolType": "WEBSOCKET",
    "RouteSelectionExpression": "'$request.body.action'",
    "ApiId": "aabbccddee"
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的在 API Gateway 中建立 WebSocket API。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateApi。

create-authorizer

以下程式碼範例顯示如何使用 create-authorizer。

AWS CLI

為 HTTP API 建立 JWT 授權方

下列create-authorizer範例會建立使用 Amazon Cognito 做為身分提供者的 JWT 授權方。

```
aws apigatewayv2 create-authorizer \
    --name my-jwt-authorizer \
    --api-id a1b2c3d4 \
    --authorizer-type JWT \
    --identity-source '$request.header.Authorization' \
    --jwt-configuration Audience=123456abc,Issuer=https://cognito-idp.us-
west-2.amazonaws.com/us-west-2_abc123
```

輸出：

```
{
    "AuthorizerId": "a1b2c3",
    "AuthorizerType": "JWT",
    "IdentitySource": [
        "$request.header.Authorization"
    ],
    "JwtConfiguration": {
        "Audience": [
            "123456abc"
        ],
        "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_abc123"
    },
    "Name": "my-jwt-authorizer"
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的使用 JWT 授權方控制 HTTP API 的存取。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAuthorizer。

create-deployment

以下程式碼範例顯示如何使用 create-deployment。

AWS CLI

為 API 建立部署

下列create-deployment範例會建立 API 的部署，並將該部署與 API 的 dev階段建立關聯。

```
aws apigatewayv2 create-deployment \
    --api-id a1b2c3d4 \
    --stage-name dev
```

輸出：

```
{
    "AutoDeployed": false,
    "CreatedDate": "2020-04-06T23:38:08Z",
    "DeploymentId": "53lz9l",
    "DeploymentStatus": "DEPLOYED"
```

```
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的 API 部署。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDeployment。

`create-domain-name`

以下程式碼範例顯示如何使用 `create-domain-name`。

AWS CLI

建立自訂網域名稱

下列`create-domain-name`範例會建立 API 的區域性自訂網域名稱。

```
aws apigatewayv2 create-domain-name \
    --domain-name regional.example.com \
    --domain-name-configurations CertificateArn=arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678
```

輸出：

```
{
    "ApiMappingSelectionExpression": "$request.basepath",
    "DomainName": "regional.example.com",
    "DomainNameConfigurations": [
        {
            "ApiGatewayDomainName": "d-id.execute-api.us-west-2.amazonaws.com",
            "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
            "EndpointType": "REGIONAL",
            "HostedZoneId": "123456789111",
            "SecurityPolicy": "TLS_1_2",
            "DomainNameStatus": "AVAILABLE"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的在 API Gateway 中設定區域性自訂網域名稱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDomainName。

`create-integration`

以下程式碼範例顯示如何使用 `create-integration`。

AWS CLI

建立 WebSocket API 整合

下列`create-integration`範例會建立 WebSocket API 的模擬整合。

```
aws apigatewayv2 create-integration \
    --api-id aabbccddee \
    --passthrough-behavior WHEN_NO_MATCH \
    --timeout-in-millis 29000 \
    --connection-type INTERNET \
    --integration-type MOCK
```

輸出：

```
{
    "ConnectionType": "INTERNET",
    "IntegrationId": "0abcdef",
    "IntegrationResponseSelectionExpression": "${integration.response.statuscode}",
    "IntegrationType": "MOCK",
    "PassthroughBehavior": "WHEN_NO_MATCH",
    "PayloadFormatVersion": "1.0",
    "TimeoutInMillis": 29000
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的在 API Gateway 中設定 WebSocket API 整合請求。 Amazon API Gateway

建立 HTTP API 整合

下列`create-integration`範例會建立 HTTP API 的 AWS Lambda 整合。

```
aws apigatewayv2 create-integration \
    --api-id a1b2c3d4 \
    --integration-type AWS_PROXY \
```

```
    --integration-uri arn:aws:lambda:us-west-2:123456789012:function:my-function \
    --payload-format-version 2.0
```

輸出：

```
{
    "ConnectionType": "INTERNET",
    "IntegrationId": "0abcdef",
    "IntegrationMethod": "POST",
    "IntegrationType": "AWS_PROXY",
    "IntegrationUri": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "PayloadFormatVersion": "2.0",
    "TimeoutInMillis": 30000
}
```

如需詳細資訊，請參閱《Amazon APIs》中的設定 HTTP API 的整合。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateIntegration。

create-route

以下程式碼範例顯示如何使用 create-route。

AWS CLI

為 WebSocket 或 HTTP API 建立 $default 路由

下列create-route範例會建立 WebSocket 或 HTTP API 的$default路由。

```
aws apigatewayv2 create-route \
    --api-id aabbccddee \
    --route-key '$default'
```

輸出：

```
{
    "ApiKeyRequired": false,
    "AuthorizationType": "NONE",
    "RouteKey": "$default",
    "RouteId": "1122334"
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的使用 WebSocket APIs路由 Amazon API Gateway

建立 HTTP API 的路由

下列create-route範例會建立名為 signup的路由，接受 POST 請求。

```
aws apigatewayv2 create-route \
    --api-id aabbccddee \
    --route-key 'POST /signup'
```

輸出：

```
{
    "ApiKeyRequired": false,
    "AuthorizationType": "NONE",
    "RouteKey": "POST /signup",
    "RouteId": "1122334"
}
```

如需詳細資訊，請參閱《Amazon APIs》中的使用 HTTP API 的路由 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRoute。

create-routing-rule

以下程式碼範例顯示如何使用 create-routing-rule。

AWS CLI

建立路由規則

下列create-routing-rule範例會建立優先順序為 的路由規則50。

```
aws apigatewayv2 create-routing-rule \
    --domain-name 'regional.example.com' \
    --priority 50 \
    --conditions '[ \
        { \
            "MatchBasePaths": { \
                "AnyOf": [ \
                    "PetStoreShopper" \
                ] \
```

```
            } \
        } \
    ]' \
    --actions '[ \
        { \
            "InvokeApi": { \
                "ApiId": "abcd1234", \
                "Stage": "prod" \
            } \
        } \
    ]'
```

輸出：

```
{
    "Actions": [
        {
            "InvokeApi": {
                "ApiId": "abcd1234",
                "Stage": "prod",
                "StripBasePath": false
            }
        }
    ],
    "Conditions": [
        {
            "MatchBasePaths": {
                "AnyOf": [
                    "PetStoreShopper"
                ]
            }
        }
    ],
    "Priority": 50,
    "RoutingRuleArn": "arn:aws:apigateway:us-east-2:123456789012:/domainnames/
regional.example.com/routingrules/aaa111",
    "RoutingRuleId": "aaa111"
}
```

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的將 API 階段連線至 REST APIs 自訂網域名稱的路由規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRoutingRule。

create-stage

以下程式碼範例顯示如何使用 create-stage。

AWS CLI

建立階段

下列create-stage範例會為 API 建立名為 dev 的階段。

```
aws apigatewayv2 create-stage \
    --api-id a1b2c3d4 \
    --stage-name dev
```

輸出：

```
{
    "CreatedDate": "2020-04-06T23:23:46Z",
    "DefaultRouteSettings": {
        "DetailedMetricsEnabled": false
    },
    "LastUpdatedDate": "2020-04-06T23:23:46Z",
    "RouteSettings": {},
    "StageName": "dev",
    "StageVariables": {},
    "Tags": {}
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的使用 HTTP API 的階段。Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateStage。

create-vpc-link

以下程式碼範例顯示如何使用 create-vpc-link。

AWS CLI

建立 HTTP API 的 VPC 連結

下列create-vpc-link範例會建立 HTTP APIs VPC 連結。

```
aws apigatewayv2 create-vpc-link \
    --name MyVpcLink \
    --subnet-ids subnet-aaaa subnet-bbbb \
    --security-group-ids sg1234 sg5678
```

輸出：

```
{
    "CreatedDate": "2020-04-07T00:11:46Z",
    "Name": "MyVpcLink",
    "SecurityGroupIds": [
        "sg1234",
        "sg5678"
    ],
    "SubnetIds": [
        "subnet-aaaa",
        "subnet-bbbb"
    ],
    "Tags": {},
    "VpcLinkId": "abcd123",
    "VpcLinkStatus": "PENDING",
    "VpcLinkStatusMessage": "VPC link is provisioning ENIs",
    "VpcLinkVersion": "V2"
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的使用 HTTP APIs VPC 連結。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVpcLink。

delete-access-log-settings

以下程式碼範例顯示如何使用 delete-access-log-settings。

AWS CLI

停用 API 的存取記錄

下列delete-access-log-settings範例會刪除 API $default階段的存取日誌設定。若要停用階段的存取記錄，請刪除其存取日誌設定。

```
aws apigatewayv2 delete-access-log-settings \
    --api-id a1b2c3d4 \
```

```
    --stage-name '$default'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的設定 HTTP API 的記錄。Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAccessLogSettings。

delete-api-mapping

以下程式碼範例顯示如何使用 delete-api-mapping。

AWS CLI

刪除 API 映射

下列delete-api-mapping範例會刪除api.example.com自訂網域名稱的 API 映射。

```
aws apigatewayv2 delete-api-mapping \
    --api-mapping-id a1b2c3 \
    --domain-name api.example.com
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的在 API Gateway 中設定區域性自訂網域名稱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteApiMapping。

delete-api

以下程式碼範例顯示如何使用 delete-api。

AWS CLI

刪除 API

下列delete-api範例會刪除 API。

```
aws apigatewayv2 delete-api \
```

```
    --api-id a1b2c3d4
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的使用 HTTP API 和使用 WebSocket APIs。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteApi。

`delete-authorizer`

以下程式碼範例顯示如何使用 `delete-authorizer`。

AWS CLI

刪除授權方

下列`delete-authorizer`範例會刪除 授權方。

```
aws apigatewayv2 delete-authorizer \
    --api-id a1b2c3d4 \
    --authorizer-id a1b2c3
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的使用 JWT 授權方控制 HTTP API 的存取。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAuthorizer。

`delete-cors-configuration`

以下程式碼範例顯示如何使用 `delete-cors-configuration`。

AWS CLI

刪除 HTTP API 的 CORS 組態

下列`delete-cors-configuration`範例會刪除其 CORS 組態，以停用 HTTP API 的 CORS。

```
aws apigatewayv2 delete-cors-configuration \
```

```
    --api-id a1b2c3d4
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的設定 HTTP API 的 CORS。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCorsConfiguration。

delete-deployment

以下程式碼範例顯示如何使用 delete-deployment。

AWS CLI

刪除部署

下列delete-deployment範例會刪除 API 的部署。

```
aws apigatewayv2 delete-deployment \
    --api-id a1b2c3d4 \
    --deployment-id a1b2c3
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的 API 部署。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDeployment。

delete-domain-name

以下程式碼範例顯示如何使用 delete-domain-name。

AWS CLI

刪除自訂網域名稱

下列delete-domain-name範例會刪除自訂網域名稱。

```
aws apigatewayv2 delete-domain-name \
    --domain-name api.example.com
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的在 API Gateway 中設定區域性自訂網域名稱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDomainName。

delete-integration

以下程式碼範例顯示如何使用 delete-integration。

AWS CLI

刪除整合

下列delete-integration範例會刪除 API 整合。

```
aws apigatewayv2 delete-integration \
    --api-id a1b2c3d4 \
    --integration-id a1b2c3
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的設定 HTTP APIs 的整合和設定 WebSocket API 整合。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteIntegration。

delete-route-settings

以下程式碼範例顯示如何使用 delete-route-settings。

AWS CLI

刪除路由設定

下列delete-route-settings範例會刪除指定路由的路由設定。

```
aws apigatewayv2 delete-route-settings \
    --api-id a1b2c3d4 \
    --stage-name dev \
    --route-key 'GET /pets'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon APIs》中的使用 HTTP API 的路由。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRouteSettings。

delete-route

以下程式碼範例顯示如何使用 delete-route。

AWS CLI

刪除路由

下列delete-route範例會刪除 API 路由。

```
aws apigatewayv2 delete-route \
    --api-id a1b2c3d4 \
    --route-id a1b2c3
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon APIs》中的使用 HTTP API 的路由。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRoute。

delete-routing-rule

以下程式碼範例顯示如何使用 delete-routing-rule。

AWS CLI

刪除路由規則

下列delete-routing-rule範例會刪除自訂網域名稱的路由規則。

```
aws apigatewayv2 delete-routing-rule \
    --domain-name 'regional.example.com' \
    --routing-rule-id aaa111
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的將 API 階段連線至 REST APIs 自訂網域名稱的路由規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRoutingRule。

delete-stage

以下程式碼範例顯示如何使用 delete-stage。

AWS CLI

刪除階段

下列delete-stage範例會刪除 API 的test階段。

```
aws apigatewayv2 delete-stage \
    --api-id a1b2c3d4 \
    --stage-name test
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon APIs》中的使用 HTTP API 的階段。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteStage。

delete-vpc-link

以下程式碼範例顯示如何使用 delete-vpc-link。

AWS CLI

刪除 HTTP API 的 VPC 連結

下列delete-vpc-link範例會刪除 VPC 連結。

```
aws apigatewayv2 delete-vpc-link \
    --vpc-link-id abcd123
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的使用 HTTP APIs VPC 連結。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVpcLink。

`export-api`

以下程式碼範例顯示如何使用 `export-api`。

AWS CLI

匯出 HTTP API 的 OpenAPI 定義

下列`export-api`範例會將名為 之 API 階段的 OpenAPI 3.0 定義匯出`prod`至名為 的 YAML 檔案`stage-definition.yaml`。匯出的定義檔案預設會包含 API Gateway 延伸。

```
aws apigatewayv2 export-api \
    --api-id a1b2c3d4 \
    --output-type YAML \
    --specification OAS30 \
    --stage-name prod \
    stage-definition.yaml
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的從 API Gateway 匯出 HTTP API。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ExportApi。

`get-api-mapping`

以下程式碼範例顯示如何使用 `get-api-mapping`。

AWS CLI

取得自訂網域名稱的 API 映射相關資訊

下列`get-api-mapping`範例顯示`api.example.com`自訂網域名稱的 API 映射相關資訊。

```
aws apigatewayv2 get-api-mapping \
    --api-mapping-id a1b2c3 \
    --domain-name api.example.com
```

輸出：

```
{
    "ApiId": "a1b2c3d4",
    "ApiMappingId": "a1b2c3d5",
    "ApiMappingKey": "myTestApi"
    "Stage": "test"
}
```

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的在 API Gateway 中設定區域性自訂網域名稱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetApiMapping。

get-api-mappings

以下程式碼範例顯示如何使用 get-api-mappings。

AWS CLI

取得自訂網域名稱的 API 映射

下列get-api-mappings範例顯示api.example.com自訂網域名稱的所有 API 映射清單。

```
aws apigatewayv2 get-api-mappings \
    --domain-name api.example.com
```

輸出：

```
{
    "Items": [
        {
            "ApiId": "a1b2c3d4",
            "ApiMappingId": "a1b2c3d5",
            "ApiMappingKey": "myTestApi"
            "Stage": "test"
        },
        {
            "ApiId": "a5b6c7d8",
            "ApiMappingId": "a1b2c3d6",
            "ApiMappingKey": "myDevApi"
            "Stage": "dev"
        },
```

```
    ]
}
```

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的在 API Gateway 中設定區域性自訂網域名稱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetApiMappings。

get-api

以下程式碼範例顯示如何使用 get-api。

AWS CLI

擷取 API 的相關資訊

下列get-api範例顯示 API 的相關資訊。

```
aws apigatewayv2 get-api \
    --api-id a1b2c3d4
```

輸出：

```
{
    "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
    "ApiId": "a1b2c3d4",
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "CreatedDate": "2020-03-28T00:32:37Z",
    "Name": "my-api",
    "ProtocolType": "HTTP",
    "RouteSelectionExpression": "$request.method $request.path",
    "Tags": {
        "department": "finance"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetApi。

get-apis

以下程式碼範例顯示如何使用 get-apis。

AWS CLI

擷取 APIs清單

下列get-apis範例列出目前使用者的所有 APIs。

```
aws apigatewayv2 get-apis
```

輸出：

```
{
    "Items": [
        {
            "ApiEndpoint": "wss://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
            "ApiId": "a1b2c3d4",
            "ApiKeySelectionExpression": "$request.header.x-api-key",
            "CreatedDate": "2020-04-07T20:21:59Z",
            "Name": "my-websocket-api",
            "ProtocolType": "WEBSOCKET",
            "RouteSelectionExpression": "$request.body.message",
            "Tags": {}
        },
        {
            "ApiEndpoint": "https://a1b2c3d5.execute-api.us-west-2.amazonaws.com",
            "ApiId": "a1b2c3d5",
            "ApiKeySelectionExpression": "$request.header.x-api-key",
            "CreatedDate": "2020-04-07T20:23:50Z",
            "Name": "my-http-api",
            "ProtocolType": "HTTP",
            "RouteSelectionExpression": "$request.method $request.path",
            "Tags": {}
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的使用 HTTP API 和使用 WebSocket APIs。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetApis。

get-authorizer

以下程式碼範例顯示如何使用 get-authorizer。

AWS CLI

擷取授權方的相關資訊

下列get-authorizer範例顯示 授權方的相關資訊。

```
aws apigatewayv2 get-authorizer \
    --api-id a1b2c3d4 \
    --authorizer-id a1b2c3
```

輸出：

```
{
    "AuthorizerId": "a1b2c3",
    "AuthorizerType": "JWT",
    "IdentitySource": [
        "$request.header.Authorization"
    ],
    "JwtConfiguration": {
        "Audience": [
            "123456abc"
        ],
        "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_abc123"
    },
    "Name": "my-jwt-authorizer"
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的使用 JWT 授權方控制 HTTP API 的存取。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAuthorizer。

get-authorizers

以下程式碼範例顯示如何使用 get-authorizers。

AWS CLI

擷取 API 的授權方清單

下列get-authorizers範例顯示 API 的所有授權方清單。

```
aws apigatewayv2 get-authorizers \
    --api-id a1b2c3d4
```

輸出：

```
{
    "Items": [
        {
            "AuthorizerId": "a1b2c3",
            "AuthorizerType": "JWT",
            "IdentitySource": [
                "$request.header.Authorization"
            ],
            "JwtConfiguration": {
                "Audience": [
                    "123456abc"
                ],
                "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-
west-2_abc123"
            },
            "Name": "my-jwt-authorizer"
        },
        {
            "AuthorizerId": "a1b2c4",
            "AuthorizerType": "JWT",
            "IdentitySource": [
                "$request.header.Authorization"
            ],
            "JwtConfiguration": {
                "Audience": [
                    "6789abcde"
                ],
                "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-
west-2_abc234"
            },
            "Name": "new-jwt-authorizer"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的使用 JWT 授權方控制 HTTP API 的存取。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAuthorizers。

get-deployment

以下程式碼範例顯示如何使用 get-deployment。

AWS CLI

擷取部署的相關資訊

下列get-deployment範例顯示部署的相關資訊。

```
aws apigatewayv2 get-deployment \
    --api-id a1b2c3d4 \
    --deployment-id abcdef
```

輸出：

```
{
    "AutoDeployed": true,
    "CreatedDate": "2020-04-07T23:58:40Z",
    "DeploymentId": "abcdef",
    "DeploymentStatus": "DEPLOYED",
    "Description": "Automatic deployment triggered by changes to the Api
 configuration"
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的 API 部署。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDeployment。

get-deployments

以下程式碼範例顯示如何使用 get-deployments。

AWS CLI

擷取部署清單

下列get-deployments範例顯示所有 API 部署的清單。

```
aws apigatewayv2 get-deployments \
```

```
    --api-id a1b2c3d4
```

輸出：

```
{
    "Items": [
        {
            "AutoDeployed": true,
            "CreatedDate": "2020-04-07T23:58:40Z",
            "DeploymentId": "abcdef",
            "DeploymentStatus": "DEPLOYED",
            "Description": "Automatic deployment triggered by changes to the Api
 configuration"
        },
        {
            "AutoDeployed": true,
            "CreatedDate": "2020-04-06T00:33:00Z",
            "DeploymentId": "bcdefg",
            "DeploymentStatus": "DEPLOYED",
            "Description": "Automatic deployment triggered by changes to the Api
 configuration"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的 API 部署。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDeployments。

get-domain-name

以下程式碼範例顯示如何使用 get-domain-name。

AWS CLI

擷取自訂網域名稱的相關資訊

下列get-domain-name範例顯示自訂網域名稱的相關資訊。

```
aws apigatewayv2 get-domain-name \
    --domain-name api.example.com
```

輸出：

```
{
    "ApiMappingSelectionExpression": "$request.basepath",
    "DomainName": "api.example.com",
    "DomainNameConfigurations": [
        {
            "ApiGatewayDomainName": "d-1234.execute-api.us-west-2.amazonaws.com",
            "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
            "EndpointType": "REGIONAL",
            "HostedZoneId": "123456789111",
            "SecurityPolicy": "TLS_1_2",
            "DomainNameStatus": "AVAILABLE"
        }
    ],
    "Tags": {}
}
```

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的在 API Gateway 中設定區域性自訂網域名稱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDomainName。

get-domain-names

以下程式碼範例顯示如何使用 get-domain-names。

AWS CLI

擷取自訂網域名稱的清單

下列get-domain-names範例顯示目前使用者的所有自訂網域名稱清單。

```
aws apigatewayv2 get-domain-names
```

輸出：

```
{
    "Items": [
        {
            "ApiMappingSelectionExpression": "$request.basepath",
            "DomainName": "api.example.com",
            "DomainNameConfigurations": [
```

```
                {
                    "ApiGatewayDomainName": "d-1234.execute-api.us-
west-2.amazonaws.com",
                    "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
                    "EndpointType": "REGIONAL",
                    "HostedZoneId": "123456789111",
                    "SecurityPolicy": "TLS_1_2",
                    "DomainNameStatus": "AVAILABLE"
                }
            ]
        },
        {
            "ApiMappingSelectionExpression": "$request.basepath",
            "DomainName": "newApi.example.com",
            "DomainNameConfigurations": [
                {
                    "ApiGatewayDomainName": "d-5678.execute-api.us-
west-2.amazonaws.com",
                    "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
                    "EndpointType": "REGIONAL",
                    "HostedZoneId": "123456789222",
                    "SecurityPolicy": "TLS_1_2",
                    "DomainNameStatus": "AVAILABLE"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的在 API Gateway 中設定區域性自訂網域名稱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDomainNames。

get-integration

以下程式碼範例顯示如何使用 `get-integration`。

AWS CLI

擷取整合的相關資訊

下列get-integration範例顯示 整合的相關資訊。

```
aws apigatewayv2 get-integration \
    --api-id a1b2c3d4 \
    --integration-id a1b2c3
```

輸出：

```
{
    "ApiGatewayManaged": true,
    "ConnectionType": "INTERNET",
    "IntegrationId": "a1b2c3",
    "IntegrationMethod": "POST",
    "IntegrationType": "AWS_PROXY",
    "IntegrationUri": "arn:aws:lambda:us-west-2:12356789012:function:hello12",
    "PayloadFormatVersion": "2.0",
    "TimeoutInMillis": 30000
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的設定 HTTP APIs 的整合和設定 WebSocket API 整合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIntegration。

get-integrations

以下程式碼範例顯示如何使用 get-integrations。

AWS CLI

擷取整合清單

下列get-integrations範例顯示所有 API 整合的清單。

```
aws apigatewayv2 get-integrations \
    --api-id a1b2c3d4
```

輸出：

```
{
    "Items": [
```

```
        {
            "ApiGatewayManaged": true,
            "ConnectionType": "INTERNET",
            "IntegrationId": "a1b2c3",
            "IntegrationMethod": "POST",
            "IntegrationType": "AWS_PROXY",
            "IntegrationUri": "arn:aws:lambda:us-west-2:123456789012:function:my-
function",
            "PayloadFormatVersion": "2.0",
            "TimeoutInMillis": 30000
        },
        {
            "ConnectionType": "INTERNET",
            "IntegrationId": "a1b2c4",
            "IntegrationMethod": "ANY",
            "IntegrationType": "HTTP_PROXY",
            "IntegrationUri": "https://www.example.com",
            "PayloadFormatVersion": "1.0",
            "TimeoutInMillis": 30000
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的設定 HTTP APIs 的整合和設定 WebSocket API 整合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIntegrations。

get-route

以下程式碼範例顯示如何使用 get-route。

AWS CLI

擷取路由的相關資訊

下列get-route範例顯示路由的相關資訊。

```
aws apigatewayv2 get-route \
    --api-id a1b2c3d4 \
    --route-id 72jz1wk
```

輸出：

```
{
    "ApiKeyRequired": false,
    "AuthorizationType": "NONE",
    "RouteId": "72jz1wk",
    "RouteKey": "ANY /pets",
    "Target": "integrations/a1b2c3"
}
```

如需詳細資訊，請參閱《Amazon APIs》中的使用 HTTP API 的路由 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRoute。

get-routes

以下程式碼範例顯示如何使用 get-routes。

AWS CLI

擷取路由清單

下列get-routes範例顯示所有 API 路由的清單。

```
aws apigatewayv2 get-routes \
    --api-id a1b2c3d4
```

輸出：

```
{
    "Items": [
        {
            "ApiKeyRequired": false,
            "AuthorizationType": "NONE",
            "RouteId": "72jz1wk",
            "RouteKey": "ANY /admin",
            "Target": "integrations/a1b2c3"
        },
        {
            "ApiGatewayManaged": true,
            "ApiKeyRequired": false,
            "AuthorizationType": "NONE",
            "RouteId": "go65gqi",
            "RouteKey": "$default",
```

```
            "Target": "integrations/a1b2c4"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon APIs》中的使用 HTTP API 的路由 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRoutes。

get-routing-rule

以下程式碼範例顯示如何使用 get-routing-rule。

AWS CLI

取得路由規則

下列get-routing-rule範例會取得網域名稱的路由規則。

```
aws apigatewayv2 get-routing-rule \
    --domain-name 'regional.example.com' \
    --routing-rule-id aaa111
```

輸出：

```
{
    "Actions": [
        {
            "InvokeApi": {
                "ApiId": "abcd1234",
                "Stage": "prod",
                "StripBasePath": false
            }
        }
    ],
    "Conditions": [
        {
            "MatchBasePaths": {
                "AnyOf": [
                    "PetStoreShopper"
                ]
            }
```

```
        }
    ],
    "Priority": 50,
    "RoutingRuleArn": "arn:aws:apigateway:us-east-2:123456789012:/domainnames/
regional.example.com/routingrules/aaa111",
    "RoutingRuleId": "aaa111"
}
```

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的將 API 階段連線至 REST APIs 自訂網域名稱的路由規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRoutingRule。

get-stage

以下程式碼範例顯示如何使用 get-stage。

AWS CLI

擷取階段的相關資訊

下列get-stage範例顯示 API prod階段的相關資訊。

```
aws apigatewayv2 get-stage \
    --api-id a1b2c3d4 \
    --stage-name prod
```

輸出：

```
{
    "CreatedDate": "2020-04-08T00:36:05Z",
    "DefaultRouteSettings": {
        "DetailedMetricsEnabled": false
    },
    "DeploymentId": "x1zwyv",
    "LastUpdatedDate": "2020-04-08T00:36:13Z",
    "RouteSettings": {},
    "StageName": "prod",
    "StageVariables": {
        "function": "my-prod-function"
    },
    "Tags": {}
```

```
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的使用 HTTP API 的階段。Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetStage。

get-stages

以下程式碼範例顯示如何使用 get-stages。

AWS CLI

擷取階段清單

下列get-stages範例列出 API 的所有階段。

```
aws apigatewayv2 get-stages \
    --api-id a1b2c3d4
```

輸出：

```
{
    "Items": [
        {
            "ApiGatewayManaged": true,
            "AutoDeploy": true,
            "CreatedDate": "2020-04-08T00:08:44Z",
            "DefaultRouteSettings": {
                "DetailedMetricsEnabled": false
            },
            "DeploymentId": "dty748",
            "LastDeploymentStatusMessage": "Successfully deployed stage with
 deployment ID 'dty748'",
            "LastUpdatedDate": "2020-04-08T00:09:49Z",
            "RouteSettings": {},
            "StageName": "$default",
            "StageVariables": {},
            "Tags": {}
        },
        {
            "AutoDeploy": true,
            "CreatedDate": "2020-04-08T00:35:06Z",
```

```
            "DefaultRouteSettings": {
                "DetailedMetricsEnabled": false
            },
            "LastUpdatedDate": "2020-04-08T00:35:48Z",
            "RouteSettings": {},
            "StageName": "dev",
            "StageVariables": {
                "function": "my-dev-function"
            },
            "Tags": {}
        },
        {
            "CreatedDate": "2020-04-08T00:36:05Z",
            "DefaultRouteSettings": {
                "DetailedMetricsEnabled": false
            },
            "DeploymentId": "x1zwyv",
            "LastUpdatedDate": "2020-04-08T00:36:13Z",
            "RouteSettings": {},
            "StageName": "prod",
            "StageVariables": {
                "function": "my-prod-function"
            },
            "Tags": {}
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon APIs》中的使用 HTTP API 的階段。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetStages。

get-tags

以下程式碼範例顯示如何使用 get-tags。

AWS CLI

擷取資源的標籤清單

下列get-tags範例列出 API 的所有標籤。

```
aws apigatewayv2 get-tags \
```

```
    --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4
```

輸出：

```
{
    "Tags": {
        "owner": "dev-team",
        "environment": "prod"
    }
}
```

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的標記 API Gateway 資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTags。

get-vpc-link

以下程式碼範例顯示如何使用 get-vpc-link。

AWS CLI

擷取 VPC 連結的相關資訊

下列get-vpc-link範例顯示 VPC 連結的相關資訊。

```
aws apigatewayv2 get-vpc-link \
    --vpc-link-id abcd123
```

輸出：

```
{
    "CreatedDate": "2020-04-07T00:27:47Z",
    "Name": "MyVpcLink",
    "SecurityGroupIds": [
        "sg1234",
        "sg5678"
    ],
    "SubnetIds": [
        "subnet-aaaa",
        "subnet-bbbb"
    ],
```

```
    "Tags": {},
    "VpcLinkId": "abcd123",
    "VpcLinkStatus": "AVAILABLE",
    "VpcLinkStatusMessage": "VPC link is ready to route traffic",
    "VpcLinkVersion": "V2"
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的使用 HTTP APIs VPC 連結。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetVpcLink。

get-vpc-links

以下程式碼範例顯示如何使用 get-vpc-links。

AWS CLI

擷取 VPC 連結清單

下列get-vpc-links範例顯示目前使用者的所有 VPC 連結清單。

```
aws apigatewayv2 get-vpc-links
```

輸出：

```
{
    "Items": [
        {
            "CreatedDate": "2020-04-07T00:27:47Z",
            "Name": "MyVpcLink",
            "SecurityGroupIds": [
                "sg1234",
                "sg5678"
            ],
            "SubnetIds": [
                "subnet-aaaa",
                "subnet-bbbb"
            ],
            "Tags": {},
            "VpcLinkId": "abcd123",
            "VpcLinkStatus": "AVAILABLE",
            "VpcLinkStatusMessage": "VPC link is ready to route traffic",
            "VpcLinkVersion": "V2"
```

```
        }
        {
            "CreatedDate": "2020-04-07T00:27:47Z",
            "Name": "MyOtherVpcLink",
            "SecurityGroupIds": [
                "sg1234",
                "sg5678"
            ],
            "SubnetIds": [
                "subnet-aaaa",
                "subnet-bbbb"
            ],
            "Tags": {},
            "VpcLinkId": "abcd456",
            "VpcLinkStatus": "AVAILABLE",
            "VpcLinkStatusMessage": "VPC link is ready to route traffic",
            "VpcLinkVersion": "V2"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的使用 HTTP APIs VPC 連結。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetVpcLinks。

`import-api`

以下程式碼範例顯示如何使用 `import-api`。

AWS CLI

匯入 HTTP API

下列`import-api`範例會從名為 的 OpenAPI 3`api-definition.yaml`.0 定義檔案建立 HTTP API。

```
aws apigatewayv2 import-api \
    --body file://api-definition.yaml
```

`api-definition.yaml` 的內容：

```
openapi: 3.0.1
```

```
info:
    title: My Lambda API
    version: v1.0
paths:
    /hello:
        x-amazon-apigateway-any-method:
            x-amazon-apigateway-integration:
                payloadFormatVersion: 2.0
                type: aws_proxy
                httpMethod: POST
                uri: arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123456789012:function:hello/invocations
                connectionType: INTERNET
```

輸出：

```
{
    "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
    "ApiId": "a1b2c3d4",
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "CreatedDate": "2020-04-08T17:19:38+00:00",
    "Name": "My Lambda API",
    "ProtocolType": "HTTP",
    "RouteSelectionExpression": "$request.method $request.path",
    "Tags": {},
    "Version": "v1.0"
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的使用 HTTP API 的 OpenAPI 定義 APIs。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ImportApi。

`list-routing-rules`

以下程式碼範例顯示如何使用 `list-routing-rules`。

AWS CLI

列出路由規則

下列`list-routing-rules`範例列出網域名稱的路由規則。

```
aws apigatewayv2 list-routing-rules \
    --domain-name 'regional.example.com'
```

輸出：

```
{
    "RoutingRules": [
        {
            "Actions": [
                {
                    "InvokeApi": {
                        "ApiId": "abcd1234",
                        "Stage": "prod",
                        "StripBasePath": false
                    }
                }
            ],
            "Conditions": [
                {
                    "MatchBasePaths": {
                        "AnyOf": [
                            "PetStoreShopper"
                        ]
                    }
                }
            ],
            "Priority": 150,
            "RoutingRuleArn": "arn:aws:apigateway:us-east-1:123456789012:/
domainnames/regional.example.com/routingrules/aaa111",
            "RoutingRuleId": "aaa111"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的將 API 階段連線至 REST APIs 自訂網域名稱的路由規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRoutingRules。

put-routing-rule

以下程式碼範例顯示如何使用 put-routing-rule。

AWS CLI

更新路由規則

下列put-routing-rule範例會更新路由規則的優先順序。

```
aws apigatewayv2 put-routing-rule \
    --domain-name 'regional.example.com' \
    --priority 150 \
    --conditions '[ \
        { \
            "MatchBasePaths": { \
                "AnyOf": [ \
                    "PetStoreShopper" \
                ] \
            } \
        } \
    ]' \
    --actions '[ \
        { \
            "InvokeApi": { \
                "ApiId": "abcd1234", \
                "Stage": "prod" \
            } \
        } \
    ]'
```

輸出：

```
{
    "Actions": [
        {
            "InvokeApi": {
                "ApiId": "abcd1234",
                "Stage": "prod",
                "StripBasePath": false
            }
        }
    ],
    "Conditions": [
        {
            "MatchBasePaths": {
                "AnyOf": [
```

```
                    "PetStoreShopper"
                ]
            }
        }
    ],
    "Priority": 150,
    "RoutingRuleArn": "arn:aws:apigateway:us-east-2:123456789012:/domainnames/
regional.example.com/routingrules/aaa111",
    "RoutingRuleId": "aaa111"
}
```

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的將 API 階段連線至 REST APIs 自訂網域名稱的路由規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutRoutingRule。

reimport-api

以下程式碼範例顯示如何使用 reimport-api。

AWS CLI

重新匯入 HTTP API

下列reimport-api範例會更新現有的 HTTP API，以使用 中指定的 OpenAPI 3.0 定義api-definition.yaml。

```
aws apigatewayv2 reimport-api \
    --body file://api-definition.yaml \
    --api-id a1b2c3d4
```

api-definition.yaml 的內容：

```
openapi: 3.0.1
info:
    title: My Lambda API
    version: v1.0
paths:
    /hello:
        x-amazon-apigateway-any-method:
            x-amazon-apigateway-integration:
                payloadFormatVersion: 2.0
```

```
                type: aws_proxy
                httpMethod: POST
                uri: arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:12356789012:function:hello/invocations
                connectionType: INTERNET
```

輸出：

```
{
    "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
    "ApiId": "a1b2c3d4",
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "CreatedDate": "2020-04-08T17:19:38+00:00",
    "Name": "My Lambda API",
    "ProtocolType": "HTTP",
    "RouteSelectionExpression": "$request.method $request.path",
    "Tags": {},
    "Version": "v1.0"
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的使用 HTTP APIs OpenAPI 定義。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReimportApi。

`tag-resource`

以下程式碼範例顯示如何使用 `tag-resource`。

AWS CLI

標記資源

下列`tag-resource`範例會將索引鍵名稱為 `Department`且值為 的標籤`Accounting`新增至指定的 API。

```
aws apigatewayv2 tag-resource \
    --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4 \
    --tags Department=Accounting
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的標記 API Gateway 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源移除標籤

下列untag-resource範例Owner會從指定的 API 移除具有金鑰名稱 Project和 的標籤。

```
aws apigatewayv2 untag-resource \
    --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4 \
    --tag-keys Project Owner
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的標記 API Gateway 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-api-mapping

以下程式碼範例顯示如何使用 update-api-mapping。

AWS CLI

更新 API 映射

下列update-api-mapping範例會變更自訂網域名稱的 API 映射。因此，使用指定 API 和階段之自訂網域名稱的基本 URL 會變成 https://api.example.com/dev。

```
aws apigatewayv2 update-api-mapping \
    --api-id a1b2c3d4 \
    --stage dev \
    --domain-name api.example.com \
```

```
    --api-mapping-id 0qzs2sy7bh \
    --api-mapping-key dev
```

輸出：

```
{
    "ApiId": "a1b2c3d4",
    "ApiMappingId": "0qzs2sy7bh",
    "ApiMappingKey": "dev"
    "Stage": "dev"
}
```

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的在 API Gateway 中設定區域性自訂網域名稱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateApiMapping。

update-api

以下程式碼範例顯示如何使用 update-api。

AWS CLI

啟用 HTTP API 的 CORS

下列update-api範例會更新指定 API 的 CORS 組態，以允許來自 的請求https://www.example.com。

```
aws apigatewayv2 update-api \
    --api-id a1b2c3d4 \
    --cors-configuration AllowOrigins=https://www.example.com
```

輸出：

```
{
    "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
    "ApiId": "a1b2c3d4",
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "CorsConfiguration": {
        "AllowCredentials": false,
        "AllowHeaders": [
```

```
            "header1",
            "header2"
        ],
        "AllowMethods": [
            "GET",
            "OPTIONS"
        ],
        "AllowOrigins": [
            "https://www.example.com"
        ]
    },
    "CreatedDate": "2020-04-08T18:39:37+00:00",
    "Name": "my-http-api",
    "ProtocolType": "HTTP",
    "RouteSelectionExpression": "$request.method $request.path",
    "Tags": {},
    "Version": "v1.0"
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的設定 HTTP API 的 CORS。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateApi。

update-authorizer

以下程式碼範例顯示如何使用 update-authorizer。

AWS CLI

更新授權方

下列update-authorizer範例會將 JWT 授權方的身分來源變更為名為 的標頭Authorization。

```
aws apigatewayv2 update-authorizer \
    --api-id a1b2c3d4 \
    --authorizer-id a1b2c3 \
    --identity-source '$request.header.Authorization'
```

輸出：

```
{
```

```
    "AuthorizerId": "a1b2c3",
    "AuthorizerType": "JWT",
    "IdentitySource": [
        "$request.header.Authorization"
    ],
    "JwtConfiguration": {
        "Audience": [
            "123456abc"
        ],
        "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_abc123"
    },
    "Name": "my-jwt-authorizer"
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的使用 JWT 授權方控制 HTTP API 的存取。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAuthorizer。

update-deployment

以下程式碼範例顯示如何使用 update-deployment。

AWS CLI

若要變更部署的描述

下列update-deployment範例會更新部署的描述。

```
aws apigatewayv2 update-deployment \
    --api-id a1b2c3d4 \
    --deployment-id abcdef \
    --description 'Manual deployment to fix integration test failures.'
```

輸出：

```
{
    "AutoDeployed": false,
    "CreatedDate": "2020-02-05T16:21:48+00:00",
    "DeploymentId": "abcdef",
    "DeploymentStatus": "DEPLOYED",
    "Description": "Manual deployment to fix integration test failures."
```

```
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的在 API Gateway 中開發 HTTP API。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDeployment。

update-domain-name

以下程式碼範例顯示如何使用 update-domain-name。

AWS CLI

更新自訂網域名稱

下列update-domain-name範例會指定api.example.com自訂網域名稱的新 ACM 憑證。

```
aws apigatewayv2 update-domain-name \
    --domain-name api.example.com \
    --domain-name-configurations CertificateArn=arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678
```

輸出：

```
{
    "ApiMappingSelectionExpression": "$request.basepath",
    "DomainName": "regional.example.com",
    "DomainNameConfigurations": [
        {
            "ApiGatewayDomainName": "d-id.execute-api.us-west-2.amazonaws.com",
            "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
            "EndpointType": "REGIONAL",
            "HostedZoneId": "123456789111",
            "SecurityPolicy": "TLS_1_2",
            "DomainNameStatus": "AVAILABLE"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon API Gateway API Gateway 開發人員指南》中的在 API Gateway 中設定區域性自訂網域名稱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDomainName。

update-integration

以下程式碼範例顯示如何使用 update-integration。

AWS CLI

更新 Lambda 整合

下列update-integration範例會更新現有的 AWS Lambda 整合，以使用指定的 Lambda 函數。

```
aws apigatewayv2 update-integration \
    --api-id a1b2c3d4 \
    --integration-id a1b2c3 \
    --integration-uri arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123456789012:function:my-new-function/invocations
```

輸出：

```
{
    "ConnectionType": "INTERNET",
    "IntegrationId": "a1b2c3",
    "IntegrationMethod": "POST",
    "IntegrationType": "AWS_PROXY",
    "IntegrationUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/
functions/arn:aws:lambda:us-west-2:123456789012:function:my-new-function/
invocations",
    "PayloadFormatVersion": "2.0",
    "TimeoutInMillis": 5000
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的設定 HTTP APIs 的整合和設定 WebSocket API 整合。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateIntegration。

update-route

以下程式碼範例顯示如何使用 update-route。

AWS CLI

範例 1：更新路由的整合

下列update-route範例會更新指定路由的整合。

```
aws apigatewayv2 update-route \
    --api-id a1b2c3d4 \
    --route-id a1b2c3 \
    --target integrations/a1b2c6
```

輸出：

```
{
    "ApiKeyRequired": false,
    "AuthorizationType": "NONE",
    "RouteId": "a1b2c3",
    "RouteKey": "ANY /pets",
    "Target": "integrations/a1b2c6"
}
```

範例 2：將 授權方新增至路由

下列update-route範例會將指定的路由更新為使用 JWT 授權方。

```
aws apigatewayv2 update-route \
    --api-id a1b2c3d4  \
    --route-id a1b2c3  \
    --authorization-type JWT \
    --authorizer-id a1b2c5 \
    --authorization-scopes user.id user.email
```

輸出：

```
{
    "ApiKeyRequired": false,
    "AuthorizationScopes": [
        "user.id",
        "user.email"
    ],
    "AuthorizationType": "JWT",
    "AuthorizerId": "a1b2c5",
```

```
    "OperationName": "GET HTTP",
    "RequestParameters": {},
    "RouteId": "a1b2c3",
    "RouteKey": "GET /pets",
    "Target": "integrations/a1b2c6"
}
```

如需詳細資訊，請參閱《Amazon APIs Gateway 開發人員指南》中的使用 JWT 授權方控制 HTTP API 的存取。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRoute。

update-stage

以下程式碼範例顯示如何使用 update-stage。

AWS CLI

設定自訂限流

下列update-stage範例會設定 API 指定階段和路由的自訂限流。

```
aws apigatewayv2 update-stage \
    --api-id a1b2c3d4 \
    --stage-name dev \
    --route-settings '{"GET /pets":
{"ThrottlingBurstLimit":100,"ThrottlingRateLimit":2000}}'
```

輸出：

```
{
    "CreatedDate": "2020-04-05T16:21:16+00:00",
    "DefaultRouteSettings": {
        "DetailedMetricsEnabled": false
    },
    "DeploymentId": "shktxb",
    "LastUpdatedDate": "2020-04-08T22:23:17+00:00",
    "RouteSettings": {
        "GET /pets": {
            "ThrottlingBurstLimit": 100,
            "ThrottlingRateLimit": 2000.0
        }
```

```
    },
    "StageName": "dev",
    "StageVariables": {},
    "Tags": {}
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的保護 HTTP API。 Amazon API Gateway

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateStage。

update-vpc-link

以下程式碼範例顯示如何使用 update-vpc-link。

AWS CLI

更新 VPC 連結

下列update-vpc-link範例會更新 VPC 連結的名稱。建立 VPC 連結後，您無法變更其安全群組或子網路。

```
aws apigatewayv2 update-vpc-link \
    --vpc-link-id abcd123 \
    --name MyUpdatedVpcLink
```

輸出：

```
{
    "CreatedDate": "2020-04-07T00:27:47Z",
    "Name": "MyUpdatedVpcLink",
    "SecurityGroupIds": [
        "sg1234",
        "sg5678"
    ],
    "SubnetIds": [
        "subnet-aaaa",
        "subnet-bbbb"
    ],
    "Tags": {},
    "VpcLinkId": "abcd123",
    "VpcLinkStatus": "AVAILABLE",
```

```
    "VpcLinkStatusMessage": "VPC link is ready to route traffic",
    "VpcLinkVersion": "V2"
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的使用 HTTP APIs VPC 連結。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateVpcLink。

使用 的 API Gateway Management API 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 API Gateway Management API 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`delete-connection`

以下程式碼範例顯示如何使用 `delete-connection`。

AWS CLI

刪除 WebSocket 連線

下列`delete-connection`範例會中斷用戶端與指定 WebSocket API 的連線。

```
aws apigatewaymanagementapi delete-connection \
    --connection-id L0SM9cOFvHcCIhw= \
    --endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的在後端服務中使用 @connections 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteConnection。

`get-connection`

以下程式碼範例顯示如何使用 `get-connection`。

AWS CLI

取得 WebSocket 連線的相關資訊

下列`get-connection`範例說明與指定 WebSocket API 的連線。

```
aws apigatewaymanagementapi get-connection \
    --connection-id L0SM9cOFvHcCIhw= \
    --endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

輸出：

```
{
    "ConnectedAt": "2020-04-30T20:10:33.236Z",
    "Identity": {
        "SourceIp": "192.0.2.1"
    },
    "LastActiveAt": "2020-04-30T20:10:42.997Z"
}
```

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的在後端服務中使用 @connections 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetConnection。

`post-to-connection`

以下程式碼範例顯示如何使用 `post-to-connection`。

AWS CLI

將資料傳送至 WebSocket 連線

下列`post-to-connection`範例會傳送訊息至連線至指定 WebSocket API 的用戶端。

```
aws apigatewaymanagementapi post-to-connection \
```

```
    --connection-id L0SM9cOFvHcCIhw= \
    --data "Hello from API Gateway!" \
    --endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon API Gateway 開發人員指南》中的在後端服務中使用 @connections 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PostToConnection。

使用 的 App Mesh 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 App Mesh 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-mesh`

以下程式碼範例顯示如何使用 `create-mesh`。

AWS CLI

範例 1：建立新的服務網格

下列`create-mesh`範例會建立服務網格。

```
aws appmesh create-mesh \
    --mesh-name app1
```

輸出：

```
{
    "mesh":{
        "meshName":"app1",
        "metadata":{
            "arn":"arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
            "createdAt":1563809909.282,
            "lastUpdatedAt":1563809909.282,
            "uid":"a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version":1
        },
        "spec":{},
        "status":{
            "status":"ACTIVE"
        }
    }
}
```

範例 2：建立具有多個標籤的新服務網格

下列create-mesh範例會建立具有多個標籤的服務網格。

```
aws appmesh create-mesh \
    --mesh-name app2 \
    --tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3
```

輸出：

```
{
    "mesh":{
        "meshName":"app2",
        "metadata":{
            "arn":"arn:aws:appmesh:us-east-1:123456789012:mesh/app2",
            "createdAt":1563822121.877,
            "lastUpdatedAt":1563822121.877,
            "uid":"a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version":1
        },
        "spec":{},
        "status":{
            "status":"ACTIVE"
        }
    }
```

```
}
```

如需詳細資訊，請參閱《App Mesh 使用者指南》中的 Service Meshes。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateMesh。

create-route

以下程式碼範例顯示如何使用 create-route。

AWS CLI

建立新的 gRPC 路由

下列create-route範例使用 JSON 輸入檔案來建立 gRPC 路由。具有以 123 開頭之中繼資料的 GRPC 流量會路由至名為 serviceBgrpc 的虛擬節點。如果嘗試與路由的目標通訊時，發生特定的 gRPC、HTTP 或 TCP 失敗，則會重試路由三次。每次重試嘗試之間會有 15 秒的延遲。

```
aws appmesh create-route \
    --cli-input-json file://create-route-grpc.json
```

create-route-grpc.json 的內容：

```
{
    "meshName" : "apps",
    "routeName" : "grpcRoute",
    "spec" : {
       "grpcRoute" : {
          "action" : {
             "weightedTargets" : [
                {
                   "virtualNode" : "serviceBgrpc",
                   "weight" : 100
                }
             ]
          },
          "match" : {
             "metadata" : [
                {
                   "invert" : false,
                   "match" : {
                      "prefix" : "123"
```

```
                    },
                    "name" : "myMetadata"
                }
            ],
            "methodName" : "GetColor",
            "serviceName" : "com.amazonaws.services.ColorService"
         },
         "retryPolicy" : {
            "grpcRetryEvents" : [ "deadline-exceeded" ],
            "httpRetryEvents" : [ "server-error", "gateway-error" ],
            "maxRetries" : 3,
            "perRetryTimeout" : {
               "unit" : "s",
               "value" : 15
            },
            "tcpRetryEvents" : [ "connection-error" ]
         }
      },
      "priority" : 100
   },
   "virtualRouterName" : "serviceBgrpc"
}
```

輸出：

```
{
    "route": {
        "meshName": "apps",
        "metadata": {
            "arn": "arn:aws:appmesh:us-west-2:123456789012:mesh/apps/virtualRouter/
serviceBgrpc/route/grpcRoute",
            "createdAt": 1572010806.008,
            "lastUpdatedAt": 1572010806.008,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "routeName": "grpcRoute",
        "spec": {
            "grpcRoute": {
                "action": {
                    "weightedTargets": [
                        {
                            "virtualNode": "serviceBgrpc",
```

```
                            "weight": 100
                        }
                    ]
                },
                "match": {
                    "metadata": [
                        {
                            "invert": false,
                            "match": {
                                "prefix": "123"
                            },
                            "name": "mymetadata"
                        }
                    ],
                    "methodName": "GetColor",
                    "serviceName": "com.amazonaws.services.ColorService"
                },
                "retryPolicy": {
                    "grpcRetryEvents": [
                        "deadline-exceeded"
                    ],
                    "httpRetryEvents": [
                        "server-error",
                        "gateway-error"
                    ],
                    "maxRetries": 3,
                    "perRetryTimeout": {
                        "unit": "s",
                        "value": 15
                    },
                    "tcpRetryEvents": [
                        "connection-error"
                    ]
                }
            },
            "priority": 100
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualRouterName": "serviceBgrpc"
    }
}
```

建立新的 HTTP 或 HTTP/2 路由

下列create-route範例使用 JSON 輸入檔案來建立 HTTP/2 路由。若要建立 HTTP 路由，請以規格下的 http2RoutehttpRoute。所有指向標頭值開頭為 123 的任何 URL 字首的 HTTP/2 流量都會路由至名為 serviceBhttp2 的虛擬節點。嘗試與路由的目標通訊時，如果有特定的 HTTP 或 TCP 失敗，則會重試路由三次。每次重試嘗試之間會有 15 秒的延遲。

```
aws appmesh create-route \
    --cli-input-json file://create-route-http2.json
```

create-route-http2.json 的內容：

```
{
    "meshName": "apps",
    "routeName": "http2Route",
    "spec": {
        "http2Route": {
            "action": {
                "weightedTargets": [
                    {
                        "virtualNode": "serviceBhttp2",
                        "weight": 100
                    }
                ]
            },
            "match": {
                "headers": [
                    {
                        "invert": false,
                        "match": {
                            "prefix": "123"
                        },
                        "name": "clientRequestId"
                    }
                ],
                "method": "POST",
                "prefix": "/",
                "scheme": "http"
            },
            "retryPolicy": {
                "httpRetryEvents": [
                    "server-error",
                    "gateway-error"
```

```
                ],
                "maxRetries": 3,
                "perRetryTimeout": {
                    "unit": "s",
                    "value": 15
                },
                "tcpRetryEvents": [
                    "connection-error"
                ]
            }
        },
        "priority": 200
    },
    "virtualRouterName": "serviceBhttp2"
}
```

輸出：

```
{
    "route": {
        "meshName": "apps",
        "metadata": {
            "arn": "arn:aws:appmesh:us-west-2:123456789012:mesh/apps/virtualRouter/
serviceBhttp2/route/http2Route",
            "createdAt": 1572011008.352,
            "lastUpdatedAt": 1572011008.352,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "routeName": "http2Route",
        "spec": {
            "http2Route": {
                "action": {
                    "weightedTargets": [
                        {
                            "virtualNode": "serviceBhttp2",
                            "weight": 100
                        }
                    ]
                },
                "match": {
                    "headers": [
                        {
```

```
                            "invert": false,
                            "match": {
                                "prefix": "123"
                            },
                            "name": "clientRequestId"
                        }
                    ],
                    "method": "POST",
                    "prefix": "/",
                    "scheme": "http"
                },
                "retryPolicy": {
                    "httpRetryEvents": [
                        "server-error",
                        "gateway-error"
                    ],
                    "maxRetries": 3,
                    "perRetryTimeout": {
                        "unit": "s",
                        "value": 15
                    },
                    "tcpRetryEvents": [
                        "connection-error"
                    ]
                }
            },
            "priority": 200
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualRouterName": "serviceBhttp2"
    }
}
```

建立新的 TCP 路由

下列`create-route`範例使用 JSON 輸入檔案來建立 TCP 路由。75% 的流量會路由至名為 serviceBtcp 的虛擬節點，而 25% 的流量會路由至名為 serviceBv2tcp 的虛擬節點。為不同的目標指定不同的權重，是部署新版本應用程式的有效方法。您可以調整權重，讓 100% 的流量最終路由到具有新應用程式版本的目標。

```
aws appmesh create-route \
```

```
    --cli-input-json file://create-route-tcp.json
```

create-route-tcp.json 的內容：

```
{
    "meshName": "apps",
    "routeName": "tcpRoute",
    "spec": {
        "priority": 300,
        "tcpRoute": {
            "action": {
                "weightedTargets": [
                    {
                        "virtualNode": "serviceBtcp",
                        "weight": 75
                    },
                    {
                        "virtualNode": "serviceBv2tcp",
                        "weight": 25
                    }
                ]
            }
        }
    },
    "virtualRouterName": "serviceBtcp"
}
```

輸出：

```
{
    "route": {
        "meshName": "apps",
        "metadata": {
            "arn": "arn:aws:appmesh:us-west-2:123456789012:mesh/apps/virtualRouter/
serviceBtcp/route/tcpRoute",
            "createdAt": 1572011436.26,
            "lastUpdatedAt": 1572011436.26,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "routeName": "tcpRoute",
        "spec": {
            "priority": 300,
```

```
            "tcpRoute": {
                "action": {
                    "weightedTargets": [
                        {
                            "virtualNode": "serviceBtcp",
                            "weight": 75
                        },
                        {
                            "virtualNode": "serviceBv2tcp",
                            "weight": 25
                        }
                    ]
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualRouterName": "serviceBtcp"
    }
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的路由。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRoute。

create-virtual-gateway

以下程式碼範例顯示如何使用 create-virtual-gateway。

AWS CLI

建立新的虛擬閘道

下列create-virtual-gateway範例使用 JSON 輸入檔案，使用連接埠 9080 建立具有 HTTP 接聽程式的虛擬閘道。

```
aws appmesh create-virtual-gateway \
    --mesh-name meshName \
    --virtual-gateway-name virtualGatewayName \
    --cli-input-json file://create-virtual-gateway.json
```

create-virtual-gateway.json 的內容：

```
{
    "spec": {
      "listeners": [
        {
          "portMapping": {
            "port": 9080,
            "protocol": "http"
          }
        }
      ]
    }
}
```

輸出：

```
{
    "virtualGateway": {
        "meshName": "meshName",
        "metadata": {
            "arn": "arn:aws:appmesh:us-west-2:123456789012:mesh/meshName/
virtualGateway/virtualGatewayName",
            "createdAt": "2022-04-06T10:42:42.015000-05:00",
            "lastUpdatedAt": "2022-04-06T10:42:42.015000-05:00",
            "meshOwner": "123456789012",
            "resourceOwner": "123456789012",
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "spec": {
            "listeners": [
                {
                    "portMapping": {
                        "port": 9080,
                        "protocol": "http"
                    }
                }
            ]
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualGatewayName": "virtualGatewayName"
    }
```

```
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的虛擬閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVirtualGateway。

create-virtual-node

以下程式碼範例顯示如何使用 create-virtual-node。

AWS CLI

範例 1：建立使用 DNS 進行探索的新虛擬節點

下列create-virtual-node範例使用 JSON 輸入檔案來建立使用 DNS 進行服務探索的虛擬節點。

```
aws appmesh create-virtual-node \
    --cli-input-json file://create-virtual-node-dns.json
```

create-virtual-node-dns.json 的內容：

```
{
    "meshName": "app1",
    "spec": {
        "listeners": [
            {
                "portMapping": {
                    "port": 80,
                    "protocol": "http"
                }
            }
        ],
        "serviceDiscovery": {
            "dns": {
                "hostname": "serviceBv1.svc.cluster.local"
            }
        }
    },
    "virtualNodeName": "vnServiceBv1"
}
```

輸出：

```
{
    "virtualNode": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/
vnServiceBv1",
            "createdAt": 1563810019.874,
            "lastUpdatedAt": 1563810019.874,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "spec": {
            "listeners": [
                {
                    "portMapping": {
                        "port": 80,
                        "protocol": "http"
                    }
                }
            ],
            "serviceDiscovery": {
                "dns": {
                    "hostname": "serviceBv1.svc.cluster.local"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualNodeName": "vnServiceBv1"
    }
}
```

範例 2：建立使用 AWS Cloud Map 進行探索的新虛擬節點

下列`create-virtual-node`範例使用 JSON 輸入檔案來建立使用 AWS Cloud Map 進行服務探索的虛擬節點。

```
aws appmesh create-virtual-node \
    --cli-input-json file://create-virtual-node-cloud-map.json
```

`create-virtual-node-cloud-map.json` 的內容：

```
{
    "meshName": "app1",
    "spec": {
        "backends": [
            {
                "virtualService": {
                    "virtualServiceName": "serviceA.svc.cluster.local"
                }
            }
        ],
        "listeners": [
            {
                "portMapping": {
                    "port": 80,
                    "protocol": "http"
                }
            }
        ],
        "serviceDiscovery": {
            "awsCloudMap": {
                "attributes": [
                    {
                        "key": "Environment",
                        "value": "Testing"
                    }
                ],
                "namespaceName": "namespace1",
                "serviceName": "serviceA"
            }
        }
    },
    "virtualNodeName": "vnServiceA"
}
```

輸出：

```
{
    "virtualNode": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/
vnServiceA",
            "createdAt": 1563810859.465,
```

```
            "lastUpdatedAt": 1563810859.465,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "spec": {
            "backends": [
                {
                    "virtualService": {
                        "virtualServiceName": "serviceA.svc.cluster.local"
                    }
                }
            ],
            "listeners": [
                {
                    "portMapping": {
                        "port": 80,
                        "protocol": "http"
                    }
                }
            ],
            "serviceDiscovery": {
                "awsCloudMap": {
                    "attributes": [
                        {
                            "key": "Environment",
                            "value": "Testing"
                        }
                    ],
                    "namespaceName": "namespace1",
                    "serviceName": "serviceA"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualNodeName": "vnServiceA"
    }
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的虛擬節點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVirtualNode。

create-virtual-router

以下程式碼範例顯示如何使用 create-virtual-router。

AWS CLI

建立新的虛擬路由器

下列create-virtual-router範例使用 JSON 輸入檔案，使用連接埠 80 建立具有 HTTP 接聽程式的虛擬路由器。

```
aws appmesh create-virtual-router \
    --cli-input-json file://create-virtual-router.json
```

create-virtual-router.json 的內容：

```
{
    "meshName": "app1",
    "spec": {
        "listeners": [
            {
                "portMapping": {
                    "port": 80,
                    "protocol": "http"
                }
            }
        ]
    },
    "virtualRouterName": "vrServiceB"
}
```

輸出：

```
{
    "virtualRouter": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB",
            "createdAt": 1563810546.59,
            "lastUpdatedAt": 1563810546.59,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
```

```
            "version": 1
        },
        "spec": {
            "listeners": [
                {
                    "portMapping": {
                        "port": 80,
                        "protocol": "http"
                    }
                }
            ]
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualRouterName": "vrServiceB"
    }
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的虛擬路由器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVirtualRouter。

create-virtual-service

以下程式碼範例顯示如何使用 create-virtual-service。

AWS CLI

範例 1：使用虛擬節點供應商建立新的虛擬服務

下列create-virtual-service範例使用 JSON 輸入檔案來建立具有虛擬節點提供者的虛擬服務。

```
aws appmesh create-virtual-service \
    --cli-input-json file://create-virtual-service-virtual-node.json
```

create-virtual-service-virtual-node.json 的內容：

```
{
    "meshName": "app1",
    "spec": {
        "provider": {
```

```
            "virtualNode": {
                "virtualNodeName": "vnServiceA"
            }
        }
    },
    "virtualServiceName": "serviceA.svc.cluster.local"
}
```

輸出：

```
{
    "virtualService": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceA.svc.cluster.local",
            "createdAt": 1563810859.474,
            "lastUpdatedAt": 1563810967.179,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 2
        },
        "spec": {
            "provider": {
                "virtualNode": {
                    "virtualNodeName": "vnServiceA"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualServiceName": "serviceA.svc.cluster.local"
    }
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的虛擬節點。

範例 2：使用虛擬路由器供應商建立新的虛擬服務

下列create-virtual-service範例使用 JSON 輸入檔案，透過虛擬路由器供應商建立虛擬服務。

```
aws appmesh create-virtual-service \
```

```
    --cli-input-json file://create-virtual-service-virtual-router.json
```

create-virtual-service-virtual-router.json 的內容：

```
{
    "meshName": "app1",
    "spec": {
        "provider": {
            "virtualRouter": {
                "virtualRouterName": "vrServiceB"
            }
        }
    },
    "virtualServiceName": "serviceB.svc.cluster.local"
}
```

輸出：

```
{
    "virtualService": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceB.svc.cluster.local",
            "createdAt": 1563908363.999,
            "lastUpdatedAt": 1563908363.999,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "spec": {
            "provider": {
                "virtualRouter": {
                    "virtualRouterName": "vrServiceB"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualServiceName": "serviceB.svc.cluster.local"
    }
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的虛擬服務<https://docs.aws.amazon.com/app-mesh/latest/userguide/virtual_services.html>

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVirtualService。

delete-mesh

以下程式碼範例顯示如何使用 delete-mesh。

AWS CLI

刪除服務網格

下列delete-mesh範例會刪除指定的服務網格。

```
aws appmesh delete-mesh \
    --mesh-name app1
```

輸出：

```
{
    "mesh": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
            "createdAt": 1563809909.282,
            "lastUpdatedAt": 1563824981.248,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 2
        },
        "spec": {
            "egressFilter": {
                "type": "ALLOW_ALL"
            }
        },
        "status": {
            "status": "DELETED"
        }
    }
}
```

如需詳細資訊，請參閱《App Mesh 使用者指南》中的 Service Meshes。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteMesh。

delete-route

以下程式碼範例顯示如何使用 delete-route。

AWS CLI

刪除路由

下列delete-route範例會刪除指定的路由。

```
aws appmesh delete-route \
    --mesh-name app1 \
    --virtual-router-name vrServiceB \
    --route-name toVnServiceB-weighted
```

輸出：

```
{
    "route": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB/route/toVnServiceB-weighted",
            "createdAt": 1563811384.015,
            "lastUpdatedAt": 1563823915.936,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 3
        },
        "routeName": "toVnServiceB-weighted",
        "spec": {
            "httpRoute": {
                "action": {
                    "weightedTargets": [
                        {
                            "virtualNode": "vnServiceBv1",
                            "weight": 80
                        },
                        {
                            "virtualNode": "vnServiceBv2",
                            "weight": 20
```

```
                        }
                    ]
                },
                "match": {
                    "prefix": "/"
                }
            }
        },
        "status": {
            "status": "DELETED"
        },
        "virtualRouterName": "vrServiceB"
    }
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的路由。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRoute。

delete-virtual-node

以下程式碼範例顯示如何使用 delete-virtual-node。

AWS CLI

刪除虛擬節點

下列delete-virtual-node範例會刪除指定的虛擬節點。

```
aws appmesh delete-virtual-node \
    --mesh-name app1 \
    --virtual-node-name vnServiceBv2
```

輸出：

```
{
    "virtualNode": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/
vnServiceBv2",
            "createdAt": 1563810117.297,
            "lastUpdatedAt": 1563824700.678,
```

```
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 2
        },
        "spec": {
            "backends": [],
            "listeners": [
                {
                    "portMapping": {
                        "port": 80,
                        "protocol": "http"
                    }
                }
            ],
            "serviceDiscovery": {
                "dns": {
                    "hostname": "serviceBv2.svc.cluster.local"
                }
            }
        },
        "status": {
            "status": "DELETED"
        },
        "virtualNodeName": "vnServiceBv2"
    }
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的虛擬節點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVirtualNode。

delete-virtual-router

以下程式碼範例顯示如何使用 delete-virtual-router。

AWS CLI

刪除虛擬路由器

下列delete-virtual-router範例會刪除指定的虛擬路由器。

```
aws appmesh delete-virtual-router \
    --mesh-name app1 \
    --virtual-router-name vrServiceB
```

輸出：

```
{
    "virtualRouter": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB",
            "createdAt": 1563810546.59,
            "lastUpdatedAt": 1563824253.467,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 3
        },
        "spec": {
            "listeners": [
                {
                    "portMapping": {
                        "port": 80,
                        "protocol": "http"
                    }
                }
            ]
        },
        "status": {
            "status": "DELETED"
        },
        "virtualRouterName": "vrServiceB"
    }
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的虛擬路由器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVirtualRouter。

delete-virtual-service

以下程式碼範例顯示如何使用 delete-virtual-service。

AWS CLI

刪除虛擬服務

下列delete-virtual-service範例會刪除指定的虛擬服務。

```
aws appmesh delete-virtual-service \
    --mesh-name app1 \
    --virtual-service-name serviceB.svc.cluster.local
```

輸出：

```
{
    "virtualService": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceB.svc.cluster.local",
            "createdAt": 1563908363.999,
            "lastUpdatedAt": 1563913940.866,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 3
        },
        "spec": {},
        "status": {
            "status": "DELETED"
        },
        "virtualServiceName": "serviceB.svc.cluster.local"
    }
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的虛擬服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVirtualService。

describe-mesh

以下程式碼範例顯示如何使用 describe-mesh。

AWS CLI

描述服務網格

下列describe-mesh範例會傳回指定服務網格的詳細資訊。

```
aws appmesh describe-mesh \
    --mesh-name app1
```

輸出：

```
{
    "mesh": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
            "createdAt": 1563809909.282,
            "lastUpdatedAt": 1563809909.282,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "spec": {},
        "status": {
            "status": "ACTIVE"
        }
    }
}
```

如需詳細資訊，請參閱《App Mesh 使用者指南》中的 Service Meshes。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeMesh。

describe-route

以下程式碼範例顯示如何使用 describe-route。

AWS CLI

描述路由

下列describe-route範例會傳回指定路由的詳細資訊。

```
aws appmesh describe-route \
    --mesh-name app1 \
    --virtual-router-name vrServiceB \
    --route-name toVnServiceB-weighted
```

輸出：

```
{
    "route": {
```

```
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB/route/toVnServiceB-weighted",
            "createdAt": 1563811384.015,
            "lastUpdatedAt": 1563811384.015,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "routeName": "toVnServiceB-weighted",
        "spec": {
            "httpRoute": {
                "action": {
                    "weightedTargets": [
                        {
                            "virtualNode": "vnServiceBv1",
                            "weight": 90
                        },
                        {
                            "virtualNode": "vnServiceBv2",
                            "weight": 10
                        }
                    ]
                },
                "match": {
                    "prefix": "/"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualRouterName": "vrServiceB"
    }
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的路由。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRoute。

describe-virtual-node

以下程式碼範例顯示如何使用 describe-virtual-node。

AWS CLI

描述虛擬節點

下列describe-virtual-node範例會傳回指定虛擬節點的詳細資訊。

```
aws appmesh describe-virtual-node \
    --mesh-name app1 \
    --virtual-node-name vnServiceBv1
```

輸出：

```
{
    "virtualNode": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/
vnServiceBv1",
            "createdAt": 1563810019.874,
            "lastUpdatedAt": 1563810019.874,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "spec": {
            "backends": [],
            "listeners": [
                {
                    "portMapping": {
                        "port": 80,
                        "protocol": "http"
                    }
                }
            ],
            "serviceDiscovery": {
                "dns": {
                    "hostname": "serviceBv1.svc.cluster.local"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualNodeName": "vnServiceBv1"
```

```
    }
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的虛擬節點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVirtualNode。

describe-virtual-router

以下程式碼範例顯示如何使用 describe-virtual-router。

AWS CLI

描述虛擬路由器

下列describe-virtual-router範例會傳回指定虛擬路由器的詳細資訊。

```
aws appmesh describe-virtual-router \
    --mesh-name app1 \
    --virtual-router-name vrServiceB
```

輸出：

```
{
    "virtualRouter": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB",
            "createdAt": 1563810546.59,
            "lastUpdatedAt": 1563810546.59,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "spec": {
            "listeners": [
                {
                    "portMapping": {
                        "port": 80,
                        "protocol": "http"
                    }
                }
```

```
            ]
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualRouterName": "vrServiceB"
    }
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的虛擬路由器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVirtualRouter。

describe-virtual-service

以下程式碼範例顯示如何使用 describe-virtual-service。

AWS CLI

描述虛擬服務

下列describe-virtual-service範例會傳回指定虛擬服務的詳細資訊。

```
aws appmesh describe-virtual-service \
    --mesh-name app1 \
    --virtual-service-name serviceB.svc.cluster.local
```

輸出：

```
{
    "virtualService": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceB.svc.cluster.local",
            "createdAt": 1563908363.999,
            "lastUpdatedAt": 1563908363.999,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 1
        },
        "spec": {
            "provider": {
```

```
                "virtualRouter": {
                    "virtualRouterName": "vrServiceB"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualServiceName": "serviceB.svc.cluster.local"
    }
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的虛擬服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVirtualService。

`list-meshes`

以下程式碼範例顯示如何使用 `list-meshes`。

AWS CLI

列出服務網格

下列`list-meshes`範例列出目前區域中的所有服務網格 AWS 。

```
aws appmesh list-meshes
```

輸出：

```
{
    "meshes": [
        {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
            "meshName": "app1"
        }
    ]
}
```

如需詳細資訊，請參閱《App Mesh 使用者指南》中的 Service Meshes。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMeshes。

list-routes

以下程式碼範例顯示如何使用 list-routes。

AWS CLI

列出路由

下列list-routes範例列出指定虛擬路由器的所有路由。

```
aws appmesh list-routes \
    --mesh-name app1 \
    --virtual-router-name vrServiceB
```

輸出：

```
{
    "routes": [
        {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB/route/toVnServiceB",
            "meshName": "app1",
            "routeName": "toVnServiceB-weighted",
            "virtualRouterName": "vrServiceB"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的路由。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRoutes。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出資源的標籤

下列list-tags-for-resource範例會列出指派給指定資源的所有標籤。

```
aws appmesh list-tags-for-resource \
    --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1
```

輸出：

```
{
    "tags": [
        {
            "key": "key1",
            "value": "value1"
        },
        {
            "key": "key2",
            "value": "value2"
        },
        {
            "key": "key3",
            "value": "value3"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

list-virtual-nodes

以下程式碼範例顯示如何使用 list-virtual-nodes。

AWS CLI

列出虛擬節點

下列list-virtual-nodes範例列出指定服務網格中的所有虛擬節點。

```
aws appmesh list-virtual-nodes \
    --mesh-name app1
```

輸出：

```
{
```

```
    "virtualNodes": [
        {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/
vnServiceBv1",
            "meshName": "app1",
            "virtualNodeName": "vnServiceBv1"
        },
        {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/
vnServiceBv2",
            "meshName": "app1",
            "virtualNodeName": "vnServiceBv2"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的虛擬節點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListVirtualNodes。

list-virtual-routers

以下程式碼範例顯示如何使用 `list-virtual-routers`。

AWS CLI

列出虛擬路由器

下列`list-virtual-routers`範例列出指定服務網格中的所有虛擬路由器。

```
aws appmesh list-virtual-routers \
    --mesh-name app1
```

輸出：

```
{
    "virtualRouters": [
        {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB",
            "meshName": "app1",
            "virtualRouterName": "vrServiceB"
```

```
        }
    ]
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的虛擬路由器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListVirtualRouters。

`list-virtual-services`

以下程式碼範例顯示如何使用 `list-virtual-services`。

AWS CLI

列出虛擬服務

下列`list-virtual-services`範例列出指定服務網格中的所有虛擬服務。

```
aws appmesh list-virtual-services \
    --mesh-name app1
```

輸出：

```
{
    "virtualServices": [
        {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceA.svc.cluster.local",
            "meshName": "app1",
            "virtualServiceName": "serviceA.svc.cluster.local"
        },
        {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceB.svc.cluster.local",
            "meshName": "app1",
            "virtualServiceName": "serviceB.svc.cluster.local"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的虛擬服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListVirtualServices。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

標記資源

下列tag-resource範例會將key1具有 值value1的標籤新增至指定的資源。

```
aws appmesh tag-resource \
    --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1 \
    --tags key=key1,value=value1
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

取消標記資源

下列untag-resource範例key1會從指定的資源移除具有 金鑰的標籤。

```
aws appmesh untag-resource \
    --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1 \
    --tag-keys key1
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-mesh

以下程式碼範例顯示如何使用 update-mesh。

AWS CLI

更新服務網格

下列update-mesh範例使用 JSON 輸入檔案來更新服務網格，以允許透過 Envoy 代理轉送所有外部輸出流量。

```
aws appmesh update-mesh \
    --cli-input-json file://update-mesh.json
```

update-mesh.json 的內容：

```
{
    "meshName": "app1",
    "spec": {
        "egressFilter": {
            "type": "ALLOW_ALL"
        }
    }
}
```

輸出：

```
{
    "mesh": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
            "createdAt": 1563809909.282,
            "lastUpdatedAt": 1563812829.687,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 2
        },
        "spec": {
            "egressFilter": {
                "type": "ALLOW_ALL"
            }
        },
        "status": {
            "status": "ACTIVE"
        }
    }
```

```
}
```

如需詳細資訊，請參閱《App Mesh 使用者指南》中的 Service Meshes。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateMesh。

update-route

以下程式碼範例顯示如何使用 update-route。

AWS CLI

更新路由

下列update-route範例使用 JSON 輸入檔案來更新路由的權重。

```
aws appmesh update-route \
    --cli-input-json file://update-route-weighted.json
```

update-route-weighted.json 的內容：

```
{
    "meshName": "app1",
    "routeName": "toVnServiceB-weighted",
    "spec": {
        "httpRoute": {
            "action": {
                "weightedTargets": [
                    {
                        "virtualNode": "vnServiceBv1",
                        "weight": 80
                    },
                    {
                        "virtualNode": "vnServiceBv2",
                        "weight": 20
                    }
                ]
            },
            "match": {
                "prefix": "/"
            }
        }
```

```
    },
    "virtualRouterName": "vrServiceB"
}
```

輸出：

```
{
    "route": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB/route/toVnServiceB-weighted",
            "createdAt": 1563811384.015,
            "lastUpdatedAt": 1563819600.022,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 2
        },
        "routeName": "toVnServiceB-weighted",
        "spec": {
            "httpRoute": {
                "action": {
                    "weightedTargets": [
                        {
                            "virtualNode": "vnServiceBv1",
                            "weight": 80
                        },
                        {
                            "virtualNode": "vnServiceBv2",
                            "weight": 20
                        }
                    ]
                },
                "match": {
                    "prefix": "/"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualRouterName": "vrServiceB"
    }
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的路由。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRoute。

update-virtual-node

以下程式碼範例顯示如何使用 update-virtual-node。

AWS CLI

更新虛擬節點

下列update-virtual-node範例使用 JSON 輸入檔案，將運作狀態檢查新增至虛擬節點。

```
aws appmesh update-virtual-node \
    --cli-input-json file://update-virtual-node.json
```

update-virtual-node.json 的內容：

```
{
    "clientToken": "500",
    "meshName": "app1",
    "spec": {
        "listeners": [
            {
                "healthCheck": {
                    "healthyThreshold": 5,
                    "intervalMillis": 10000,
                    "path": "/",
                    "port": 80,
                    "protocol": "http",
                    "timeoutMillis": 3000,
                    "unhealthyThreshold": 3
                },
                "portMapping": {
                    "port": 80,
                    "protocol": "http"
                }
            }
        ],
        "serviceDiscovery": {
            "dns": {
                "hostname": "serviceBv1.svc.cluster.local"
```

```
            }
        }
    },
    "virtualNodeName": "vnServiceBv1"
}
```

輸出：

```
{
    "virtualNode": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/
vnServiceBv1",
            "createdAt": 1563810019.874,
            "lastUpdatedAt": 1563819234.825,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 2
        },
        "spec": {
            "listeners": [
                {
                    "healthCheck": {
                        "healthyThreshold": 5,
                        "intervalMillis": 10000,
                        "path": "/",
                        "port": 80,
                        "protocol": "http",
                        "timeoutMillis": 3000,
                        "unhealthyThreshold": 3
                    },
                    "portMapping": {
                        "port": 80,
                        "protocol": "http"
                    }
                }
            ],
            "serviceDiscovery": {
                "dns": {
                    "hostname": "serviceBv1.svc.cluster.local"
                }
            }
        },
```

```
        "status": {
            "status": "ACTIVE"
        },
        "virtualNodeName": "vnServiceBv1"
    }
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的虛擬節點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateVirtualNode。

update-virtual-router

以下程式碼範例顯示如何使用 update-virtual-router。

AWS CLI

更新虛擬路由器

下列update-virtual-router範例使用 JSON 輸入檔案來更新虛擬路由器接聽程式連接埠。

```
aws appmesh update-virtual-router \
    --cli-input-json file://update-virtual-router.json
```

update-virtual-router.json 的內容：

```
{
    "meshName": "app1",
    "spec": {
        "listeners": [
            {
                "portMapping": {
                    "port": 8080,
                    "protocol": "http"
                }
            }
        ]
    },
    "virtualRouterName": "vrServiceB"
}
```

輸出：

```
{
    "virtualRouter": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB",
            "createdAt": 1563810546.59,
            "lastUpdatedAt": 1563819431.352,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 2
        },
        "spec": {
            "listeners": [
                {
                    "portMapping": {
                        "port": 8080,
                        "protocol": "http"
                    }
                }
            ]
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualRouterName": "vrServiceB"
    }
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的虛擬路由器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateVirtualRouter。

update-virtual-service

以下程式碼範例顯示如何使用 update-virtual-service。

AWS CLI

更新虛擬服務

下列update-virtual-service範例使用 JSON 輸入檔案來更新虛擬服務，以使用虛擬路由器供應商。

```
aws appmesh update-virtual-service \
    --cli-input-json file://update-virtual-service.json
```

update-virtual-service.json 的內容：

```
{
    "meshName": "app1",
    "spec": {
        "provider": {
            "virtualRouter": {
                "virtualRouterName": "vrServiceA"
            }
        }
    },
    "virtualServiceName": "serviceA.svc.cluster.local"
}
```

輸出：

```
{
    "virtualService": {
        "meshName": "app1",
        "metadata": {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceA.svc.cluster.local",
            "createdAt": 1563810859.474,
            "lastUpdatedAt": 1563820257.411,
            "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "version": 3
        },
        "spec": {
            "provider": {
                "virtualRouter": {
                    "virtualRouterName": "vrServiceA"
                }
            }
        },
        "status": {
            "status": "ACTIVE"
        },
        "virtualServiceName": "serviceA.svc.cluster.local"
    }
```

```
}
```

如需詳細資訊，請參閱《AWS App Mesh 使用者指南》中的虛擬服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateVirtualService。

使用 的 App Runner 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 App Runner 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-custom-domain`

以下程式碼範例顯示如何使用 `associate-custom-domain`。

AWS CLI

將網域名稱和 www 子網域與服務建立關聯

下列`associate-custom-domain`範例會將您控制的自訂網域名稱與 App Runner 服務建立關聯。網域名稱是根網域 `example.com`，包括特殊大小寫子網域 `www.example.com`。

```
aws apprunner associate-custom-domain \
    --cli-input-json file://input.json
```

`input.json` 的內容：

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
```

```
    "DomainName": "example.com",
    "EnableWWWSubdomain": true
}
```

輸出：

```
{
    "CustomDomain": {
        "CertificateValidationRecords": [
            {
                "Name": "_70d3f50a94f7c72dc28784cf55db2f6b.example.com",
                "Status": "PENDING_VALIDATION",
                "Type": "CNAME",
                "Value": "_1270c137383c6307b6832db02504c4b0.bsgbmzkfwj.acm-
validations.aws."
            },
            {
                "Name": "_287870d3f50a94f7c72dc4cf55db2f6b.www.example.com",
                "Status": "PENDING_VALIDATION",
                "Type": "CNAME",
                "Value": "_832db01270c137383c6307b62504c4b0.mzkbsgbfwj.acm-
validations.aws."
            }
        ],
        "DomainName": "example.com",
        "EnableWWWSubdomain": true,
        "Status": "CREATING"
    },
    "DNSTarget": "psbqam834h.us-east-1.awsapprunner.com",
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateCustomDomain。

create-auto-scaling-configuration

以下程式碼範例顯示如何使用 create-auto-scaling-configuration。

AWS CLI

建立高可用性自動擴展組態

下列create-auto-scaling-configuration範例會將 MinSize設定為 5，以建立針對高可用性最佳化的自動擴展組態。透過此組態，App Runner 會嘗試將您的服務執行個體分散到最多的可用區域，最多五個，視 AWS 區域而定。

呼叫會傳回AutoScalingConfiguration物件，並將其他設定設為其預設值。在此範例中，這是建立名為 之組態的第一個呼叫high-availability。修訂版設定為 1，它是最新的修訂版。

```
aws apprunner create-auto-scaling-configuration \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "AutoScalingConfigurationName": "high-availability",
    "MinSize": 5
}
```

輸出：

```
{
    "AutoScalingConfiguration": {
        "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-
availability/1/2f50e7656d7819fead0f59672e68042e",
        "AutoScalingConfigurationName": "high-availability",
        "AutoScalingConfigurationRevision": 1,
        "CreatedAt": "2020-11-03T00:29:17Z",
        "Latest": true,
        "Status": "ACTIVE",
        "MaxConcurrency": 100,
        "MaxSize": 50,
        "MinSize": 5
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAutoScalingConfiguration。

create-connection

以下程式碼範例顯示如何使用 create-connection。

AWS CLI

建立 GitHub 連線

下列create-connection範例會建立私有 GitHub 程式碼儲存庫的連線。成功呼叫後的連線狀態為 PENDING_HANDSHAKE。這是因為與供應商的身分驗證交握尚未發生。使用 App Runner 主控台完成交握。

```
aws apprunner create-connection \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "ConnectionName": "my-github-connection",
    "ProviderType": "GITHUB"
}
```

輸出：

```
{
    "Connection": {
        "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/my-
github-connection",
        "ConnectionName": "my-github-connection",
        "Status": "PENDING_HANDSHAKE",
        "CreatedAt": "2020-11-03T00:32:51Z",
        "ProviderType": "GITHUB"
    }
}
```

如需詳細資訊，請參閱《App Runner 開發人員指南》中的管理 App Runner 連線。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateConnection。

create-service

以下程式碼範例顯示如何使用 create-service。

AWS CLI

範例 1：建立原始碼儲存庫服務

下列create-service範例會根據 Python 原始程式碼儲存庫建立 App Runner 服務。

```
aws apprunner create-service \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "ServiceName": "python-app",
    "SourceConfiguration": {
        "AuthenticationConfiguration": {
            "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-connection/e7656250f67242d7819feade6800f59e"
        },
        "AutoDeploymentsEnabled": true,
        "CodeRepository": {
            "RepositoryUrl": "https://github.com/my-account/python-hello",
            "SourceCodeVersion": {
                "Type": "BRANCH",
                "Value": "main"
            },
            "CodeConfiguration": {
                "ConfigurationSource": "API",
                "CodeConfigurationValues": {
                    "Runtime": "PYTHON_3",
                    "BuildCommand": "pip install -r requirements.txt",
                    "StartCommand": "python server.py",
                    "Port": "8080",
                    "RuntimeEnvironmentVariables": [
                        {
                            "NAME": "Jane"
                        }
                    ]
                }
            }
        }
    },
    "InstanceConfiguration": {
        "CPU": "1 vCPU",
        "Memory": "3 GB"
    }
}
```

輸出：

```
{
    "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
    "Service": {
        "CreatedAt": "2020-11-20T19:05:25Z",
        "UpdatedAt": "2020-11-20T19:05:25Z",
        "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceName": "python-app",
        "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
        "SourceConfiguration": {
            "AuthenticationConfiguration": {
                "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
            },
            "AutoDeploymentsEnabled": true,
            "CodeRepository": {
                "CodeConfiguration": {
                    "CodeConfigurationValues": {
                        "BuildCommand": "pip install -r requirements.txt",
                        "Port": "8080",
                        "Runtime": "PYTHON_3",
                        "RuntimeEnvironmentVariables": [
                            {
                                "NAME": "Jane"
                            }
                        ],
                        "StartCommand": "python server.py"
                    },
                    "ConfigurationSource": "Api"
                },
                "RepositoryUrl": "https://github.com/my-account/python-hello",
                "SourceCodeVersion": {
                    "Type": "BRANCH",
                    "Value": "main"
                }
            }
        },
        "Status": "OPERATION_IN_PROGRESS",
        "InstanceConfiguration": {
            "CPU": "1 vCPU",
```

```
            "Memory": "3 GB"
        }
    }
}
```

範例 2：建立原始程式碼儲存庫服務

下列create-service範例會根據 Python 原始程式碼儲存庫建立 App Runner 服務。

```
aws apprunner create-service \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "ServiceName": "python-app",
    "SourceConfiguration": {
        "AuthenticationConfiguration": {
            "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-connection/e7656250f67242d7819feade6800f59e"
        },
        "AutoDeploymentsEnabled": true,
        "CodeRepository": {
            "RepositoryUrl": "https://github.com/my-account/python-hello",
            "SourceCodeVersion": {
                "Type": "BRANCH",
                "Value": "main"
            },
            "CodeConfiguration": {
                "ConfigurationSource": "API",
                "CodeConfigurationValues": {
                    "Runtime": "PYTHON_3",
                    "BuildCommand": "pip install -r requirements.txt",
                    "StartCommand": "python server.py",
                    "Port": "8080",
                    "RuntimeEnvironmentVariables": [
                        {
                            "NAME": "Jane"
                        }
                    ]
                }
            }
        }
```

```
    },
    "InstanceConfiguration": {
        "CPU": "1 vCPU",
        "Memory": "3 GB"
    }
}
```

輸出：

```
{
    "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
    "Service": {
        "CreatedAt": "2020-11-20T19:05:25Z",
        "UpdatedAt": "2020-11-20T19:05:25Z",
        "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceName": "python-app",
        "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
        "SourceConfiguration": {
            "AuthenticationConfiguration": {
                "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
            },
            "AutoDeploymentsEnabled": true,
            "CodeRepository": {
                "CodeConfiguration": {
                    "CodeConfigurationValues": {
                        "BuildCommand": "pip install -r requirements.txt",
                        "Port": "8080",
                        "Runtime": "PYTHON_3",
                        "RuntimeEnvironmentVariables": [
                            {
                                "NAME": "Jane"
                            }
                        ],
                        "StartCommand": "python server.py"
                    },
                    "ConfigurationSource": "Api"
                },
                "RepositoryUrl": "https://github.com/my-account/python-hello",
                "SourceCodeVersion": {
```

```
                    "Type": "BRANCH",
                    "Value": "main"
                }
            }
        },
        "Status": "OPERATION_IN_PROGRESS",
        "InstanceConfiguration": {
            "CPU": "1 vCPU",
            "Memory": "3 GB"
        }
    }
}
```

範例 3：建立來源映像儲存庫服務

下列`create-service`範例會根據儲存在 Elastic Container Registry (ECR) 中的映像來建立 App Runner 服務。

```
aws apprunner create-service \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "ServiceName": "golang-container-app",
    "SourceConfiguration": {
        "AuthenticationConfiguration": {
            "AccessRoleArn": "arn:aws:iam::123456789012:role/my-ecr-role"
        },
        "AutoDeploymentsEnabled": true,
        "ImageRepository": {
            "ImageIdentifier": "123456789012.dkr.ecr.us-east-1.amazonaws.com/golang-
app:latest",
            "ImageConfiguration": {
                "Port": "8080",
                "RuntimeEnvironmentVariables": [
                    {
                        "NAME": "Jane"
                    }
                ]
            },
            "ImageRepositoryType": "ECR"
        }
```

```
    },
    "InstanceConfiguration": {
        "CPU": "1 vCPU",
        "Memory": "3 GB"
    }
}
```

輸出：

```
{
    "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
    "Service": {
        "CreatedAt": "2020-11-06T23:15:30Z",
        "UpdatedAt": "2020-11-06T23:15:30Z",
        "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/golang-
container-app/51728f8a20ce46d39b25398a6c8e9d1a",
        "ServiceId": "51728f8a20ce46d39b25398a6c8e9d1a",
        "ServiceName": "golang-container-app",
        "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
        "SourceConfiguration": {
            "AuthenticationConfiguration": {
                "AccessRoleArn": "arn:aws:iam::123456789012:role/my-ecr-role"
            },
            "AutoDeploymentsEnabled": true,
            "ImageRepository": {
                "ImageIdentifier": "123456789012.dkr.ecr.us-east-1.amazonaws.com/
golang-app:latest",
                "ImageConfiguration": {
                    "Port": "8080",
                    "RuntimeEnvironmentVariables": [
                        {
                            "NAME": "Jane"
                        }
                    ]
                },
                "ImageRepositoryType": "ECR"
            }
        },
        "Status": "OPERATION_IN_PROGRESS",
        "InstanceConfiguration": {
            "CPU": "1 vCPU",
            "Memory": "3 GB"
        }
```

```
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateService。

delete-auto-scaling-configuration

以下程式碼範例顯示如何使用 delete-auto-scaling-configuration。

AWS CLI

範例 1：刪除自動擴展組態的最新作用中修訂

下列delete-auto-scaling-configuration範例會刪除 App Runner 自動擴展組態的最新作用中修訂。若要刪除最新的作用中修訂，請指定以組態名稱結尾的 Amazon Resource Name (ARN)，而不指定修訂元件。

在此範例中，有兩個修訂在此動作之前。因此，刪除修訂版 2 （最新）。不過，它現在會顯示 "Latest": false，因為刪除之後，它不再是最新的作用中修訂。

```
aws apprunner delete-auto-scaling-configuration \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability"
}
```

輸出：

```
{
    "AutoScalingConfiguration": {
        "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/2/
e76562f50d78042e819fead0f59672e6",
        "AutoScalingConfigurationName": "high-availability",
        "AutoScalingConfigurationRevision": 2,
        "CreatedAt": "2021-02-25T17:42:59Z",
```

```
        "DeletedAt": "2021-03-02T08:07:06Z",
        "Latest": false,
        "Status": "INACTIVE",
        "MaxConcurrency": 30,
        "MaxSize": 90,
        "MinSize": 5
    }
}
```

範例 2：刪除自動擴展組態的特定修訂

下列delete-auto-scaling-configuration範例會刪除 App Runner 自動擴展組態的特定修訂。若要刪除特定修訂，請指定包含修訂編號的 ARN。

在此範例中，此動作之前存在數個修訂。動作會刪除修訂版 1。

```
aws apprunner delete-auto-scaling-configuration \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/1"
}
```

輸出：

```
{
    "AutoScalingConfiguration": {
        "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-
availability/1/2f50e7656d7819fead0f59672e68042e",
        "AutoScalingConfigurationName": "high-availability",
        "AutoScalingConfigurationRevision": 1,
        "CreatedAt": "2020-11-03T00:29:17Z",
        "DeletedAt": "2021-03-02T08:07:06Z",
        "Latest": false,
        "Status": "INACTIVE",
        "MaxConcurrency": 100,
        "MaxSize": 50,
        "MinSize": 5
```

```
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAutoScalingConfiguration。

`delete-connection`

以下程式碼範例顯示如何使用 `delete-connection`。

AWS CLI

刪除連線

下列`delete-connection`範例會刪除 App Runner 連線。成功呼叫後的連線狀態為 `DELETED`。這是因為連線不再可用。

```
aws apprunner delete-connection \
    --cli-input-json file://input.json
```

`input.json` 的內容：

```
{
    "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/my-github-connection"
}
```

輸出：

```
{
    "Connection": {
        "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/my-github-connection",
        "ConnectionName": "my-github-connection",
        "Status": "DELETED",
        "CreatedAt": "2020-11-03T00:32:51Z",
        "ProviderType": "GITHUB"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteConnection。

delete-service

以下程式碼範例顯示如何使用 delete-service。

AWS CLI

刪除服務

下列delete-service範例會刪除 App Runner 服務。

```
aws apprunner delete-service \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

輸出：

```
{
    "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
    "Service": {
        "CreatedAt": "2020-11-20T19:05:25Z",
        "UpdatedAt": "2020-11-20T19:05:25Z",
        "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceName": "python-app",
        "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
        "SourceConfiguration": {
            "AuthenticationConfiguration": {
                "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
            },
            "AutoDeploymentsEnabled": true,
            "CodeRepository": {
                "CodeConfiguration": {
                    "CodeConfigurationValues": {
```

```
                        "BuildCommand": "pip install -r requirements.txt",
                        "Port": "8080",
                        "Runtime": "PYTHON_3",
                        "RuntimeEnvironmentVariables": [
                            {
                                "NAME": "Jane"
                            }
                        ],
                        "StartCommand": "python server.py"
                    },
                    "ConfigurationSource": "Api"
                },
                "RepositoryUrl": "https://github.com/my-account/python-hello",
                "SourceCodeVersion": {
                    "Type": "BRANCH",
                    "Value": "main"
                }
            }
        },
        "Status": "OPERATION_IN_PROGRESS",
        "InstanceConfiguration": {
            "CPU": "1 vCPU",
            "Memory": "3 GB"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteService。

describe-auto-scaling-configuration

以下程式碼範例顯示如何使用 describe-auto-scaling-configuration。

AWS CLI

範例 1：描述自動擴展組態的最新作用中修訂

下列describe-auto-scaling-configuration範例會取得 App Runner 自動擴展組態的最新作用中修訂的描述。若要描述最新的作用中修訂，請指定以組態名稱結尾的 ARN，而不指定修訂元件。

在此範例中，有兩個修訂。因此，會說明修訂版 2（最新）。產生的物件會顯示 "Latest": true。

```
aws apprunner describe-auto-scaling-configuration \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability"
}
```

輸出：

```
{
    "AutoScalingConfiguration": {
        "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/2/
e76562f50d78042e819fead0f59672e6",
        "AutoScalingConfigurationName": "high-availability",
        "AutoScalingConfigurationRevision": 2,
        "CreatedAt": "2021-02-25T17:42:59Z",
        "Latest": true,
        "Status": "ACTIVE",
        "MaxConcurrency": 30,
        "MaxSize": 90,
        "MinSize": 5
    }
}
```

範例 2：描述自動擴展組態的特定修訂

下列describe-auto-scaling-configuration範例會取得 App Runner 自動擴展組態特定修訂的說明。若要描述特定修訂，請指定包含修訂編號的 ARN。

在此範例中，存在多個修訂，並1查詢修訂。產生的物件會顯示 "Latest": false。

```
aws apprunner describe-auto-scaling-configuration \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
```

```
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/1"
}
```

輸出：

```
{
    "AutoScalingConfiguration": {
        "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-
availability/1/2f50e7656d7819fead0f59672e68042e",
        "AutoScalingConfigurationName": "high-availability",
        "AutoScalingConfigurationRevision": 1,
        "CreatedAt": "2020-11-03T00:29:17Z",
        "Latest": false,
        "Status": "ACTIVE",
        "MaxConcurrency": 100,
        "MaxSize": 50,
        "MinSize": 5
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAutoScalingConfiguration。

describe-custom-domains

以下程式碼範例顯示如何使用 describe-custom-domains。

AWS CLI

取得與服務相關聯的自訂網域名稱描述

下列describe-custom-domains範例會取得與 App Runner 服務相關聯之自訂網域名稱的描述和狀態。

```
aws apprunner describe-custom-domains \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
```

```
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "DomainName": "example.com",
    "EnableWWWSubdomain": true
}
```

輸出：

```
{
    "CustomDomains": [
        {
            "CertificateValidationRecords": [
                {
                    "Name": "_70d3f50a94f7c72dc28784cf55db2f6b.example.com",
                    "Status": "PENDING_VALIDATION",
                    "Type": "CNAME",
                    "Value": "_1270c137383c6307b6832db02504c4b0.bsgbmzkfwj.acm-
validations.aws."
                },
                {
                    "Name": "_287870d3f50a94f7c72dc4cf55db2f6b.www.example.com",
                    "Status": "PENDING_VALIDATION",
                    "Type": "CNAME",
                    "Value": "_832db01270c137383c6307b62504c4b0.mzkbsgbfwj.acm-
validations.aws."
                }
            ],
            "DomainName": "example.com",
            "EnableWWWSubdomain": true,
            "Status": "PENDING_CERTIFICATE_DNS_VALIDATION"
        },
        {
            "CertificateValidationRecords": [
                {
                    "Name": "_a94f784c70d3f507c72dc28f55db2f6b.deals.example.com",
                    "Status": "SUCCESS",
                    "Type": "CNAME",
                    "Value": "_2db02504c1270c137383c6307b6834b0.bsgbmzkfwj.acm-
validations.aws."
                }
            ],
            "DomainName": "deals.example.com",
            "EnableWWWSubdomain": false,
```

```
            "Status": "ACTIVE"
        }
    ],
    "DNSTarget": "psbqam834h.us-east-1.awsapprunner.com",
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCustomDomains。

describe-service

以下程式碼範例顯示如何使用 describe-service。

AWS CLI

描述服務

下列describe-service範例會取得 App Runner 服務的描述。

```
aws apprunner describe-service \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

輸出：

```
{
    "Service": {
        "CreatedAt": "2020-11-20T19:05:25Z",
        "UpdatedAt": "2020-11-20T19:05:25Z",
        "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceName": "python-app",
        "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
```

```
        "SourceConfiguration": {
            "AuthenticationConfiguration": {
                "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
            },
            "AutoDeploymentsEnabled": true,
            "CodeRepository": {
                "CodeConfiguration": {
                    "CodeConfigurationValues": {
                        "BuildCommand": "pip install -r requirements.txt",
                        "Port": "8080",
                        "Runtime": "PYTHON_3",
                        "RuntimeEnvironmentVariables": [
                            {
                                "NAME": "Jane"
                            }
                        ],
                        "StartCommand": "python server.py"
                    },
                    "ConfigurationSource": "Api"
                },
                "RepositoryUrl": "https://github.com/my-account/python-hello",
                "SourceCodeVersion": {
                    "Type": "BRANCH",
                    "Value": "main"
                }
            }
        },
        "Status": "RUNNING",
        "InstanceConfiguration": {
            "CPU": "1 vCPU",
            "Memory": "3 GB"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeService。

disassociate-custom-domain

以下程式碼範例顯示如何使用 disassociate-custom-domain。

AWS CLI

取消網域名稱與服務的關聯

下列disassociate-custom-domain範例會取消網域example.com與 App Runner 服務的關聯。呼叫也會取消與根網域www.example.com相關聯的子網域關聯。

```
aws apprunner disassociate-custom-domain \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "DomainName": "example.com"
}
```

輸出：

```
{
    "CustomDomain": {
        "CertificateValidationRecords": [
            {
                "Name": "_70d3f50a94f7c72dc28784cf55db2f6b.example.com",
                "Status": "PENDING_VALIDATION",
                "Type": "CNAME",
                "Value": "_1270c137383c6307b6832db02504c4b0.bsgbmzkfwj.acm-
validations.aws."
            },
            {
                "Name": "_287870d3f50a94f7c72dc4cf55db2f6b.www.example.com",
                "Status": "PENDING_VALIDATION",
                "Type": "CNAME",
                "Value": "_832db01270c137383c6307b62504c4b0.mzkbsgbfwj.acm-
validations.aws."
            }
        ],
        "DomainName": "example.com",
        "EnableWWWSubdomain": true,
        "Status": "DELETING"
    },
```

```
    "DNSTarget": "psbqam834h.us-east-1.awsapprunner.com",
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateCustomDomain。

`list-auto-scaling-configurations`

以下程式碼範例顯示如何使用 `list-auto-scaling-configurations`。

AWS CLI

若要取得 App Runner 自動擴展組態的分頁清單

下列`list-auto-scaling-configurations`範例列出您 AWS 帳戶中的所有 App Runner 自動擴展組態。每個回應中最多列出五個自動擴展組態。 `LatestOnly` `AutoScalingConfigurationName`和 不會指定。其預設值會導致列出所有作用中組態的最新修訂版。

在此範例中，回應包含兩個結果，但沒有其他結果，因此不會`NextToken`傳回任何結果。

```
aws apprunner list-auto-scaling-configurations \
    --cli-input-json file://input.json
```

`input.json` 的內容：

```
{
    "MaxResults": 5
}
```

輸出：

```
{
    "AutoScalingConfigurationSummaryList": [
        {
            "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/2/
e76562f50d78042e819fead0f59672e6",
            "AutoScalingConfigurationName": "high-availability",
```

```
            "AutoScalingConfigurationRevision": 2
        },
        {
            "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/low-
cost/1/50d7804e7656fead0f59672e62f2e819",
            "AutoScalingConfigurationName": "low-cost",
            "AutoScalingConfigurationRevision": 1
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAutoScalingConfigurations。

`list-connections`

以下程式碼範例顯示如何使用 `list-connections`。

AWS CLI

範例 1：列出所有連線

下列`list-connections`範例列出 AWS 帳戶中的所有 App Runner 連線。

```
aws apprunner list-connections
```

輸出：

```
{
    "ConnectionSummaryList": [
        {
            "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-connection",
            "ConnectionName": "my-github-connection",
            "Status": "AVAILABLE",
            "CreatedAt": "2020-11-03T00:32:51Z",
            "ProviderType": "GITHUB"
        },
        {
            "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-org-connection",
            "ConnectionName": "my-github-org-connection",
```

```
            "Status": "AVAILABLE",
            "CreatedAt": "2020-11-03T02:54:17Z",
            "ProviderType": "GITHUB"
        }
    ]
}
```

範例 2：依名稱列出連線

下列list-connections範例會依名稱列出連線。

```
aws apprunner list-connections \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "ConnectionName": "my-github-org-connection"
}
```

輸出：

```
{
    "ConnectionSummaryList": [
        {
            "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-org-connection",
            "ConnectionName": "my-github-org-connection",
            "Status": "AVAILABLE",
            "CreatedAt": "2020-11-03T02:54:17Z",
            "ProviderType": "GITHUB"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListConnections。

list-operations

以下程式碼範例顯示如何使用 list-operations。

AWS CLI

列出服務上發生的操作

下列list-operations範例列出到目前為止在 App Runner 服務上發生的所有操作。在此範例中，服務是新的，且只CREATE_SERVICE發生單一 類型的操作。

```
aws apprunner list-operations \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

輸出：

```
{
    "OperationSummaryList": [
        {
            "EndedAt": 1606156217,
            "Id": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
            "StartedAt": 1606156014,
            "Status": "SUCCEEDED",
            "TargetArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
            "Type": "CREATE_SERVICE",
            "UpdatedAt": 1606156217
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListOperations。

list-services

以下程式碼範例顯示如何使用 list-services。

AWS CLI

取得 App Runner 服務的分頁清單

下列list-services範例列出 AWS 帳戶中的所有 App Runner 服務。每個回應中最多列出兩個服務。此範例顯示第一個請求。回應包含兩個結果和權杖，可用於下一個請求。當後續回應不包含字符時，已列出所有 服務。

```
aws apprunner list-services \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "MaxResults": 2
}
```

輸出：

```
{
    "NextToken":
 "eyJDdXN0b21lckFjY291bnRJZCI6IjI3MDIwNTQwMjg0NSIsIlNlcnZpY2VTdGF0dXNDb2RlIjoiUFJPVklTU9OU
    "ServiceSummaryList": [
        {
            "CreatedAt": "2020-11-20T19:05:25Z",
            "UpdatedAt": "2020-11-23T12:41:37Z",
            "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
            "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
            "ServiceName": "python-app",
            "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
            "Status": "RUNNING"
        },
        {
            "CreatedAt": "2020-11-06T23:15:30Z",
            "UpdatedAt": "2020-11-23T13:21:22Z",
            "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/golang-
container-app/ab8f94cfe29a460fb8760afd2ee87555",
            "ServiceId": "ab8f94cfe29a460fb8760afd2ee87555",
            "ServiceName": "golang-container-app",
            "ServiceUrl": "e2m8rrrx33.us-east-1.awsapprunner.com",
            "Status": "RUNNING"
```

```
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListServices。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出與 App Runner 服務相關聯的標籤

下列`list-tags-for-resource`範例列出與 App Runner 服務相關聯的所有標籤。

```
aws apprunner list-tags-for-resource \
    --cli-input-json file://input.json
```

`input.json` 的內容：

```
{
    "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

輸出：

```
{
    "Tags": [
        {
            "Key": "Department",
            "Value": "Retail"
        },
        {
            "Key": "CustomerId",
            "Value": "56439872357912"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

pause-service

以下程式碼範例顯示如何使用 pause-service。

AWS CLI

暫停服務

下列pause-service範例會暫停 App Runner 服務。

```
aws apprunner pause-service \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

輸出：

```
{
    "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
    "Service": {
        "CreatedAt": "2020-11-20T19:05:25Z",
        "UpdatedAt": "2020-11-23T12:41:37Z",
        "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceName": "python-app",
        "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
        "SourceConfiguration": {
            "AuthenticationConfiguration": {
                "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
            },
            "AutoDeploymentsEnabled": true,
```

```
            "CodeRepository": {
                "CodeConfiguration": {
                    "CodeConfigurationValues": {
                        "BuildCommand": "pip install -r requirements.txt",
                        "Port": "8080",
                        "Runtime": "PYTHON_3",
                        "RuntimeEnvironmentVariables": [
                            {
                                "NAME": "Jane"
                            }
                        ],
                        "StartCommand": "python server.py"
                    },
                    "ConfigurationSource": "Api"
                },
                "RepositoryUrl": "https://github.com/my-account/python-hello",
                "SourceCodeVersion": {
                    "Type": "BRANCH",
                    "Value": "main"
                }
            }
        },
        "Status": "OPERATION_IN_PROGRESS",
        "InstanceConfiguration": {
            "CPU": "1 vCPU",
            "Memory": "3 GB"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PauseService。

resume-service

以下程式碼範例顯示如何使用 resume-service。

AWS CLI

恢復服務

下列resume-service範例會繼續 App Runner 服務。

```
aws apprunner resume-service \
```

```
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

輸出：

```
{
    "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
    "Service": {
        "CreatedAt": "2020-11-20T19:05:25Z",
        "UpdatedAt": "2020-11-23T12:41:37Z",
        "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceName": "python-app",
        "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
        "SourceConfiguration": {
            "AuthenticationConfiguration": {
                "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
            },
            "AutoDeploymentsEnabled": true,
            "CodeRepository": {
                "CodeConfiguration": {
                    "CodeConfigurationValues": {
                        "BuildCommand": "pip install -r requirements.txt",
                        "Port": "8080",
                        "Runtime": "PYTHON_3",
                        "RuntimeEnvironmentVariables": [
                            {
                                "NAME": "Jane"
                            }
                        ],
                        "StartCommand": "python server.py"
                    },
                    "ConfigurationSource": "Api"
                },
```

```
                "RepositoryUrl": "https://github.com/my-account/python-hello",
                "SourceCodeVersion": {
                    "Type": "BRANCH",
                    "Value": "main"
                }
            }
        },
        "Status": "OPERATION_IN_PROGRESS",
        "InstanceConfiguration": {
            "CPU": "1 vCPU",
            "Memory": "3 GB"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResumeService。

`start-deployment`

以下程式碼範例顯示如何使用 `start-deployment`。

AWS CLI

啟動手動部署

下列`start-deployment`範例會執行 App Runner 服務的手動部署。

```
aws apprunner start-deployment \
    --cli-input-json file://input.json
```

`input.json` 的內容：

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

輸出：

```
{
    "OperationId": "853a7d5b-fc9f-4730-831b-fd8037ab832a"
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartDeployment。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤新增至 App Runner 服務

下列tag-resource範例會將兩個標籤新增至 App Runner 服務。

```
aws apprunner tag-resource \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "Tags": [
        {
            "Key": "Department",
            "Value": "Retail"
        },
        {
            "Key": "CustomerId",
            "Value": "56439872357912"
        }
    ]
}
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從 App Runner 服務移除標籤

下列untag-resource範例會從 App Runner 服務移除兩個標籤。

```
aws apprunner untag-resource \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "TagKeys": [
        "Department",
        "CustomerId"
    ]
}
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-service

以下程式碼範例顯示如何使用 update-service。

AWS CLI

更新記憶體大小

下列update-service範例會將 App Runner 服務的執行個體（擴展單位）記憶體大小更新為 2048 MiB。

當呼叫成功時，App Runner 會啟動非同步更新程序。呼叫傳回的Service結構會反映此呼叫套用的新記憶體值。

```
aws apprunner update-service \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "InstanceConfiguration": {
        "Memory": "4 GB"
    }
}
```

輸出：

```
{
    "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
    "Service": {
        "CreatedAt": "2020-11-20T19:05:25Z",
        "UpdatedAt": "2020-11-23T12:41:37Z",
        "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
        "ServiceName": "python-app",
        "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
        "SourceConfiguration": {
            "AuthenticationConfiguration": {
                "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
            },
            "AutoDeploymentsEnabled": true,
            "CodeRepository": {
                "CodeConfiguration": {
                    "CodeConfigurationValues": {
                        "BuildCommand": "pip install -r requirements.txt",
                        "Port": "8080",
                        "Runtime": "PYTHON_3",
                        "RuntimeEnvironmentVariables": [
                            {
                                "NAME": "Jane"
                            }
                        ],
                        "StartCommand": "python server.py"
                    },
                    "ConfigurationSource": "Api"
                },
                "RepositoryUrl": "https://github.com/my-account/python-hello",
```

```
                "SourceCodeVersion": {
                    "Type": "BRANCH",
                    "Value": "main"
                }
            }
        },
        "Status": "OPERATION_IN_PROGRESS",
        "InstanceConfiguration": {
            "CPU": "1 vCPU",
            "Memory": "4 GB"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateService。

AWS AppConfig 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS AppConfig。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-application`

以下程式碼範例顯示如何使用 `create-application`。

AWS CLI

建立應用程式

下列`create-application`範例會在 AWS AppConfig 中建立應用程式。

```
aws appconfig create-application \
    --name "example-application" \
    --description "An application used for creating an example."
```

輸出：

```
{
    "Description": "An application used for creating an example.",
    "Id": "339ohji",
    "Name": "example-application"
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 1：建立 anAppConfig 應用程式。AWS AppConfig

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateApplication。

create-configuration-profile

以下程式碼範例顯示如何使用 create-configuration-profile。

AWS CLI

建立組態設定檔

下列create-configuration-profile範例會使用儲存在 Systems Manager 功能的 Parameter Store 中的組態來建立組態描述檔。

```
aws appconfig create-configuration-profile \
    --application-id "339ohji" \
    --name "Example-Configuration-Profile" \
    --location-uri "ssm-parameter://Example-Parameter" \
    --retrieval-role-arn "arn:aws:iam::111122223333:role/Example-App-Config-Role"
```

輸出：

```
{
    "ApplicationId": "339ohji",
    "Description": null,
    "Id": "ur8hx2f",
    "LocationUri": "ssm-parameter://Example-Parameter",
    "Name": "Example-Configuration-Profile",
```

```
    "RetrievalRoleArn": "arn:aws:iam::111122223333:role/Example-App-Config-Role",
    "Type": null,
    "Validators": null
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 3：建立組態和組態設定檔。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateConfigurationProfile。

create-environment

以下程式碼範例顯示如何使用 create-environment。

AWS CLI

建立環境

下列create-environment範例會使用您使用 create-application 建立的應用程式來建立名為 Example-Environment 的 AWS AppConfig 環境。

```
aws appconfig create-environment \
    --application-id "339ohji" \
    --name "Example-Environment"
```

輸出：

```
{
    "ApplicationId": "339ohji",
    "Description": null,
    "Id": "54j1r29",
    "Monitors": null,
    "Name": "Example-Environment",
    "State": "ReadyForDeployment"
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 2：建立環境。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateEnvironment。

create-extension-association

以下程式碼範例顯示如何使用 create-extension-association。

AWS CLI

建立延伸模組關聯

下列create-extension-association範例會在 AWS AppConfig 中建立新的延伸模組關聯。

```
aws appconfig create-extension-association \
    --region us-west-2 \
    --extension-identifier S3-backup-extension \
    --resource-identifier "arn:aws:appconfig:us-west-2:123456789012:application/
Finance" \
    --parameters S3bucket=FinanceConfigurationBackup
```

輸出：

```
{
  "Id": "a1b2c3d4",
  "ExtensionArn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-backup-
extension/1",
  "ResourceArn": "arn:aws:appconfig:us-west-2:123456789012:application/Finance",
  "Parameters": {
    "S3bucket": "FinanceConfigurationBackup"
  },
  "ExtensionVersionNumber": 1
}
```

如需詳細資訊，請參閱《AWS AppConfig 使用者指南》中的使用 AppConfig 擴充功能。 AWS AppConfig

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateExtensionAssociation。

create-extension

以下程式碼範例顯示如何使用 create-extension。

AWS CLI

建立 延伸模組

下列create-extension範例會建立新的擴充功能 in AWS AppConfig。

```
aws appconfig create-extension \
    --region us-west-2 \
```

```
    --name S3-backup-extension \
    --
actions PRE_CREATE_HOSTED_CONFIGURATION_VERSION=[{Name=S3backup,Uri=arn:aws:lambda:us-
west-2:123456789012:function:s3backupfunction,RoleArn=arn:aws:iam::123456789012:role/
appconfigextensionrole}] \
    --parameters S3bucket={Required=true}
```

輸出：

```
{
    "Id": "1A2B3C4D",
    "Name": "S3-backup-extension",
    "VersionNumber": 1,
    "Arn": "arn:aws:appconfig:us-west-2:123456789012:extension/1A2B3C4D/1",
    "Actions": {
        "PRE_CREATE_HOSTED_CONFIGURATION_VERSION": [
            {
                "Name": "S3backup",
                "Uri": "arn:aws:lambda:us-
west-2:123456789012:function:s3backupfunction",
                "RoleArn": "arn:aws:iam::123456789012:role/appconfigextensionrole"
            }
        ]
    },
    "Parameters": {
        "S3bucket": {
            "Required": true
        }
    }
}
```

如需詳細資訊，請參閱《AWS AppConfig 使用者指南》中的使用 AppConfig 擴充功能。 AWS AppConfig

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateExtension。

create-hosted-configuration-version

以下程式碼範例顯示如何使用 create-hosted-configuration-version。

AWS CLI

建立託管組態版本

下列create-hosted-configuration-version範例會在 AWS AppConfig 託管組態存放區中建立新的組態。必須先將組態內容轉換為 base64。

```
aws appconfig create-hosted-configuration-version \
    --application-id "339ohji" \
    --configuration-profile-id "ur8hx2f" \
    --
content eyAiTmFtZSI6ICJFeGFtcGxlQXBwbGljYXRpb24iLCAiSWQiOiBFeGFtcGxlSUQsICJSYW5rIjogNyB9
 \
    --content-type "application/json" \
    configuration_version_output_file
```

configuration_version_output_file 的內容：

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

輸出：

```
{
    "ApplicationId": "339ohji",
    "ConfigurationProfileId": "ur8hx2f",
    "VersionNumber": "1",
    "ContentType": "application/json"
}
```

如需詳細資訊，請參閱《AWS AppConfig 使用者指南》中的關於 AppConfig 託管組態存放區。AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateHostedConfigurationVersion。

delete-application

以下程式碼範例顯示如何使用 delete-application。

AWS CLI

如欲刪除應用程式

下列delete-application範例會刪除指定的應用程式。

```
aws appconfig delete-application \
```

```
--application-id 339ohji
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 1：建立 anAppConfig 應用程式。
AWS AppConfig

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteApplication。

delete-configuration-profile

以下程式碼範例顯示如何使用 delete-configuration-profile。

AWS CLI

刪除組態描述檔

下列delete-configuration-profile範例會刪除指定的組態設定檔。

```
aws appconfig delete-configuration-profile \
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 3：建立組態和組態設定檔。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteConfigurationProfile。

delete-deployment-strategy

以下程式碼範例顯示如何使用 delete-deployment-strategy。

AWS CLI

刪除部署策略

下列delete-deployment-strategy範例會刪除指定的部署策略。

```
aws appconfig delete-deployment-strategy \
    --deployment-strategy-id 1225qzk
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 4：建立部署策略。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDeploymentStrategy。

delete-environment

以下程式碼範例顯示如何使用 delete-environment。

AWS CLI

刪除環境

下列delete-environment範例會刪除指定的應用程式環境。

```
aws appconfig delete-environment \
    --application-id 339ohji \
    --environment-id 54j1r29
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 2：建立環境。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteEnvironment。

delete-extension-association

以下程式碼範例顯示如何使用 delete-extension-association。

AWS CLI

刪除延伸模組關聯

下列delete-extension-association範例會從 AWS AppConfig 刪除延伸模組關聯。

```
aws appconfig delete-extension-association \
    --region us-west-2 \
    --extension-association-id a1b2c3d4
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS AppConfig 使用者指南》中的使用 AppConfig 擴充功能。 AWS AppConfig

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteExtensionAssociation。

delete-extension

以下程式碼範例顯示如何使用 delete-extension。

AWS CLI

刪除 延伸模組

下列delete-extension範例會從 AWS AppConfig 刪除延伸模組。

```
aws appconfig delete-extension \
    --region us-west-2 \
    --extension-identifier S3-backup-extension
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS AppConfig 使用者指南》中的使用 AppConfig 擴充功能。 AWS AppConfig

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteExtension。

delete-hosted-configuration-version

以下程式碼範例顯示如何使用 delete-hosted-configuration-version。

AWS CLI

刪除託管組態版本

下列delete-hosted-configuration-version範例會刪除在 AWS AppConfig 託管組態存放區中託管的組態版本。

```
aws appconfig delete-hosted-configuration-version \
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f \
    --version-number 1
```

Output：：此命令不會產生輸出。

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 3：建立組態和組態設定檔。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteHostedConfigurationVersion。

get-application

以下程式碼範例顯示如何使用 get-application。

AWS CLI

列出應用程式的詳細資訊

下列get-application範例列出指定應用程式的詳細資訊。

```
aws appconfig get-application \
    --application-id 339ohji
```

輸出：

```
{
    "Description": "An application used for creating an example.",
    "Id": "339ohji",
    "Name": "example-application"
}
```

如需詳細資訊，請參閱AWS AppConfig 使用者指南》中的 HowAppConfig 運作。 AWS AppConfig

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetApplication。

get-configuration-profile

以下程式碼範例顯示如何使用 get-configuration-profile。

AWS CLI

擷取組態設定檔詳細資訊

下列get-configuration-profile範例會傳回指定組態設定檔的詳細資訊。

```
aws appconfig get-configuration-profile \
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f
```

輸出：

```
{
```

```
    "ApplicationId": "339ohji",
    "Id": "ur8hx2f",
    "Name": "Example-Configuration-Profile",
    "LocationUri": "ssm-parameter://Example-Parameter",
    "RetrievalRoleArn": "arn:aws:iam::111122223333:role/Example-App-Config-Role"
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 3：建立組態和組態設定檔。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetConfigurationProfile。

get-configuration

以下程式碼範例顯示如何使用 get-configuration。

AWS CLI

擷取組態詳細資訊

下列get-configuration範例會傳回範例應用程式的組態詳細資訊。在後續呼叫 get-configuration 時，如果版本已變更，請使用 client-configuration-version 參數來更新應用程式的組態。只有在版本變更時才會更新組態，以避免因為呼叫 get-configuration 而產生超額費用。

```
aws appconfig get-configuration \
    --application "example-application" \
    --environment "Example-Environment" \
    --configuration "Example-Configuration-Profile" \
    --client-id "test-id" \
    configuration-output-file
```

configuration-output-file 的內容：

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

輸出：

```
{
    "ConfigurationVersion": "1",
    "ContentType": "application/json"
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 6：接收組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetConfiguration。

get-deployment-strategy

以下程式碼範例顯示如何使用 get-deployment-strategy。

AWS CLI

擷取部署策略的詳細資訊

下列get-deployment-strategy範例列出指定部署策略的詳細資訊。

```
aws appconfig get-deployment-strategy \
    --deployment-strategy-id 1225qzk
```

輸出：

```
{
    "Id": "1225qzk",
    "Name": "Example-Deployment",
    "DeploymentDurationInMinutes": 15,
    "GrowthType": "LINEAR",
    "GrowthFactor": 25.0,
    "FinalBakeTimeInMinutes": 0,
    "ReplicateTo": "SSM_DOCUMENT"
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 4：建立部署策略。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDeploymentStrategy。

get-deployment

以下程式碼範例顯示如何使用 get-deployment。

AWS CLI

擷取部署詳細資訊

下列get-deployment範例列出在指定環境和部署中部署至應用程式的詳細資訊。

```
aws appconfig get-deployment \
    --application-id 339ohji \
    --environment-id 54j1r29 \
    --deployment-number 1
```

輸出：

```
{
    "ApplicationId": "339ohji",
    "EnvironmentId": "54j1r29",
    "DeploymentStrategyId": "1225qzk",
    "ConfigurationProfileId": "ur8hx2f",
    "DeploymentNumber": 1,
    "ConfigurationName": "Example-Configuration-Profile",
    "ConfigurationLocationUri": "ssm-parameter://Example-Parameter",
    "ConfigurationVersion": "1",
    "DeploymentDurationInMinutes": 15,
    "GrowthType": "LINEAR",
    "GrowthFactor": 25.0,
    "FinalBakeTimeInMinutes": 0,
    "State": "COMPLETE",
    "EventLog": [
        {
            "EventType": "DEPLOYMENT_COMPLETED",
            "TriggeredBy": "APPCONFIG",
            "Description": "Deployment completed",
            "OccurredAt": "2021-09-17T21:59:03.888000+00:00"
        },
        {
            "EventType": "BAKE_TIME_STARTED",
            "TriggeredBy": "APPCONFIG",
            "Description": "Deployment bake time started",
            "OccurredAt": "2021-09-17T21:58:57.722000+00:00"
        },
        {
            "EventType": "PERCENTAGE_UPDATED",
            "TriggeredBy": "APPCONFIG",
            "Description": "Configuration available to 100.00% of clients",
            "OccurredAt": "2021-09-17T21:55:56.816000+00:00"
        },
        {
            "EventType": "PERCENTAGE_UPDATED",
            "TriggeredBy": "APPCONFIG",
```

```
            "Description": "Configuration available to 75.00% of clients",
            "OccurredAt": "2021-09-17T21:52:56.567000+00:00"
        },
        {
            "EventType": "PERCENTAGE_UPDATED",
            "TriggeredBy": "APPCONFIG",
            "Description": "Configuration available to 50.00% of clients",
            "OccurredAt": "2021-09-17T21:49:55.737000+00:00"
        },
        {
            "EventType": "PERCENTAGE_UPDATED",
            "TriggeredBy": "APPCONFIG",
            "Description": "Configuration available to 25.00% of clients",
            "OccurredAt": "2021-09-17T21:46:55.187000+00:00"
        },
        {
            "EventType": "DEPLOYMENT_STARTED",
            "TriggeredBy": "USER",
            "Description": "Deployment started",
            "OccurredAt": "2021-09-17T21:43:54.205000+00:00"
        }
    ],
    "PercentageComplete": 100.0,
    "StartedAt": "2021-09-17T21:43:54.205000+00:00",
    "CompletedAt": "2021-09-17T21:59:03.888000+00:00"
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 5：部署組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDeployment。

get-environment

以下程式碼範例顯示如何使用 get-environment。

AWS CLI

擷取環境詳細資訊

下列get-environment範例會傳回指定環境的詳細資訊和狀態。

```
aws appconfig get-environment \
    --application-id 339ohji \
```

```
    --environment-id 54j1r29
```

輸出：

```
{
    "ApplicationId": "339ohji",
    "Id": "54j1r29",
    "Name": "Example-Environment",
    "State": "ReadyForDeployment"
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 2：建立環境。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetEnvironment。

get-extension-association

以下程式碼範例顯示如何使用 get-extension-association。

AWS CLI

若要取得延伸模組關聯詳細資訊

下列get-extension-association範例顯示延伸關聯的相關資訊。

```
aws appconfig get-extension-association \
    --region us-west-2 \
    --extension-association-id a1b2c3d4
```

輸出：

```
{
    "Id": "a1b2c3d4",
    "ExtensionArn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-backup-
extension/1",
    "ResourceArn": "arn:aws:appconfig:us-west-2:123456789012:application/Finance",
    "Parameters": {
        "S3bucket": "FinanceConfigurationBackup"
    },
    "ExtensionVersionNumber": 1
}
```

如需詳細資訊，請參閱《AWS AppConfig 使用者指南》中的使用 AppConfig 擴充功能。 AWS AppConfig

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetExtensionAssociation。

get-extension

以下程式碼範例顯示如何使用 get-extension。

AWS CLI

若要取得延伸模組詳細資訊

下列get-extension範例顯示 延伸模組的相關資訊。

```
aws appconfig get-extension \
    --region us-west-2 \
    --extension-identifier S3-backup-extension
```

輸出：

```
{
    "Id": "1A2B3C4D",
    "Name": "S3-backup-extension",
    "VersionNumber": 1,
    "Arn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-backup-
extension/1",
    "Actions": {
        "PRE_CREATE_HOSTED_CONFIGURATION_VERSION": [
            {
                "Name": "S3backup",
                "Uri": "arn:aws:lambda:us-
west-2:123456789012:function:S3backupfunction",
                "RoleArn": "arn:aws:iam::123456789012:role/appconfigextensionrole"
            }
        ]
    },
    "Parameters": {
        "S3bucket": {
            "Required": true
        }
    }
```

```
}
```

如需詳細資訊，請參閱《AWS AppConfig 使用者指南》中的使用 AppConfig 擴充功能。 AWS AppConfig

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetExtension。

get-hosted-configuration-version

以下程式碼範例顯示如何使用 get-hosted-configuration-version。

AWS CLI

擷取託管組態詳細資訊

下列get-hosted-configuration-version範例會擷取 AWS AppConfig 託管組態的組態詳細資訊。

```
aws appconfig get-hosted-configuration-version \
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f \
    --version-number 1 \
    hosted-configuration-version-output
```

hosted-configuration-version-output 的內容：

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

輸出：

```
{
    "ApplicationId": "339ohji",
    "ConfigurationProfileId": "ur8hx2f",
    "VersionNumber": "1",
    "ContentType": "application/json"
}
```

如需詳細資訊，請參閱《 AWS AppConfig 使用者指南》中的關於 AppConfig 託管組態存放區。 AWS AppConfig

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetHostedConfigurationVersion。

list-applications

以下程式碼範例顯示如何使用 list-applications。

AWS CLI

列出可用的應用程式

下列list-applications範例列出您 AWS 帳戶中可用的應用程式。

```
aws appconfig list-applications
```

輸出：

```
{
    "Items": [
        {
            "Id": "339ohji",
            "Name": "test-application",
            "Description": "An application used for creating an example."
        },
        {
            "Id": "rwalwu7",
            "Name": "Test-Application"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS AppConfig 使用者指南》中的步驟 1：建立 AWS AppConfig 應用程式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListApplications。

list-configuration-profiles

以下程式碼範例顯示如何使用 list-configuration-profiles。

AWS CLI

列出可用的組態設定檔

下列list-configuration-profiles範例列出指定應用程式的可用組態設定檔。

```
aws appconfig list-configuration-profiles \
    --application-id 339ohji
```

輸出：

```
{
    "Items": [
        {
            "ApplicationId": "339ohji",
            "Id": "ur8hx2f",
            "Name": "Example-Configuration-Profile",
            "LocationUri": "ssm-parameter://Example-Parameter"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 3：建立組態和組態設定檔。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListConfigurationProfiles。

`list-deployment-strategies`

以下程式碼範例顯示如何使用 `list-deployment-strategies`。

AWS CLI

列出可用的部署策略

下列`list-deployment-strategies`範例列出您 AWS 帳戶中可用的部署策略。

```
aws appconfig list-deployment-strategies
```

輸出：

```
{
    "Items": [
        {
            "Id": "1225qzk",
            "Name": "Example-Deployment",
            "DeploymentDurationInMinutes": 15,
            "GrowthType": "LINEAR",
            "GrowthFactor": 25.0,
```

```
            "FinalBakeTimeInMinutes": 0,
            "ReplicateTo": "SSM_DOCUMENT"
        },
        {
            "Id": "AppConfig.AllAtOnce",
            "Name": "AppConfig.AllAtOnce",
            "Description": "Quick",
            "DeploymentDurationInMinutes": 0,
            "GrowthType": "LINEAR",
            "GrowthFactor": 100.0,
            "FinalBakeTimeInMinutes": 10,
            "ReplicateTo": "NONE"
        },
        {
            "Id": "AppConfig.Linear50PercentEvery30Seconds",
            "Name": "AppConfig.Linear50PercentEvery30Seconds",
            "Description": "Test/Demo",
            "DeploymentDurationInMinutes": 1,
            "GrowthType": "LINEAR",
            "GrowthFactor": 50.0,
            "FinalBakeTimeInMinutes": 1,
            "ReplicateTo": "NONE"
        },
        {
            "Id": "AppConfig.Canary10Percent20Minutes",
            "Name": "AppConfig.Canary10Percent20Minutes",
            "Description": "AWS Recommended",
            "DeploymentDurationInMinutes": 20,
            "GrowthType": "EXPONENTIAL",
            "GrowthFactor": 10.0,
            "FinalBakeTimeInMinutes": 10,
            "ReplicateTo": "NONE"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 4：建立部署策略。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDeploymentStrategies。

`list-deployments`

以下程式碼範例顯示如何使用 `list-deployments`。

AWS CLI

列出可用的部署

下列list-deployments範例列出您 AWS 帳戶中指定應用程式和環境的可用部署。

```
aws appconfig list-deployments \
    --application-id 339ohji \
    --environment-id 54j1r29
```

輸出：

```
{
    "Items": [
        {
            "DeploymentNumber": 1,
            "ConfigurationName": "Example-Configuration-Profile",
            "ConfigurationVersion": "1",
            "DeploymentDurationInMinutes": 15,
            "GgrowthType": "LINEAR",
            "GrowthFactor": 25.0,
            "FinalBakeTimeInMinutes": 0,
            "State": "COMPLETE",
            "PercentageComplete": 100.0,
            "StartedAt": "2021-09-17T21:43:54.205000+00:00",
            "CompletedAt": "2021-09-17T21:59:03.888000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 5：部署組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDeployments。

list-environments

以下程式碼範例顯示如何使用 list-environments。

AWS CLI

列出可用的環境

下列list-environments範例列出您 AWS 帳戶中指定應用程式的可用環境。

```
aws appconfig list-environments \
    --application-id 339ohji
```

輸出：

```
{
    "Items": [
        {
            "ApplicationId": "339ohji",
            "Id": "54j1r29",
            "Name": "Example-Environment",
            "State": "ReadyForDeployment"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 2：建立環境。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEnvironments。

`list-extension-associations`

以下程式碼範例顯示如何使用 `list-extension-associations`。

AWS CLI

列出您 AWS 帳戶中 AWS 區域的所有 AWS AppConfig 延伸關聯

下列`list-extension-associations`範例列出特定 AWS 區域中目前 AWS 帳戶的所有 AWS AppConfig 延伸關聯。

```
aws appconfig list-extension-associations \
    --region us-west-2
```

輸出：

```
{
    "Items": [
        {
            "Id": "a1b2c3d4",
            "ExtensionArn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-
backup-extension/1",
```

```
            "ResourceArn": "arn:aws:appconfig:us-west-2:123456789012:application/
Finance"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS AppConfig 使用者指南》中的使用 AppConfig 擴充功能。 AWS AppConfig

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListExtensionAssociations。

`list-extensions`

以下程式碼範例顯示如何使用 `list-extensions`。

AWS CLI

列出您帳戶中 AWS AWS 區域的所有 AWS AppConfig 延伸

下列`list-extensions`範例列出特定 AWS 區域中目前 AWS 帳戶的所有 AWS AppConfig 延伸。命令會傳回自訂和 AWS 撰寫的延伸項目。

```
aws appconfig list-extensions \
    --region us-west-2
```

輸出：

```
{
    "Items": [
        {
            "Id": "1A2B3C4D",
            "Name": "S3-backup-extension",
            "VersionNumber": 1,
            "Arn": "arn:aws:appconfig:us-west-2:123456789012:extension/1A2B3C4D/1"
        },
        {
            "Id": "AWS.AppConfig.FeatureFlags",
            "Name": "AppConfig Feature Flags Helper",
            "VersionNumber": 1,
            "Arn": "arn:aws:appconfig:us-west-2::extension/
AWS.AppConfig.FeatureFlags/1",
            "Description": "Validates AppConfig feature flag data automatically
 against a JSON schema that includes structure and constraints. Also transforms
```

```
 feature flag data prior to sending to the client. This extension is automatically
 associated to configuration profiles with type \"AWS.AppConfig.FeatureFlags\"."
        },
        {
            "Id": "AWS.AppConfig.JiraIntegration",
            "Name": "AppConfig integration with Atlassian Jira",
            "VersionNumber": 1,
            "Arn": "arn:aws:appconfig:us-west-2::extension/
AWS.AppConfig.JiraIntegration/1",
            "Description": "Exports feature flag data from AWS AppConfig into
 Jira. The lifecycle of each feature flag in AppConfig is tracked in Jira as an
 individual issue. Customers can see in Jira when flags are updated, turned on or
 off. Works in conjunction with the AppConfig app in the Atlassian Marketplace and
 is automatically associated to configuration profiles configured within that app."
        },
        {
            "Id": "AWS.AppConfig.DeploymentNotificationsToEventBridge",
            "Name": "AppConfig deployment events to Amazon EventBridge",
            "VersionNumber": 1,
            "Arn": "arn:aws:appconfig:us-west-2::extension/
AWS.AppConfig.DeploymentNotificationsToEventBridge/1",
            "Description": "Sends events to Amazon EventBridge when a deployment
 of configuration data in AppConfig is started, completed, or rolled back. Can
 be associated to the following resources in AppConfig: Application, Environment,
 Configuration Profile."
        },
        {
            "Id": "AWS.AppConfig.DeploymentNotificationsToSqs",
            "Name": "AppConfig deployment events to Amazon SQS",
            "VersionNumber": 1,
            "Arn": "arn:aws:appconfig:us-west-2::extension/
AWS.AppConfig.DeploymentNotificationsToSqs/1",
            "Description": "Sends messages to the configured Amazon SQS queue when
 a deployment of configuration data in AppConfig is started, completed, or rolled
 back. Can be associated to the following resources in AppConfig: Application,
 Environment, Configuration Profile."
        },
        {
            "Id": "AWS.AppConfig.DeploymentNotificationsToSns",
            "Name": "AppConfig deployment events to Amazon SNS",
            "VersionNumber": 1,
            "Description": "Sends events to the configured Amazon SNS topic when
 a deployment of configuration data in AppConfig is started, completed, or rolled
```

```
 back. Can be associated to the following resources in AppConfig: Application,
 Environment, Configuration Profile."
        }
    ]
}
```

如需詳細資訊，請參閱《AWS AppConfig 使用者指南》中的使用 AppConfig 擴充功能。 AWS AppConfig

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListExtensions。

`list-hosted-configuration-versions`

以下程式碼範例顯示如何使用 `list-hosted-configuration-versions`。

AWS CLI

列出可用的託管組態版本

下列`list-hosted-configuration-versions`範例列出指定應用程式和組態設定檔的 AWS AppConfig 託管組態存放區中託管的組態版本。

```
aws appconfig list-hosted-configuration-versions \
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f
```

輸出：

```
{
    "Items": [
        {
            "ApplicationId": "339ohji",
            "ConfigurationProfileId": "ur8hx2f",
            "VersionNumber": 1,
            "ContentType": "application/json"
        }
    ]
}
```

如需詳細資訊，請參閱《 AWS AppConfig 使用者指南》中的關於 AppConfig 託管組態存放區。 AWS AppConfig

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListHostedConfigurationVersions。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出應用程式的標籤

下列`list-tags-for-resource`範例列出指定應用程式的標籤。

```
aws appconfig list-tags-for-resource \
    --resource-arn arn:aws:appconfig:us-east-1:682428703967:application/339ohji
```

輸出：

```
{
    "Tags": {
        "group1": "1"
    }
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 1：建立 anAppConfig 應用程式。
AWS AppConfig

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`start-deployment`

以下程式碼範例顯示如何使用 `start-deployment`。

AWS CLI

啟動組態部署

下列`start-deployment`範例使用指定的環境、部署策略和組態描述檔，開始部署到應用程式。

```
aws appconfig start-deployment \
    --application-id 339ohji \
    --environment-id 54j1r29 \
```

```
    --deployment-strategy-id 1225qzk \
    --configuration-profile-id ur8hx2f \
    --configuration-version 1
```

輸出：

```
{
    "ApplicationId": "339ohji",
    "EnvironmentId": "54j1r29",
    "DeploymentStrategyId": "1225qzk",
    "ConfigurationProfileId": "ur8hx2f",
    "DeploymentNumber": 1,
    "ConfigurationName": "Example-Configuration-Profile",
    "ConfigurationLocationUri": "ssm-parameter://Example-Parameter",
    "ConfigurationVersion": "1",
    "DeploymentDurationInMinutes": 15,
    "GrowthType": "LINEAR",
    "GrowthFactor": 25.0,
    "FinalBakeTimeInMinutes": 0,
    "State": "DEPLOYING",
    "EventLog": [
        {
            "EventType": "DEPLOYMENT_STARTED",
            "TriggeredBy": "USER",
            "Description": "Deployment started",
            "OccurredAt": "2021-09-17T21:43:54.205000+00:00"
        }
    ],
    "PercentageComplete": 0.0,
    "StartedAt": "2021-09-17T21:43:54.205000+00:00"
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 5：部署組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartDeployment。

stop-deployment

以下程式碼範例顯示如何使用 stop-deployment。

AWS CLI

停止組態部署

下列stop-deployment範例會停止將應用程式組態部署到指定的環境。

```
aws appconfig stop-deployment \
    --application-id 339ohji \
    --environment-id 54j1r29 \
    --deployment-number 2
```

輸出：

```
{
    "DeploymentNumber": 0,
    "DeploymentDurationInMinutes": 0,
    "GrowthFactor": 0.0,
    "FinalBakeTimeInMinutes": 0,
    "PercentageComplete": 0.0
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 5：部署組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopDeployment。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

標記應用程式

下列tag-resource範例會標記應用程式資源。

```
aws appconfig tag-resource \
    --resource-arn arn:aws:appconfig:us-east-1:682428703967:application/339ohji \
    --tags '{"group1" : "1"}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 1：建立 anAppConfig 應用程式。
AWS AppConfig

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從應用程式移除標籤

下列untag-resource範例會從指定的應用程式移除 group1 標籤。

```
aws appconfig untag-resource \
    --resource-arn arn:aws:appconfig:us-east-1:111122223333:application/339ohji \
    --tag-keys '["group1"]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 1：建立 anAppConfig 應用程式。

AWS AppConfig

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-application

以下程式碼範例顯示如何使用 update-application。

AWS CLI

更新應用程式

下列update-application範例會更新指定應用程式的名稱。

```
aws appconfig update-application \
    --application-id 339ohji \
    --name "Example-Application"
```

輸出：

```
{
    "Id": "339ohji",
    "Name": "Example-Application",
    "Description": "An application used for creating an example."
```

```
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 1：建立 anAppConfig 應用程式。AWS AppConfig

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateApplication。

update-configuration-profile

以下程式碼範例顯示如何使用 update-configuration-profile。

AWS CLI

更新組態設定檔

下列update-configuration-profile範例會更新指定組態描述檔的描述。

```
aws appconfig update-configuration-profile \
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f \
    --description "Configuration profile used for examples."
```

輸出：

```
{
    "ApplicationId": "339ohji",
    "Id": "ur8hx2f",
    "Name": "Example-Configuration-Profile",
    "Description": "Configuration profile used for examples.",
    "LocationUri": "ssm-parameter://Example-Parameter",
    "RetrievalRoleArn": "arn:aws:iam::111122223333:role/Example-App-Config-Role"
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 3：建立組態和組態設定檔。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateConfigurationProfile。

update-deployment-strategy

以下程式碼範例顯示如何使用 update-deployment-strategy。

AWS CLI

更新部署策略

下列update-deployment-strategy範例會將指定部署策略中的最終製作時間更新為 20 分鐘。

```
aws appconfig update-deployment-strategy \
    --deployment-strategy-id 1225qzk \
    --final-bake-time-in-minutes 20
```

輸出：

```
{
    "Id": "1225qzk",
    "Name": "Example-Deployment",
    "DeploymentDurationInMinutes": 15,
    "GrowthType": "LINEAR",
    "GrowthFactor": 25.0,
    "FinalBakeTimeInMinutes": 20,
    "ReplicateTo": "SSM_DOCUMENT"
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 4：建立部署策略。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDeploymentStrategy。

update-environment

以下程式碼範例顯示如何使用 update-environment。

AWS CLI

更新環境

下列update-environment範例會更新環境的描述。

```
aws appconfig update-environment \
    --application-id 339ohji \
    --environment-id 54j1r29 \
    --description "An environment for examples."
```

輸出：

```
{
    "ApplicationId": "339ohji",
    "Id": "54j1r29",
    "Name": "Example-Environment",
    "Description": "An environment for examples.",
    "State": "RolledBack"
}
```

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 2：建立環境。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateEnvironment。

update-extension-association

以下程式碼範例顯示如何使用 update-extension-association。

AWS CLI

更新 an AWS AppConfig 延伸模組關聯

下列update-extension-association範例會將新的參數值新增至 in AWS AppConfig 中的延伸關聯。

```
aws appconfig update-extension-association \
    --region us-west-2 \
    --extension-association-id a1b2c3d4 \
    --parameters S3bucket=FinanceMobileApp
```

輸出：

```
{
    "Id": "a1b2c3d4",
    "ExtensionArn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-backup-
extension/1",
    "ResourceArn": "arn:aws:appconfig:us-west-2:123456789012:application/Finance",
    "Parameters": {
        "S3bucket": "FinanceMobileApp"
    },
    "ExtensionVersionNumber": 1
```

```
}
```

如需詳細資訊，請參閱《AWS AppConfig 使用者指南》中的使用 AppConfig 擴充功能。 AWS AppConfig

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateExtensionAssociation。

update-extension

以下程式碼範例顯示如何使用 update-extension。

AWS CLI

更新 an AWS AppConfig 延伸模組

下列update-extension範例會將額外的參數金鑰新增至 in AWS AppConfig 延伸模組。

```
aws appconfig update-extension \
    --region us-west-2 \
    --extension-identifier S3-backup-extension \
    --parameters S3bucket={Required=true},CampaignID={Required=false}
```

輸出：

```
{
    "Id": "1A2B3C4D",
    "Name": "S3-backup-extension",
    "VersionNumber": 1,
    "Arn": "arn:aws:appconfig:us-west-2:123456789012:extension/1A2B3C4D/1",
    "Actions": {
        "PRE_CREATE_HOSTED_CONFIGURATION_VERSION": [
            {
                "Name": "S3backup",
                "Uri": "arn:aws:lambda:us-
west-2:123456789012:function:S3backupfunction",
                "RoleArn": "arn:aws:iam::123456789012:role/appconfigextensionrole"
            }
        ]
    },
    "Parameters": {
        "CampaignID": {
            "Required": false
```

```
        },
        "S3bucket": {
            "Required": true
        }
    }
}
```

如需詳細資訊，請參閱《AWS AppConfig 使用者指南》中的使用 AppConfig 擴充功能。 AWS AppConfig

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateExtension。

`validate-configuration`

以下程式碼範例顯示如何使用 `validate-configuration`。

AWS CLI

驗證組態

下列`validate-configuration`範例使用組態設定檔中的驗證程式來驗證組態。

```
aws appconfig validate-configuration \
    --application-id abc1234 \
    --configuration-profile-id ur8hx2f \
    --configuration-version 1
```

命令不會產生輸出。

如需詳細資訊，請參閱 AWS AppConfig 使用者指南中的步驟 3：建立組態和組態設定檔。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ValidateConfiguration。

使用 的 Application Auto Scaling 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Application Auto Scaling 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

delete-scaling-policy

以下程式碼範例顯示如何使用 delete-scaling-policy。

AWS CLI

刪除擴展政策

此範例會刪除在預設叢集中執行的 Amazon ECS 服務 Web 應用程式擴展政策。

命令：

```
aws application-autoscaling delete-scaling-policy --policy-name web-app-cpu-lt-25 --
scalable-dimension ecs:service:DesiredCount --resource-id service/default/web-app --
service-namespace ecs
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteScalingPolicy。

delete-scheduled-action

以下程式碼範例顯示如何使用 delete-scheduled-action。

AWS CLI

刪除排程動作

模糊delete-scheduled-action範例會從指定的 Amazon AppStream 2.0 機群中刪除指定的排程動作：

```
aws application-autoscaling delete-scheduled-action \
    --service-namespace appstream \
    --scalable-dimension appstream:fleet:DesiredCapacity \
    --resource-id fleet/sample-fleet \
    --scheduled-action-name my-recurring-action
```

此命令不會產生輸出。

如需詳細資訊，請參閱《應用程式自動擴展使用者指南》中的排程擴展。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteScheduledAction。

deregister-scalable-target

以下程式碼範例顯示如何使用 deregister-scalable-target。

AWS CLI

取消註冊可擴展的目標

此範例會取消註冊在預設叢集中執行的 Amazon ECS 服務名為 Web-app 的可擴展目標。

命令：

```
aws application-autoscaling deregister-scalable-target --service-namespace ecs --
scalable-dimension ecs:service:DesiredCount --resource-id service/default/web-app
```

此範例會取消註冊自訂資源的可擴展目標。custom-resource-id.txt 檔案包含可識別資源 ID 的字串，對於自訂資源，這是透過 Amazon API Gateway 端點存取自訂資源的路徑。

命令：

```
aws application-autoscaling deregister-scalable-target --service-namespace custom-
resource --scalable-dimension custom-resource:ResourceType:Property --resource-
id file://~/custom-resource-id.txt
```

custom-resource-id.txt 檔案的內容：

```
https://example.execute-api.us-west-2.amazonaws.com/prod/
scalableTargetDimensions/1-23456789
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterScalableTarget。

describe-scalable-targets

以下程式碼範例顯示如何使用 describe-scalable-targets。

AWS CLI

描述可擴展的目標

下列describe-scalable-targets範例說明ecs服務命名空間的可擴展目標。

```
aws application-autoscaling describe-scalable-targets \
    --service-namespace ecs
```

輸出：

```
{
    "ScalableTargets": [
        {
            "ServiceNamespace": "ecs",
            "ScalableDimension": "ecs:service:DesiredCount",
            "ResourceId": "service/default/web-app",
            "MinCapacity": 1,
            "MaxCapacity": 10,
            "RoleARN": "arn:aws:iam::123456789012:role/
aws-service-role/ecs.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ECSService",
            "CreationTime": 1462558906.199,
            "SuspendedState": {
                "DynamicScalingOutSuspended": false,
                "ScheduledScalingSuspended": false,
                "DynamicScalingInSuspended": false
            },
            "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Application Auto Scaling 使用者指南》中的可與 Application Auto Scaling 搭配使用的 服務。 Auto Scaling

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeScalableTargets。

describe-scaling-activities

以下程式碼範例顯示如何使用 describe-scaling-activities。

AWS CLI

範例 1：描述指定 Amazon ECS 服務的擴展活動

下列`describe-scaling-activities`範例說明叢集中執行`web-app`之名為 的 Amazon ECS 服務擴展活動`default`。輸出會顯示由擴展政策啟動的擴展活動。

```
aws application-autoscaling describe-scaling-activities \
    --service-namespace ecs \
    --resource-id service/default/web-app
```

輸出：

```
{
    "ScalingActivities": [
        {
            "ScalableDimension": "ecs:service:DesiredCount",
            "Description": "Setting desired count to 1.",
            "ResourceId": "service/default/web-app",
            "ActivityId": "e6c5f7d1-dbbb-4a3f-89b2-51f33e766399",
            "StartTime": 1462575838.171,
            "ServiceNamespace": "ecs",
            "EndTime": 1462575872.111,
            "Cause": "monitor alarm web-app-cpu-lt-25 in state ALARM triggered
 policy web-app-cpu-lt-25",
            "StatusMessage": "Successfully set desired count to 1. Change
 successfully fulfilled by ecs.",
            "StatusCode": "Successful"
        }
    ]
}
```

如需詳細資訊，請參閱《Application Auto Scaling 使用者指南》中的 Application Auto Scaling 的擴展活動。

範例 2：描述指定 DynamoDB 資料表的擴展活動

下列`describe-scaling-activities`範例說明名為 之 DynamoDB 資料表的擴展活動`TestTable`。輸出會顯示由兩個不同的排程動作啟動的擴展活動。

```
aws application-autoscaling describe-scaling-activities \
    --service-namespace dynamodb \
```

```
    --resource-id table/TestTable
```

輸出：

```
{
    "ScalingActivities": [
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting write capacity units to 10.",
            "ResourceId": "table/my-table",
            "ActivityId": "4d1308c0-bbcf-4514-a673-b0220ae38547",
            "StartTime": 1561574415.086,
            "ServiceNamespace": "dynamodb",
            "EndTime": 1561574449.51,
            "Cause": "maximum capacity was set to 10",
            "StatusMessage": "Successfully set write capacity units to 10. Change
 successfully fulfilled by dynamodb.",
            "StatusCode": "Successful"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting min capacity to 5 and max capacity to 10",
            "ResourceId": "table/my-table",
            "ActivityId": "f2b7847b-721d-4e01-8ef0-0c8d3bacc1c7",
            "StartTime": 1561574414.644,
            "ServiceNamespace": "dynamodb",
            "Cause": "scheduled action name my-second-scheduled-action was
 triggered",
            "StatusMessage": "Successfully set min capacity to 5 and max capacity to
10",
            "StatusCode": "Successful"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting write capacity units to 15.",
            "ResourceId": "table/my-table",
            "ActivityId": "d8ea4de6-9eaa-499f-b466-2cc5e681ba8b",
            "StartTime": 1561574108.904,
            "ServiceNamespace": "dynamodb",
            "EndTime": 1561574140.255,
            "Cause": "minimum capacity was set to 15",
            "StatusMessage": "Successfully set write capacity units to 15. Change
 successfully fulfilled by dynamodb.",
```

```
            "StatusCode": "Successful"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Description": "Setting min capacity to 15 and max capacity to 20",
            "ResourceId": "table/my-table",
            "ActivityId": "3250fd06-6940-4e8e-bb1f-d494db7554d2",
            "StartTime": 1561574108.512,
            "ServiceNamespace": "dynamodb",
            "Cause": "scheduled action name my-first-scheduled-action was
 triggered",
            "StatusMessage": "Successfully set min capacity to 15 and max capacity
 to 20",
            "StatusCode": "Successful"
        }
    ]
}
```

如需詳細資訊，請參閱《Application Auto Scaling 使用者指南》中的 Application Auto Scaling 的擴展活動。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeScalingActivities。

describe-scaling-policies

以下程式碼範例顯示如何使用 describe-scaling-policies。

AWS CLI

描述擴展政策

此範例命令說明 ecs 服務命名空間的擴展政策。

命令：

```
aws application-autoscaling describe-scaling-policies --service-namespace ecs
```

輸出：

```
{
    "ScalingPolicies": [
        {
            "PolicyName": "web-app-cpu-gt-75",
```

```
            "ScalableDimension": "ecs:service:DesiredCount",
            "ResourceId": "service/default/web-app",
            "CreationTime": 1462561899.23,
            "StepScalingPolicyConfiguration": {
                "Cooldown": 60,
                "StepAdjustments": [
                    {
                        "ScalingAdjustment": 200,
                        "MetricIntervalLowerBound": 0.0
                    }
                ],
                "AdjustmentType": "PercentChangeInCapacity"
            },
            "PolicyARN": "arn:aws:autoscaling:us-
west-2:012345678910:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/ecs/
service/default/web-app:policyName/web-app-cpu-gt-75",
            "PolicyType": "StepScaling",
            "Alarms": [
                {
                    "AlarmName": "web-app-cpu-gt-75",
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:web-app-cpu-gt-75"
                }
            ],
            "ServiceNamespace": "ecs"
        },
        {
            "PolicyName": "web-app-cpu-lt-25",
            "ScalableDimension": "ecs:service:DesiredCount",
            "ResourceId": "service/default/web-app",
            "CreationTime": 1462562575.099,
            "StepScalingPolicyConfiguration": {
                "Cooldown": 1,
                "StepAdjustments": [
                    {
                        "ScalingAdjustment": -50,
                        "MetricIntervalUpperBound": 0.0
                    }
                ],
                "AdjustmentType": "PercentChangeInCapacity"
            },
            "PolicyARN": "arn:aws:autoscaling:us-
west-2:012345678910:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/ecs/
service/default/web-app:policyName/web-app-cpu-lt-25",
```

```
            "PolicyType": "StepScaling",
            "Alarms": [
                {
                    "AlarmName": "web-app-cpu-lt-25",
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:web-app-cpu-lt-25"
                }
            ],
            "ServiceNamespace": "ecs"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeScalingPolicies。

describe-scheduled-actions

以下程式碼範例顯示如何使用 describe-scheduled-actions。

AWS CLI

描述排程動作

下列describe-scheduled-actions範例顯示指定服務命名空間之排程動作的詳細資訊：

```
aws application-autoscaling describe-scheduled-actions \
    --service-namespace dynamodb
```

輸出：

```
{
    "ScheduledActions": [
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Schedule": "at(2019-05-20T18:35:00)",
            "ResourceId": "table/my-table",
            "CreationTime": 1561571888.361,
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:2d36aa3b-cdf9-4565-b290-81db519b227d:resource/
dynamodb/table/my-table:scheduledActionName/my-first-scheduled-action",
            "ScalableTargetAction": {
                "MinCapacity": 15,
```

```
                "MaxCapacity": 20
            },
            "ScheduledActionName": "my-first-scheduled-action",
            "ServiceNamespace": "dynamodb"
        },
        {
            "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
            "Schedule": "at(2019-05-20T18:40:00)",
            "ResourceId": "table/my-table",
            "CreationTime": 1561571946.021,
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:2d36aa3b-cdf9-4565-b290-81db519b227d:resource/
dynamodb/table/my-table:scheduledActionName/my-second-scheduled-action",
            "ScalableTargetAction": {
                "MinCapacity": 5,
                "MaxCapacity": 10
            },
            "ScheduledActionName": "my-second-scheduled-action",
            "ServiceNamespace": "dynamodb"
        }
    ]
}
```

如需詳細資訊，請參閱《應用程式自動擴展使用者指南》中的排程擴展。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeScheduledActions。

list-tags-for-resource

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出可擴展目標的標籤

下列`list-tags-for-resource`範例列出連接到其 ARN 所指定可擴展目標的標籤金鑰名稱和值。

```
aws application-autoscaling list-tags-for-resource \
    --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-
target/1234abcd56ab78cd901ef1234567890ab123
```

輸出：

```
{
    "Tags": {
        "environment": "production"
    }
}
```

如需詳細資訊，請參閱《Application Auto Scaling 使用者指南》中的 Application Auto Scaling 的標記支援。 Auto Scaling

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

put-scaling-policy

以下程式碼範例顯示如何使用 put-scaling-policy。

AWS CLI

範例 1：套用具備預先定義指標規格的目標追蹤擴展政策

下列put-scaling-policy範例會將具有預先定義指標規格的目標追蹤擴展政策套用至預設叢集中稱為 web-app 的 Amazon ECS 服務。此政策會將服務的平均 CPU 使用率保持在 75%，向外擴展和向內擴展冷卻時間為 60 秒。輸出包含代表您建立的兩個 CloudWatch 警示的 ARNs 和名稱。

```
aws application-autoscaling put-scaling-policy --service-namespace ecs \
--scalable-dimension ecs:service:DesiredCount \
--resource-id service/default/web-app \
--policy-name cpu75-target-tracking-scaling-policy --policy-
type TargetTrackingScaling \
--target-tracking-scaling-policy-configuration file://config.json
```

此範例假設您在目前目錄中有 config.json 檔案，其中包含下列內容：

```
{
     "TargetValue": 75.0,
     "PredefinedMetricSpecification": {
         "PredefinedMetricType": "ECSServiceAverageCPUUtilization"
     },
     "ScaleOutCooldown": 60,
    "ScaleInCooldown": 60
}
```

輸出：

```
{
    "PolicyARN": "arn:aws:autoscaling:us-west-2:012345678910:scalingPolicy:6d8972f3-
efc8-437c-92d1-6270f29a66e7:resource/ecs/service/default/web-app:policyName/cpu75-
target-tracking-scaling-policy",
    "Alarms": [
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-AlarmHigh-d4f0770c-
b46e-434a-a60f-3b36d653feca",
            "AlarmName": "TargetTracking-service/default/web-app-AlarmHigh-d4f0770c-
b46e-434a-a60f-3b36d653feca"
        },
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-AlarmLow-1b437334-
d19b-4a63-a812-6c67aaf2910d",
            "AlarmName": "TargetTracking-service/default/web-app-AlarmLow-1b437334-
d19b-4a63-a812-6c67aaf2910d"
        }
    ]
}
```

範例 2：套用具備自訂指標規格的目標追蹤擴展政策

下列put-scaling-policy範例會將具有自訂指標規格的目標追蹤擴展政策套用至預設叢集中稱為 web-app 的 Amazon ECS 服務。此政策會將服務的平均使用率保持在 75%，向外擴展和向內擴展冷卻時間為 60 秒。輸出包含代表您建立的兩個 CloudWatch 警示的 ARNs 和名稱。

```
aws application-autoscaling put-scaling-policy --service-namespace ecs \
--scalable-dimension ecs:service:DesiredCount \
--resource-id service/default/web-app \
--policy-name cms75-target-tracking-scaling-policy
--policy-type TargetTrackingScaling \
--target-tracking-scaling-policy-configuration file://config.json
```

此範例假設您在目前目錄中有 config.json 檔案，其中包含下列內容：

```
{
    "TargetValue":75.0,
    "CustomizedMetricSpecification":{
        "MetricName":"MyUtilizationMetric",
        "Namespace":"MyNamespace",
```

```
        "Dimensions": [
            {
                "Name":"MyOptionalMetricDimensionName",
                "Value":"MyOptionalMetricDimensionValue"
            }
        ],
        "Statistic":"Average",
        "Unit":"Percent"
    },
    "ScaleOutCooldown": 60,
    "ScaleInCooldown": 60
}
```

輸出：

```
{
    "PolicyARN": "arn:aws:autoscaling:us-west-2:012345678910:scalingPolicy:
 8784a896-b2ba-47a1-b08c-27301cc499a1:resource/ecs/service/default/web-
app:policyName/cms75-target-tracking-scaling-policy",
    "Alarms": [
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-
AlarmHigh-9bc77b56-0571-4276-ba0f-d4178882e0a0",
            "AlarmName": "TargetTracking-service/default/web-app-
AlarmHigh-9bc77b56-0571-4276-ba0f-d4178882e0a0"
        },
        {
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-
AlarmLow-9b6ad934-6d37-438e-9e05-02836ddcbdc4",
            "AlarmName": "TargetTracking-service/default/web-app-
AlarmLow-9b6ad934-6d37-438e-9e05-02836ddcbdc4"
        }
    ]
}
```

範例 3：只針對擴增套用目標追蹤擴展政策

下列put-scaling-policy範例會將目標追蹤擴展政策套用至預設叢集web-app中呼叫的 Amazon ECS 服務。當 Application Load Balancer RequestCountPerTarget 的指標超過閾值時，此政策會用來向外擴展 ECS 服務。輸出包含代表您建立的 CloudWatch 警示的 ARN 和名稱。

```
aws application-autoscaling put-scaling-policy \
    --service-namespace ecs \
    --scalable-dimension ecs:service:DesiredCount \
    --resource-id service/default/web-app \
    --policy-name alb-scale-out-target-tracking-scaling-policy \
    --policy-type TargetTrackingScaling \
    --target-tracking-scaling-policy-configuration file://config.json
```

config.json 的內容：

```
{
     "TargetValue": 1000.0,
     "PredefinedMetricSpecification": {
         "PredefinedMetricType": "ALBRequestCountPerTarget",
         "ResourceLabel": "app/EC2Co-EcsEl-1TKLTMITMM0EO/f37c06a68c1748aa/
targetgroup/EC2Co-Defau-LDNM7Q3ZH1ZN/6d4ea56ca2d6a18d"
     },
     "ScaleOutCooldown": 60,
    "ScaleInCooldown": 60,
    "DisableScaleIn": true
}
```

輸出：

```
{
    "PolicyARN": "arn:aws:autoscaling:us-west-2:123456789012:scalingPolicy:6d8972f3-
efc8-437c-92d1-6270f29a66e7:resource/ecs/service/default/web-app:policyName/alb-
scale-out-target-tracking-scaling-policy",
    "Alarms": [
        {
            "AlarmName": "TargetTracking-service/default/web-app-AlarmHigh-d4f0770c-
b46e-434a-a60f-3b36d653feca",
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-service/default/web-app-AlarmHigh-d4f0770c-
b46e-434a-a60f-3b36d653feca"
        }
    ]
}
```

如需詳細資訊，請參閱《Application Auto Scaling 使用者指南》中的 Application Auto Scaling 的目標追蹤擴展政策。 AWS Auto Scaling

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutScalingPolicy。

put-scheduled-action

以下程式碼範例顯示如何使用 put-scheduled-action。

AWS CLI

將排程動作新增至 DynamoDB 資料表

此範例會將排程動作新增至名為 TestTable 的 DynamoDB 資料表，以根據週期性排程進行擴展。根據指定的排程（每天 12：15pm UTC)，如果目前的容量低於為 MinCapacity 指定的值，Application Auto Scaling 會向外擴展至 MinCapacity 指定的值。

命令：

```
aws application-autoscaling put-scheduled-action --service-namespace dynamodb
 --scheduled-action-name my-recurring-action --schedule "cron(15 12 * * ? *)" --
resource-id table/TestTable --scalable-dimension dynamodb:table:WriteCapacityUnits
 --scalable-target-action MinCapacity=6
```

如需詳細資訊，請參閱《Application Auto Scaling 使用者指南》中的排程擴展。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutScheduledAction。

register-scalable-target

以下程式碼範例顯示如何使用 register-scalable-target。

AWS CLI

範例 1：將 ECS 服務註冊為可擴展的目標

下列register-scalable-target範例向 Application Auto Scaling 註冊 Amazon ECS 服務。它也會將具有金鑰名稱environment和值的標籤新增至production可擴展的目標。

```
aws application-autoscaling register-scalable-target \
    --service-namespace ecs \
    --scalable-dimension ecs:service:DesiredCount \
    --resource-id service/default/web-app \
    --min-capacity 1 --max-capacity 10 \
    --tags environment=production
```

輸出：

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

如需其他 AWS 服務和自訂資源的範例，請參閱《Application AWS Auto Scaling 使用者指南》中的可與 Application Auto Scaling 搭配使用之服務中的主題。 Auto Scaling

範例 2：暫停可擴展目標的擴展活動

下列register-scalable-target範例會暫停現有可擴展目標的擴展活動。

```
aws application-autoscaling register-scalable-target \
    --service-namespace dynamodb \
    --scalable-dimension dynamodb:table:ReadCapacityUnits \
    --resource-id table/my-table \
    --suspended-
state DynamicScalingInSuspended=true,DynamicScalingOutSuspended=true,ScheduledScalingSuspend
```

輸出：

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

如需詳細資訊，請參閱《Application Auto Scaling 使用者指南》中的暫停和繼續 Application Auto Scaling 的擴展。

範例 3：恢復可擴展目標的擴展活動

下列register-scalable-target範例會繼續現有可擴展目標的擴展活動。

```
aws application-autoscaling register-scalable-target \
    --service-namespace dynamodb \
    --scalable-dimension dynamodb:table:ReadCapacityUnits \
    --resource-id table/my-table \
    --suspended-
state DynamicScalingInSuspended=false,DynamicScalingOutSuspended=false,ScheduledScalingSuspe
```

輸出：

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

如需詳細資訊，請參閱《Application Auto Scaling 使用者指南》中的暫停和繼續 Application Auto Scaling 的擴展。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterScalableTarget。

`tag-resource`

以下程式碼範例顯示如何使用 `tag-resource`。

AWS CLI

將標籤新增至可擴展的目標

下列`tag-resource`範例會將具有金鑰名稱`environment`和值的標籤新增至其 ARN 指定的`production`可擴展目標。

```
aws application-autoscaling tag-resource \
    --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-
target/1234abcd56ab78cd901ef1234567890ab123 \
    --tags environment=production
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Application Auto Scaling 使用者指南》中的 Application Auto Scaling 的標記支援。 Auto Scaling

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

`untag-resource`

以下程式碼範例顯示如何使用 `untag-resource`。

AWS CLI

從可擴展的目標中移除標籤

下列untag-resource範例會從其 ARN 指定的environment可擴展目標中移除具有金鑰名稱的標籤對。

```
aws application-autoscaling untag-resource \
    --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123 \
    --tag-keys "environment"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Application Auto Scaling 使用者指南》中的 Application Auto Scaling 的標記支援。 Auto Scaling

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

使用 的應用程式探索服務範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Application Discovery Service 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`describe-agents`

以下程式碼範例顯示如何使用 `describe-agents`。

AWS CLI

描述具有指定 collectionStatus 狀態的客服人員

此範例命令說明集合狀態為「STARTED」或「STOPPED」的集合代理程式。

命令：

```
aws discovery describe-agents --filters
 name="collectionStatus",values="STARTED","STOPPED",condition="EQUALS" --max-
results 3
```

輸出：

```
{
       "Snapshots": [
      {
            "version": "1.0.40.0",
            "agentType": "EC2",
            "hostName": "ip-172-31-40-234",
            "collectionStatus": "STOPPED",
            "agentNetworkInfoList": [
                {
                    "macAddress": "06:b5:97:14:fc:0d",
                    "ipAddress": "172.31.40.234"
                }
            ],
            "health": "UNKNOWN",
            "agentId": "i-003305c02a776e883",
            "registeredTime": "2016-12-09T19:05:06Z",
            "lastHealthPingTime": "2016-12-09T19:05:10Z"
        },
        {
            "version": "1.0.40.0",
            "agentType": "EC2",
            "hostName": "ip-172-31-39-64",
            "collectionStatus": "STARTED",
            "agentNetworkInfoList": [
                {
                    "macAddress": "06:a1:0e:c7:b2:73",
                    "ipAddress": "172.31.39.64"
                }
            ],
            "health": "SHUTDOWN",
            "agentId": "i-003a5e5e2b36cf8bd",
            "registeredTime": "2016-11-16T16:36:25Z",
            "lastHealthPingTime": "2016-11-16T16:47:37Z"
        }
    ]
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAgents。

describe-configurations

以下程式碼範例顯示如何使用 describe-configurations。

AWS CLI

描述選取的資產組態

此範例命令說明兩個指定伺服器的組態。動作會從組態 ID 偵測資產類型。每個命令只允許一種資產類型。

命令：

```
aws discovery describe-configurations --configuration-ids "d-server-099385097ef9fbcfb" "d-server-0c4f2dd1fee22c6c1"
```

輸出：

```
{
    "configurations": [
        {
                "server.performance.maxCpuUsagePct": "0.0",
                "server.performance.maxDiskReadIOPS": "0.0",
                "server.performance.avgCpuUsagePct": "0.0",
                "server.type": "EC2",
                "server.performance.maxNetworkReadsPerSecondInKB": "0.19140625",
                "server.hostName": "ip-172-31-35-152",
                "server.configurationId": "d-server-0c4f2dd1fee22c6c1",
                "server.tags.hasMoreValues": "false",
                "server.performance.minFreeRAMInKB": "1543496.0",
                "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
                "server.performance.maxDiskReadsPerSecondInKB": "0.0",
                "server.applications": "[]",
                "server.performance.numDisks": "1",
                "server.performance.numCpus": "1",
                "server.performance.numCores": "1",
                "server.performance.maxDiskWriteIOPS": "0.0",
                "server.performance.maxNetworkWritesPerSecondInKB": "0.82421875",
                "server.performance.avgDiskWritesPerSecondInKB": "0.0",
```

```
                "server.networkInterfaceInfo": "[{\"name\":\"eth0\",
\"macAddress\":\"06:A7:7D:3F:54:57\",\"ipAddress\":\"172.31.35.152\",\"netMask\":
\"255.255.240.0\"},{\"name\":\"lo\",\"macAddress\":\"00:00:00:00:00:00\",\"ipAddress
\":\"127.0.0.1\",\"netMask\":\"255.0.0.0\"},{\"name\":\"eth0\",\"macAddress\":
\"06:A7:7D:3F:54:57\",\"ipAddress\":\"fe80::4a7:7dff:fe3f:5457\"},{\"name\":\"lo\",
\"macAddress\":\"00:00:00:00:00:00\",\"ipAddress\":\"::1\"}]",
                "server.performance.avgNetworkReadsPerSecondInKB":
 "0.04915364583333333",
                "server.tags": "[]",
                "server.applications.hasMoreValues": "false",
                "server.timeOfCreation": "2016-10-28 23:44:00.0",
                "server.agentId": "i-4447bc1b",
                "server.performance.maxDiskWritesPerSecondInKB": "0.0",
                "server.performance.avgDiskReadIOPS": "0.0",
                "server.performance.avgFreeRAMInKB": "1547210.1333333333",
                "server.performance.avgDiskReadsPerSecondInKB": "0.0",
                "server.performance.avgDiskWriteIOPS": "0.0",
                "server.performance.numNetworkCards": "2",
                "server.hypervisor": "xen",
                "server.networkInterfaceInfo.hasMoreValues": "false",
                "server.performance.avgNetworkWritesPerSecondInKB": "0.1380859375",
                "server.osName": "Linux - Amazon Linux AMI release 2015.03",
                "server.performance.totalRAMInKB": "1694732.0",
                "server.cpuType": "x64"
        },
        {
                "server.performance.maxCpuUsagePct": "100.0",
                "server.performance.maxDiskReadIOPS": "0.0",
                "server.performance.avgCpuUsagePct": "14.733333333333338",
                "server.type": "EC2",
                "server.performance.maxNetworkReadsPerSecondInKB": "13.400390625",
                "server.hostName": "ip-172-31-42-208",
                "server.configurationId": "d-server-099385097ef9fbcfb",
                "server.tags.hasMoreValues": "false",
                "server.performance.minFreeRAMInKB": "1531104.0",
                "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
                "server.performance.maxDiskReadsPerSecondInKB": "0.0",
                "server.applications": "[]",
                "server.performance.numDisks": "1",
                "server.performance.numCpus": "1",
                "server.performance.numCores": "1",
                "server.performance.maxDiskWriteIOPS": "1.0",
                "server.performance.maxNetworkWritesPerSecondInKB": "12.271484375",
```

```
                "server.performance.avgDiskWritesPerSecondInKB":
 "0.5333333333333334",
                "server.networkInterfaceInfo": "[{\"name\":\"eth0\",
\"macAddress\":\"06:4A:79:60:75:61\",\"ipAddress\":\"172.31.42.208\",\"netMask
\":\"255.255.240.0\"},{\"name\":\"eth0\",\"macAddress\":\"06:4A:79:60:75:61\",
\"ipAddress\":\"fe80::44a:79ff:fe60:7561\"},{\"name\":\"lo\",\"macAddress\":
\"00:00:00:00:00:00\",\"ipAddress\":\"::1\"},{\"name\":\"lo\",\"macAddress\":
\"00:00:00:00:00:00\",\"ipAddress\":\"127.0.0.1\",\"netMask\":\"255.0.0.0\"}]",
                "server.performance.avgNetworkReadsPerSecondInKB":
 "2.8720052083333334",
                "server.tags": "[]",
                "server.applications.hasMoreValues": "false",
                "server.timeOfCreation": "2016-10-28 23:44:30.0",
                "server.agentId": "i-c142b99e",
                "server.performance.maxDiskWritesPerSecondInKB": "4.0",
                "server.performance.avgDiskReadIOPS": "0.0",
                "server.performance.avgFreeRAMInKB": "1534946.4",
                "server.performance.avgDiskReadsPerSecondInKB": "0.0",
                "server.performance.avgDiskWriteIOPS": "0.13333333333333336",
                "server.performance.numNetworkCards": "2",
                "server.hypervisor": "xen",
                "server.networkInterfaceInfo.hasMoreValues": "false",
                "server.performance.avgNetworkWritesPerSecondInKB":
 "1.7977864583333332",
                "server.osName": "Linux - Amazon Linux AMI release 2015.03",
                "server.performance.totalRAMInKB": "1694732.0",
                "server.cpuType": "x64"
        }
    ]
}
```

描述選取的資產組態

此範例命令說明兩個指定應用程式的組態。動作會從組態 ID 偵測資產類型。每個命令只允許一種資產類型。

命令：

```
aws discovery describe-configurations --configuration-ids "d-
application-0ac39bc0e4fad0e42" "d-application-02444a45288013764q"
```

輸出：

```
{
    "configurations": [
        {
                "application.serverCount": "0",
                "application.name": "Application-12345",
                "application.lastModifiedTime": "2016-12-13 23:53:27.0",
                "application.description": "",
                "application.timeOfCreation": "2016-12-13 23:53:27.0",
                "application.configurationId": "d-application-0ac39bc0e4fad0e42"
        },
        {
                "application.serverCount": "0",
                "application.name": "Application-67890",
                "application.lastModifiedTime": "2016-12-13 23:53:33.0",
                "application.description": "",
                "application.timeOfCreation": "2016-12-13 23:53:33.0",
                "application.configurationId": "d-application-02444a45288013764"
         }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeConfigurations。

`list-configurations`

以下程式碼範例顯示如何使用 `list-configurations`。

AWS CLI

列出符合一組篩選條件的所有已探索伺服器

此範例命令會列出符合兩種主機名稱模式且未執行 Ubuntu 的已探索伺服器。

命令：

```
aws discovery list-configurations --configuration-type SERVER --filters
 name="server.hostName",values="172-31-35","172-31-42",condition="CONTAINS"
 name="server.osName",values="Ubuntu",condition="NOT_CONTAINS"
```

輸出：

```
{
```

```
    "configurations": [
      {
            "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
            "server.type": "EC2",
            "server.hostName": "ip-172-31-42-208",
            "server.timeOfCreation": "2016-10-28 23:44:30.0",
            "server.configurationId": "d-server-099385097ef9fbcfb",
            "server.osName": "Linux - Amazon Linux AMI release 2015.03",
            "server.agentId": "i-c142b99e"
        },
        {
            "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
            "server.type": "EC2",
            "server.hostName": "ip-172-31-35-152",
            "server.timeOfCreation": "2016-10-28 23:44:00.0",
            "server.configurationId": "d-server-0c4f2dd1fee22c6c1",
            "server.osName": "Linux - Amazon Linux AMI release 2015.03",
            "server.agentId": "i-4447bc1b"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListConfigurations。

使用 的 AppRegistry 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 AppRegistry 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-attribute-group`

以下程式碼範例顯示如何使用 `associate-attribute-group`。

AWS CLI

建立屬性群組的關聯

下列`associate-attribute-group`範例會將您 AWS 帳戶中的特定屬性群組與您 AWS 帳戶中的特定應用程式建立關聯。

```
aws servicecatalog-appregistry associate-attribute-group \
    --application "ExampleApplication" \
    --attribute-group "ExampleAttributeGroup"
```

輸出：

```
{
    "applicationArn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt9l",
    "attributeGroupArn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl"
}
```

如需詳細資訊，請參閱 AWS Service Catalog AppRegistry 管理員指南中的關聯和取消關聯屬性群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateAttributeGroup。

`create-application`

以下程式碼範例顯示如何使用 `create-application`。

AWS CLI

建立應用程式

下列`create-application`範例會在您的帳戶中建立新的應用程式 AWS 。

```
aws servicecatalog-appregistry create-application \
    --name "ExampleApplication"
```

輸出：

```
{
    "application": {
        "id": "0ars38r6btoohvpvd9gqrptt9l",
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt9l",
        "name": "ExampleApplication",
        "creationTime": "2023-02-28T21:10:10.820000+00:00",
        "lastUpdateTime": "2023-02-28T21:10:10.820000+00:00",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 AWS Service Catalog AppRegistry 管理員指南中的建立應用程式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateApplication。

create-attribute-group

以下程式碼範例顯示如何使用 create-attribute-group。

AWS CLI

建立屬性群組

下列create-attribute-group範例會在您的帳戶中建立新的屬性群組 AWS 。

```
aws servicecatalog-appregistry create-attribute-group \
    --name "ExampleAttributeGroup" \
    --attributes '{"SomeKey1":"SomeValue1","SomeKey2":"SomeValue2"}'
```

輸出：

```
{
    "attributeGroup": {
        "id": "01sj5xdwhbw54kejwnt09fnpcl",
```

```
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl",
        "name": "ExampleAttributeGroup",
        "creationTime": "2023-02-28T20:38:01.389000+00:00",
        "lastUpdateTime": "2023-02-28T20:38:01.389000+00:00",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 AWS Service Catalog AppRegistry 管理員指南中的建立屬性群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAttributeGroup。

delete-application

以下程式碼範例顯示如何使用 delete-application。

AWS CLI

如欲刪除應用程式

下列delete-application範例會刪除您 AWS 帳戶中的特定應用程式。

```
aws servicecatalog-appregistry delete-application \
    --application "ExampleApplication3"
```

輸出：

```
{
    "application": {
        "id": "055gw7aynr1i5mbv7kjwzx5945",
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/055gw7aynr1i5mbv7kjwzx5945",
        "name": "ExampleApplication3",
        "creationTime": "2023-02-28T22:06:28.228000+00:00",
        "lastUpdateTime": "2023-02-28T22:06:28.228000+00:00"
    }
}
```

如需詳細資訊，請參閱 AWS Service Catalog AppRegistry 管理員指南中的刪除應用程式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteApplication。

delete-attribute-group

以下程式碼範例顯示如何使用 delete-attribute-group。

AWS CLI

範例 8：刪除屬性群組

下列delete-attribute-group範例會刪除您 AWS 帳戶中的特定屬性群組。

```
aws servicecatalog-appregistry delete-attribute-group \
    --attribute-group "ExampleAttributeGroup3"
```

輸出：

```
{
    "attributeGroup": {
        "id": "011ge6y3emyjijt8dw8jn6r0hv",
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/011ge6y3emyjijt8dw8jn6r0hv",
        "name": "ExampleAttributeGroup3",
        "creationTime": "2023-02-28T22:05:35.224000+00:00",
        "lastUpdateTime": "2023-02-28T22:05:35.224000+00:00"
    }
}
```

如需詳細資訊，請參閱 AWS Service Catalog AppRegistry 管理員指南中的刪除屬性群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAttributeGroup。

get-application

以下程式碼範例顯示如何使用 get-application。

AWS CLI

取得應用程式

下列get-application範例會擷取您 AWS 帳戶中特定應用程式的中繼資料資訊。

```
aws servicecatalog-appregistry get-application \
```

```
    --application "ExampleApplication"
```

輸出：

```
{
    "id": "0ars38r6btoohvpvd9gqrptt9l",
    "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt9l",
    "name": "ExampleApplication",
    "creationTime": "2023-02-28T21:10:10.820000+00:00",
    "lastUpdateTime": "2023-02-28T21:10:10.820000+00:00",
    "associatedResourceCount": 0,
    "tags": {
        "aws:servicecatalog:applicationName": "ExampleApplication"
    },
    "integrations": {
        "resourceGroup": {
            "state": "CREATE_COMPLETE",
            "arn": "arn:aws:resource-groups:us-west-2:813737243517:group/
AWS_AppRegistry_Application-ExampleApplication"
        }
    }
}
```

如需詳細資訊，請參閱《AWS Service Catalog AppRegistry 管理員指南》中的使用應用程式詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetApplication。

get-attribute-group

以下程式碼範例顯示如何使用 get-attribute-group。

AWS CLI

取得屬性群組

下列get-attribute-group範例會擷取您 AWS 帳戶中的特定屬性群組。

```
aws servicecatalog-appregistry get-attribute-group \
    --attribute-group "ExampleAttributeGroup"
```

輸出：

```
{
    "id": "01sj5xdwhbw54kejwnt09fnpcl",
    "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl",
    "name": "ExampleAttributeGroup",
    "attributes": "{\"SomeKey1\":\"SomeValue1\",\"SomeKey2\":\"SomeValue2\"}",
    "creationTime": "2023-02-28T20:38:01.389000+00:00",
    "lastUpdateTime": "2023-02-28T20:38:01.389000+00:00",
    "tags": {
        "aws:servicecatalog:attributeGroupName": "ExampleAttributeGroup"
    }
}
```

如需詳細資訊，請參閱 AWS Service Catalog AppRegistry 管理員指南中的管理屬性群組的中繼資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAttributeGroup。

list-applications

以下程式碼範例顯示如何使用 `list-applications`。

AWS CLI

列出應用程式

下列`list-applications`範例會擷取您 AWS 帳戶中所有應用程式的清單。

```
aws servicecatalog-appregistry list-applications
```

輸出：

```
{
    "applications": [
        {
            "id": "03axw94pjfj3uan00tcgbrxnkw",
            "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/03axw94pjfj3uan00tcgbrxnkw",
            "name": "ExampleApplication2",
            "creationTime": "2023-02-28T21:59:34.094000+00:00",
```

```
            "lastUpdateTime": "2023-02-28T21:59:34.094000+00:00"
        },
        {
            "id": "055gw7aynr1i5mbv7kjwzx5945",
            "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/055gw7aynr1i5mbv7kjwzx5945",
            "name": "ExampleApplication3",
            "creationTime": "2023-02-28T22:06:28.228000+00:00",
            "lastUpdateTime": "2023-02-28T22:06:28.228000+00:00"
        },
        {
            "id": "0ars38r6btoohvpvd9gqrptt9l",
            "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt9l",
            "name": "ExampleApplication",
            "description": "This is an example application",
            "creationTime": "2023-02-28T21:10:10.820000+00:00",
            "lastUpdateTime": "2023-02-28T21:24:19.729000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Service Catalog AppRegistry 管理員指南》中的檢視應用程式詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListApplications。

`list-associated-attribute-groups`

以下程式碼範例顯示如何使用 `list-associated-attribute-groups`。

AWS CLI

列出關聯的屬性群組

下列`list-associated-attribute-groups`範例會擷取您 AWS 帳戶中與 AWS 帳戶中特定應用程式相關聯的所有屬性群組清單。

```
aws servicecatalog-appregistry list-associated-attribute-groups \
    --application "ExampleApplication"
```

輸出：

```
{
    "attributeGroups": [
        "01sj5xdwhbw54kejwnt09fnpcl"
    ]
}
```

如需詳細資訊，請參閱《AWS Service Catalog AppRegistry 管理員指南》中的關聯和取消關聯屬性群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAssociatedAttributeGroups。

`list-attribute-groups-for-application`

以下程式碼範例顯示如何使用 `list-attribute-groups-for-application`。

AWS CLI

列出應用程式的屬性群組

下列`list-attribute-groups-for-application`範例列出與您 AWS 帳戶中特定應用程式相關聯的 AWS 帳戶中所有屬性群組的詳細資訊。

```
aws servicecatalog-appregistry list-attribute-groups-for-application \
    --application "ExampleApplication"
```

輸出：

```
{
    "attributeGroupsDetails": [
        {
            "id": "01sj5xdwhbw54kejwnt09fnpcl",
            "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl",
            "name": "ExampleAttributeGroup"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Service Catalog AppRegistry 管理員指南》中的檢視屬性群組詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAttributeGroupsForApplication。

list-attribute-groups

以下程式碼範例顯示如何使用 list-attribute-groups。

AWS CLI

列出屬性群組

下列list-attribute-groups範例會擷取您 AWS 帳戶中所有屬性群組的清單。

```
aws servicecatalog-appregistry list-attribute-groups
```

輸出：

```
{
    "attributeGroups": [
        {
            "id": "011ge6y3emyjijt8dw8jn6r0hv",
            "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/011ge6y3emyjijt8dw8jn6r0hv",
            "name": "ExampleAttributeGroup3",
            "creationTime": "2023-02-28T22:05:35.224000+00:00",
            "lastUpdateTime": "2023-02-28T22:05:35.224000+00:00"
        },
        {
            "id": "01sj5xdwhbw54kejwnt09fnpcl",
            "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl",
            "name": "ExampleAttributeGroup",
            "description": "This is an example attribute group",
            "creationTime": "2023-02-28T20:38:01.389000+00:00",
            "lastUpdateTime": "2023-02-28T21:02:04.559000+00:00"
        },
        {
            "id": "03n1yffgq6d18vwrzxf0c70nm3",
            "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/03n1yffgq6d18vwrzxf0c70nm3",
            "name": "ExampleAttributeGroup2",
            "creationTime": "2023-02-28T21:57:30.687000+00:00",
            "lastUpdateTime": "2023-02-28T21:57:30.687000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Service Catalog AppRegistry 管理員指南》中的檢視屬性群組詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAttributeGroups。

update-application

以下程式碼範例顯示如何使用 update-application。

AWS CLI

更新應用程式

下列update-application範例會更新您 AWS 帳戶中的特定應用程式，以包含描述。

```
aws servicecatalog-appregistry update-application \
    --application "ExampleApplication" \
    --description "This is an example application"
```

輸出：

```
{
    "application": {
        "id": "0ars38r6btoohvpvd9gqrptt9l",
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt9l",
        "name": "ExampleApplication",
        "description": "This is an example application",
        "creationTime": "2023-02-28T21:10:10.820000+00:00",
        "lastUpdateTime": "2023-02-28T21:24:19.729000+00:00",
        "tags": {
            "aws:servicecatalog:applicationName": "ExampleApplication"
        }
    }
}
```

如需詳細資訊，請參閱 AWS Service Catalog AppRegistry 管理員指南中的編輯應用程式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateApplication。

update-attribute-group

以下程式碼範例顯示如何使用 update-attribute-group。

AWS CLI

更新屬性群組

下列update-attribute-group範例會更新您 AWS 帳戶中的特定屬性群組，以包含描述。

```
aws servicecatalog-appregistry update-attribute-group \
    --attribute-group "ExampleAttributeGroup" \
    --description "This is an example attribute group"
```

輸出：

```
{
    "attributeGroup": {
        "id": "01sj5xdwhbw54kejwnt09fnpcl",
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl",
        "name": "ExampleAttributeGroup",
        "description": "This is an example attribute group",
        "creationTime": "2023-02-28T20:38:01.389000+00:00",
        "lastUpdateTime": "2023-02-28T21:02:04.559000+00:00",
        "tags": {
            "aws:servicecatalog:attributeGroupName": "ExampleAttributeGroup"
        }
    }
}
```

如需詳細資訊，請參閱 AWS Service Catalog AppRegistry 管理員指南中的編輯屬性群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAttributeGroup。

使用 的 Athena 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Athena 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

batch-get-named-query

以下程式碼範例顯示如何使用 batch-get-named-query。

AWS CLI

傳回多個查詢的相關資訊

下列batch-get-named-query範例會傳回具有指定 IDs之具名查詢的相關資訊。

```
aws athena batch-get-named-query \
    --named-query-ids a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 a1b2c3d4-5678-90ab-cdef-EXAMPLE33333
```

輸出：

```
{
    "NamedQueries": [
        {
            "Name": "Flights Select Query",
            "Description": "Sample query to get the top 10 airports with the most number of departures since 2000",
            "Database": "sampledb",
            "QueryString": "SELECT origin, count(*) AS total_departures\nFROM \nflights_parquet\nWHERE year >= '2000'\nGROUP BY origin\nORDER BY total_departures DESC\nLIMIT 10;",
            "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "WorkGroup": "primary"
        },
        {
            "Name": "Load flights table partitions",
            "Description": "Sample query to load flights table partitions using MSCK REPAIR TABLE statement",
            "Database": "sampledb",
            "QueryString": "MSCK REPAIR TABLE flights_parquet;",
            "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "WorkGroup": "primary"
        },
        {
            "Name": "CloudFront Select Query",
            "Description": "Sample query to view requests per operating system during a particular time frame",
```

```
            "Database": "sampledb",
            "QueryString": "SELECT os, COUNT(*) count FROM cloudfront_logs WHERE
 date BETWEEN date '2014-07-05' AND date '2014-08-05' GROUP BY os;",
            "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "WorkGroup": "primary"
        }
    ],
    "UnprocessedNamedQueryIds": []
}
```

如需詳細資訊，請參閱 Amazon Athena 使用者指南中的使用 Amazon Athena 執行 SQL 查詢。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetNamedQuery。

batch-get-query-execution

以下程式碼範例顯示如何使用 batch-get-query-execution。

AWS CLI

傳回一或多個查詢執行的相關資訊

下列batch-get-query-execution範例會傳回具有指定查詢 IDs 之查詢的查詢執行資訊。

```
aws athena batch-get-query-execution \
    --query-execution-ids a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222
```

輸出：

```
{
    "QueryExecutions": [
        {
            "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Query": "create database if not exists webdata",
            "StatementType": "DDL",
            "ResultConfiguration": {
                "OutputLocation": "s3://amzn-s3-demo-bucket/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111.txt"
            },
            "QueryExecutionContext": {},
            "Status": {
                "State": "SUCCEEDED",
```

```
                "SubmissionDateTime": 1593470720.592,
                "CompletionDateTime": 1593470720.902
            },
            "Statistics": {
                "EngineExecutionTimeInMillis": 232,
                "DataScannedInBytes": 0,
                "TotalExecutionTimeInMillis": 310,
            "ResultConfiguration": {

                "QueryQueueTimeInMillis": 50,
                "ServiceProcessingTimeInMillis": 28
            },
            "WorkGroup": "AthenaAdmin"
        },
        {
            "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "Query": "select date, location, browser, uri, status from
 cloudfront_logs where method = 'GET' and status = 200 and location like 'SFO%'
 limit 10",
            "StatementType": "DML",
            "ResultConfiguration": {
                "OutputLocation": "s3://amzn-s3-demo-bucket/a1b2c3d4-5678-90ab-cdef-
EXAMPLE22222.csv"
            },
            "QueryExecutionContext": {
                "Database": "mydatabase",
                "Catalog": "awsdatacatalog"
            },
            "Status": {
                "State": "SUCCEEDED",
                "SubmissionDateTime": 1593469842.665,
                "CompletionDateTime": 1593469846.486
            },
            "Statistics": {
                "EngineExecutionTimeInMillis": 3600,
                "DataScannedInBytes": 203089,
                "TotalExecutionTimeInMillis": 3821,
                "QueryQueueTimeInMillis": 267,
                "QueryPlanningTimeInMillis": 1175
            },
            "WorkGroup": "AthenaAdmin"
        }
    ],
    "UnprocessedQueryExecutionIds": []
```

```
}
```

如需詳細資訊，請參閱 Amazon Athena 使用者指南中的使用 Amazon Athena 執行 SQL 查詢。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetQueryExecution。

create-data-catalog

以下程式碼範例顯示如何使用 create-data-catalog。

AWS CLI

建立資料目錄

下列create-data-catalog範例會建立dynamo_db_catalog資料目錄。

```
aws athena create-data-catalog \
    --name dynamo_db_catalog \
    --type LAMBDA \
    --description "DynamoDB Catalog" \
    --parameters function=arn:aws:lambda:us-
west-2:111122223333:function:dynamo_db_lambda
```

此命令不會產生輸出。若要查看結果，請使用 aws athena get-data-catalog --name dynamo_db_catalog。

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的註冊目錄：create-data-catalog。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDataCatalog。

create-named-query

以下程式碼範例顯示如何使用 create-named-query。

AWS CLI

建立具名查詢

下列create-named-query範例會在AthenaAdmin工作群組中建立已儲存的查詢，查詢 flights_parquet 資料表中是否有 2016 年 1 月從西雅圖到紐約的航班，其出發和抵達都延遲超過 10 分鐘。由於資料表中的機場代碼值是包含雙引號（例如 "SEA") 的字串，因此它們會被反斜線逸出並被單引號包圍。

```
aws athena create-named-query \
    --name "SEA to JFK delayed flights Jan 2016" \
    --description "Both arrival and departure delayed more than 10 minutes." \
    --database sampledb \
    --query-string "SELECT flightdate, carrier, flightnum, origin, dest,
 depdelayminutes, arrdelayminutes FROM sampledb.flights_parquet WHERE yr = 2016 AND
 month = 1 AND origin = '\"SEA\"' AND dest = '\"JFK\"' AND depdelayminutes > 10 AND
 arrdelayminutes > 10" \
    --work-group AthenaAdmin
```

輸出：

```
{
    "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

如需詳細資訊，請參閱 Amazon Athena 使用者指南中的使用 Amazon Athena 執行 SQL 查詢。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateNamedQuery。

create-work-group

以下程式碼範例顯示如何使用 create-work-group。

AWS CLI

建立工作群組

下列create-work-group範例會建立名為 的工作群組Data_Analyst_Group，其具有查詢結果輸出位置 s3://amzn-s3-demo-bucket。命令會建立覆寫用戶端組態設定的工作群組，其中包含查詢結果輸出位置。命令也會啟用 CloudWatch 指標，並將三個鍵值標籤對新增至工作群組，以將其與其他工作群組區分開來。請注意，引--configuration數在分隔其選項的逗號前面沒有空格。

```
aws athena create-work-group \
    --name Data_Analyst_Group \
    --configuration ResultConfiguration={OutputLocation="s3://amzn-s3-demo-
bucket"},EnforceWorkGroupConfiguration="true",PublishCloudWatchMetricsEnabled="true"
 \
    --description "Workgroup for data analysts" \
```

```
    --tags Key=Division,Value=West Key=Location,Value=Seattle Key=Team,Value="Big
Data"
```

此命令不會產生輸出。若要查看結果，請使用 `aws athena get-work-group --work-group Data_Analyst_Group`。

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的管理工作群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateWorkGroup。

delete-data-catalog

以下程式碼範例顯示如何使用 `delete-data-catalog`。

AWS CLI

刪除資料目錄

下列`delete-data-catalog`範例會刪除`UnusedDataCatalog`資料目錄。

```
aws athena delete-data-catalog \
    --name UnusedDataCatalog
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的刪除目錄：delete-data-catalog。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDataCatalog。

delete-named-query

以下程式碼範例顯示如何使用 `delete-named-query`。

AWS CLI

刪除具名查詢

下列`delete-named-query`範例會刪除具有指定 ID 的具名查詢。

```
aws athena delete-named-query \
    --named-query-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Amazon Athena 使用者指南中的使用 Amazon Athena 執行 SQL 查詢。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNamedQuery。

delete-work-group

以下程式碼範例顯示如何使用 delete-work-group。

AWS CLI

刪除工作群組

下列delete-work-group範例會刪除TeamB工作群組。

```
aws athena delete-work-group \
    --work-group TeamB
```

此命令不會產生輸出。若要確認刪除，請使用 aws athena list-work-groups。

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的管理工作群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteWorkGroup。

get-data-catalog

以下程式碼範例顯示如何使用 get-data-catalog。

AWS CLI

傳回資料目錄的相關資訊

下列get-data-catalog範例會傳回dynamo_db_catalog資料目錄的相關資訊。

```
aws athena get-data-catalog \
    --name dynamo_db_catalog
```

輸出：

```
{
    "DataCatalog": {
```

```
        "Name": "dynamo_db_catalog",
        "Description": "DynamoDB Catalog",
        "Type": "LAMBDA",
        "Parameters": {
            "catalog": "dynamo_db_catalog",
            "metadata-function": "arn:aws:lambda:us-
west-2:111122223333:function:dynamo_db_lambda",
            "record-function": "arn:aws:lambda:us-
west-2:111122223333:function:dynamo_db_lambda"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的顯示目錄詳細資訊：get-data-catalog。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDataCatalog。

get-database

以下程式碼範例顯示如何使用 get-database。

AWS CLI

傳回資料目錄中資料庫的相關資訊

下列get-database範例會傳回AwsDataCatalog資料目錄中sampledb資料庫的相關資訊。

```
aws athena get-database \
    --catalog-name AwsDataCatalog \
    --database-name sampledb
```

輸出：

```
{
    "Database": {
        "Name": "sampledb",
        "Description": "Sample database",
        "Parameters": {
            "CreatedBy": "Athena",
            "EXTERNAL": "TRUE"
        }
```

```
    }
}
```

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的顯示資料庫詳細資訊：get-database。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDatabase。

get-named-query

以下程式碼範例顯示如何使用 get-named-query。

AWS CLI

傳回具名查詢

下列get-named-query範例會傳回具有指定 ID 之查詢的相關資訊。

```
aws athena get-named-query \
    --named-query-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "NamedQuery": {
        "Name": "CloudFront Logs - SFO",
        "Description": "Shows successful GET request data for SFO",
        "Database": "default",
        "QueryString": "select date, location, browser, uri, status from
 cloudfront_logs where method = 'GET' and status = 200 and location like 'SFO%'
 limit 10",
        "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "WorkGroup": "AthenaAdmin"
    }
}
```

如需詳細資訊，請參閱 Amazon Athena 使用者指南中的使用 Amazon Athena 執行 SQL 查詢。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetNamedQuery。

get-query-execution

以下程式碼範例顯示如何使用 get-query-execution。

AWS CLI

傳回查詢執行的相關資訊

下列get-query-execution範例會傳回具有指定查詢 ID 之查詢的相關資訊。

```
aws athena get-query-execution \
    --query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "QueryExecution": {
        "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Query": "select date, location, browser, uri, status from cloudfront_logs
 where method = 'GET
' and status = 200 and location like 'SFO%' limit 10",
        "StatementType": "DML",
        "ResultConfiguration": {
            "OutputLocation": "s3://amzn-s3-demo-bucket/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111.csv"
        },
        "QueryExecutionContext": {
            "Database": "mydatabase",
            "Catalog": "awsdatacatalog"
        },
        "Status": {
            "State": "SUCCEEDED",
            "SubmissionDateTime": 1593469842.665,
            "CompletionDateTime": 1593469846.486
        },
        "Statistics": {
            "EngineExecutionTimeInMillis": 3600,
            "DataScannedInBytes": 203089,
            "TotalExecutionTimeInMillis": 3821,
            "QueryQueueTimeInMillis": 267,
            "QueryPlanningTimeInMillis": 1175
        },
        "WorkGroup": "AthenaAdmin"
    }
}
```

如需詳細資訊，請參閱 Amazon Athena 使用者指南中的使用 Amazon Athena 執行 SQL 查詢。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetQueryExecution。

get-query-results

以下程式碼範例顯示如何使用 get-query-results。

AWS CLI

傳回查詢的結果

下列get-query-results範例會傳回具有指定查詢 ID 的查詢結果。

```
aws athena get-query-results \
    --query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "ResultSet": {
        "Rows": [
            {
                "Data": [
                    {
                        "VarCharValue": "date"
                    },
                    {
                        "VarCharValue": "location"
                    },
                    {
                        "VarCharValue": "browser"
                    },
                    {
                        "VarCharValue": "uri"
                    },
                    {
                        "VarCharValue": "status"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
```

```
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Safari"
                    },
                    {
                        "VarCharValue": "/test-image-2.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Opera"
                    },
                    {
                        "VarCharValue": "/test-image-2.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Firefox"
```

```
                    },
                    {
                        "VarCharValue": "/test-image-3.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Lynx"
                    },
                    {
                        "VarCharValue": "/test-image-3.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "IE"
                    },
                    {
                        "VarCharValue": "/test-image-2.jpeg"
                    },
                    {
                        "VarCharValue": "200"
```

```
                }
            ]
        },
        {
            "Data": [
                {
                    "VarCharValue": "2014-07-05"
                },
                {
                    "VarCharValue": "SFO4"
                },
                {
                    "VarCharValue": "Opera"
                },
                {
                    "VarCharValue": "/test-image-1.jpeg"
                },
                {
                    "VarCharValue": "200"
                }
            ]
        },
        {
            "Data": [
                {
                    "VarCharValue": "2014-07-05"
                },
                {
                    "VarCharValue": "SFO4"
                },
                {
                    "VarCharValue": "Chrome"
                },
                {
                    "VarCharValue": "/test-image-3.jpeg"
                },
                {
                    "VarCharValue": "200"
                }
            ]
        },
        {
            "Data": [
                {
```

```
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Firefox"
                    },
                    {
                        "VarCharValue": "/test-image-2.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Chrome"
                    },
                    {
                        "VarCharValue": "/test-image-3.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
```

```
                    "VarCharValue": "IE"
                },
                {
                    "VarCharValue": "/test-image-2.jpeg"
                },
                {
                    "VarCharValue": "200"
                }
            ]
        }
    ],
    "ResultSetMetadata": {
        "ColumnInfo": [
            {
                "CatalogName": "hive",
                "SchemaName": "",
                "TableName": "",
                "Name": "date",
                "Label": "date",
                "Type": "date",
                "Precision": 0,
                "Scale": 0,
                "Nullable": "UNKNOWN",
                "CaseSensitive": false
            },
            {
                "CatalogName": "hive",
                "SchemaName": "",
                "TableName": "",
                "Name": "location",
                "Label": "location",
                "Type": "varchar",
                "Precision": 2147483647,
            "Data": [

                "Scale": 0,
                "Nullable": "UNKNOWN",
                "CaseSensitive": true
            },
            {
                "CatalogName": "hive",
                "SchemaName": "",
                "TableName": "",
                "Name": "browser",
```

```
                    "Label": "browser",
                    "Type": "varchar",
                    "Precision": 2147483647,
                    "Scale": 0,
                    "Nullable": "UNKNOWN",
                    "CaseSensitive": true
                },
                {
                    "CatalogName": "hive",
                    "SchemaName": "",
                    "TableName": "",
                    "Name": "uri",
                    "Label": "uri",
                    "Type": "varchar",
                    "Precision": 2147483647,
                    "Scale": 0,
                    "Nullable": "UNKNOWN",
                    "CaseSensitive": true
                },
                {
                    "CatalogName": "hive",
                    "SchemaName": "",
                    "TableName": "",
                    "Name": "status",
                    "Label": "status",
                    "Type": "integer",
                    "Precision": 10,
                    "Scale": 0,
                    "Nullable": "UNKNOWN",
                    "CaseSensitive": false
                }
            ]
        }
    },
    "UpdateCount": 0
}
```

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的使用查詢結果、輸出檔案和查詢歷史記錄。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetQueryResults。

get-table-metadata

以下程式碼範例顯示如何使用 get-table-metadata。

AWS CLI

傳回資料表的中繼資料資訊

下列get-table-metadata範例會從AwsDataCatalog資料目錄的sampledb資料庫傳回counties資料表的中繼資料資訊，包括資料欄名稱及其資料類型。

```
aws athena get-table-metadata \
    --catalog-name AwsDataCatalog \
    --database-name sampledb \
    --table-name counties
```

輸出：

```
{
    "TableMetadata": {
        "Name": "counties",
        "CreateTime": 1593559968.0,
        "LastAccessTime": 0.0,
        "TableType": "EXTERNAL_TABLE",
        "Columns": [
            {
                "Name": "name",
                "Type": "string",
                "Comment": "from deserializer"
            },
            {
                "Name": "boundaryshape",
                "Type": "binary",
                "Comment": "from deserializer"
            },
            {
                "Name": "motto",
                "Type": "string",
                "Comment": "from deserializer"
            },
            {
                "Name": "population",
                "Type": "int",
```

```
                "Comment": "from deserializer"
            }
        ],
        "PartitionKeys": [],
        "Parameters": {
            "EXTERNAL": "TRUE",
            "inputformat": "com.esri.json.hadoop.EnclosedJsonInputFormat",
            "location": "s3://amzn-s3-demo-bucket/json",
            "outputformat":
 "org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
            "serde.param.serialization.format": "1",
            "serde.serialization.lib": "com.esri.hadoop.hive.serde.JsonSerde",
            "transient_lastDdlTime": "1593559968"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的顯示資料表詳細資訊：get-table-metadata。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTableMetadata。

get-work-group

以下程式碼範例顯示如何使用 get-work-group。

AWS CLI

傳回工作群組的相關資訊

下列get-work-group範例會傳回AthenaAdmin工作群組的相關資訊。

```
aws athena get-work-group \
    --work-group AthenaAdmin
```

輸出：

```
{
    "WorkGroup": {
        "Name": "AthenaAdmin",
        "State": "ENABLED",
        "Configuration": {
```

```
            "ResultConfiguration": {
                "OutputLocation": "s3://amzn-s3-demo-bucket/"
            },
            "EnforceWorkGroupConfiguration": false,
            "PublishCloudWatchMetricsEnabled": true,
            "RequesterPaysEnabled": false
        },
        "Description": "Workgroup for Athena administrators",
        "CreationTime": 1573677174.105
    }
}
```

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的管理工作群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetWorkGroup。

`list-data-catalogs`

以下程式碼範例顯示如何使用 `list-data-catalogs`。

AWS CLI

列出向 Athena 註冊的資料目錄

下列`list-data-catalogs`範例列出向 Athena 註冊的資料目錄。

```
aws athena list-data-catalogs
```

輸出：

```
{
    "DataCatalogsSummary": [
        {
            "CatalogName": "AwsDataCatalog",
            "Type": "GLUE"
        },
        {
            "CatalogName": "cw_logs_catalog",
            "Type": "LAMBDA"
        },
        {
            "CatalogName": "cw_metrics_catalog",
```

```
            "Type": "LAMBDA"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的列出已註冊目錄：list-data-catalogs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDataCatalogs。

`list-databases`

以下程式碼範例顯示如何使用 `list-databases`。

AWS CLI

列出資料目錄中的資料庫

下列`list-databases`範例列出`AwsDataCatalog`資料目錄中的資料庫。

```
aws athena list-databases \
    --catalog-name AwsDataCatalog
```

輸出：

```
{
    "DatabaseList": [
        {
            "Name": "default"
        },
        {
            "Name": "mydatabase"
        },
        {
            "Name": "newdb"
        },
        {
            "Name": "sampledb",
            "Description": "Sample database",
            "Parameters": {
                "CreatedBy": "Athena",
                "EXTERNAL": "TRUE"
            }
        },
```

```
        {
            "Name": "webdata"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的列出目錄中的資料庫：list-databases。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDatabases。

list-named-queries

以下程式碼範例顯示如何使用 list-named-queries。

AWS CLI

列出工作群組的具名查詢

下列list-named-queries範例列出AthenaAdmin工作群組的具名查詢。

```
aws athena list-named-queries \
    --work-group AthenaAdmin
```

輸出：

```
{
    "NamedQueryIds": [
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
    ]
}
```

如需詳細資訊，請參閱 Amazon Athena 使用者指南中的使用 Amazon Athena 執行 SQL 查詢。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListNamedQueries。

list-query-executions

以下程式碼範例顯示如何使用 list-query-executions。

AWS CLI

列出指定工作群組中查詢IDs

下列list-query-executions範例列出AthenaAdmin工作群組中最多十IDs。

```
aws athena list-query-executions \
    --work-group AthenaAdmin \
    --max-items 10
```

輸出：

```
{
    "QueryExecutionIds": [
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11110",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11114",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11115",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11116",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11117",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11118",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11119"
    ],
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMH0="
}
```

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的使用查詢結果、輸出檔案和查詢歷史記錄。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListQueryExecutions。

list-table-metadata

以下程式碼範例顯示如何使用 list-table-metadata。

AWS CLI

列出資料目錄指定資料庫中資料表的中繼資料

下列list-table-metadata範例會傳回AwsDataCatalog資料目錄geography資料庫中最多兩個資料表的中繼資料資訊。

```
aws athena list-table-metadata \
    --catalog-name AwsDataCatalog \
    --database-name geography \
    --max-items 2
```

輸出：

```
{
    "TableMetadataList": [
        {
            "Name": "country_codes",
            "CreateTime": 1586553454.0,
            "TableType": "EXTERNAL_TABLE",
            "Columns": [
                {
                    "Name": "country",
                    "Type": "string",
                    "Comment": "geo id"
                },
                {
                    "Name": "alpha-2 code",
                    "Type": "string",
                    "Comment": "geo id2"
                },
                {
                    "Name": "alpha-3 code",
                    "Type": "string",
                    "Comment": "state name"
                },
                {
                    "Name": "numeric code",
                    "Type": "bigint",
                    "Comment": ""
                },
                {
                    "Name": "latitude",
                    "Type": "bigint",
                    "Comment": "location (latitude)"
                },
                {
                    "Name": "longitude",
                    "Type": "bigint",
                    "Comment": "location (longitude)"
```

```
                }
            ],
            "Parameters": {
                "areColumnsQuoted": "false",
                "classification": "csv",
                "columnsOrdered": "true",
                "delimiter": ",",
                "has_encrypted_data": "false",
                "inputformat": "org.apache.hadoop.mapred.TextInputFormat",
                "location": "s3://amzn-s3-demo-bucket/csv/countrycode",
                "outputformat":
"org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
                "serde.param.field.delim": ",",
                "serde.serialization.lib":
"org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
                "skip.header.line.count": "1",
                "typeOfData": "file"
            }
        },
        {
            "Name": "county_populations",
            "CreateTime": 1586553446.0,
            "TableType": "EXTERNAL_TABLE",
            "Columns": [
                {
                    "Name": "id",
                    "Type": "string",
                    "Comment": "geo id"
                },
                {
                    "Name": "country",

                    "Name": "id2",
                    "Type": "string",
                    "Comment": "geo id2"
                },
                {
                    "Name": "county",
                    "Type": "string",
                    "Comment": "county name"
                },
                {
                    "Name": "state",
                    "Type": "string",
```

```
                    "Comment": "state name"
                },
                {
                    "Name": "population estimate 2018",
                    "Type": "string",
                    "Comment": ""
                }
            ],
            "Parameters": {
                "areColumnsQuoted": "false",
                "classification": "csv",
                "columnsOrdered": "true",
                "delimiter": ",",
                "has_encrypted_data": "false",
                "inputformat": "org.apache.hadoop.mapred.TextInputFormat",
                "location": "s3://amzn-s3-demo-bucket/csv/CountyPopulation",
                "outputformat":
 "org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
                "serde.param.field.delim": ",",
                "serde.serialization.lib":
 "org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
                "skip.header.line.count": "1",
                "typeOfData": "file"
            }
        }
    ],
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的顯示資料庫中所有資料表的中繼資料：list-table-metadata。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTableMetadata。

list-tags-for-resource

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

範例 1：列出工作群組的標籤

下列`list-tags-for-resource`範例列出`Data_Analyst_Group`工作群組的標籤。

```
aws athena list-tags-for-resource \
    --resource-arn arn:aws:athena:us-west-2:111122223333:workgroup/
Data_Analyst_Group
```

輸出：

```
{
    "Tags": [
        {
            "Key": "Division",
            "Value": "West"
        },
        {
            "Key": "Team",
            "Value": "Big Data"
        },
        {
            "Key": "Location",
            "Value": "Seattle"
        }
    ]
}
```

範例 2：列出資料目錄的標籤

下列list-tags-for-resource範例列出dynamo_db_catalog資料目錄的標籤。

```
aws athena list-tags-for-resource \
    --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/
dynamo_db_catalog
```

輸出：

```
{
    "Tags": [
        {
            "Key": "Division",
            "Value": "Mountain"
        },
        {
            "Key": "Organization",
            "Value": "Retail"
```

```
        },
        {
            "Key": "Product_Line",
            "Value": "Shoes"
        },
        {
            "Key": "Location",
            "Value": "Denver"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的列出資源的標籤：list-tags-for-resource。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`list-work-groups`

以下程式碼範例顯示如何使用 `list-work-groups`。

AWS CLI

列出工作群組

下列`list-work-groups`範例列出目前帳戶中的工作群組。

```
aws athena list-work-groups
```

輸出：

```
{
    "WorkGroups": [
        {
            "Name": "Data_Analyst_Group",
            "State": "ENABLED",
            "Description": "",
            "CreationTime": 1578006683.016
        },
        {
            "Name": "AthenaAdmin",
            "State": "ENABLED",
```

```
            "Description": "",
            "CreationTime": 1573677174.105
        },
        {
            "Name": "primary",
            "State": "ENABLED",
            "Description": "",
            "CreationTime": 1567465222.723
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的管理工作群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListWorkGroups。

start-query-execution

以下程式碼範例顯示如何使用 start-query-execution。

AWS CLI

範例 1：在工作群組中對指定資料庫和資料目錄中的指定資料表執行查詢

下列start-query-execution範例使用AthenaAdmin工作群組在AwsDataCatalog資料目錄中 的 cloudfront_logs cflogsdatabase資料表上執行查詢。

```
aws athena start-query-execution \
    --query-string "select date, location, browser, uri, status from cloudfront_logs
 where method = 'GET' and status = 200 and location like 'SFO%' limit 10" \
    --work-group "AthenaAdmin" \
    --query-execution-context Database=cflogsdatabase,Catalog=AwsDataCatalog
```

輸出：

```
{
"QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

如需詳細資訊，請參閱 Amazon Athena 使用者指南中的使用 Amazon Athena 執行 SQL 查詢。

範例 2：執行使用指定工作群組在指定資料目錄中建立資料庫的查詢

下列start-query-execution範例使用AthenaAdmin工作群組在預設資料目錄 newdb中建立資料庫AwsDataCatalog。

```
aws athena start-query-execution \
    --query-string "create database if not exists newdb" \
    --work-group "AthenaAdmin"
```

輸出：

```
{
"QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11112"
}
```

如需詳細資訊，請參閱 Amazon Athena 使用者指南中的使用 Amazon Athena 執行 SQL 查詢。

範例 3：執行查詢，在指定資料庫和資料目錄中的資料表上建立檢視

下列start-query-execution範例使用 中cloudfront_logs資料表上的 SELECT陳述式cflogsdatabase來建立檢視 cf10。

```
aws athena start-query-execution \
    --query-string  "CREATE OR REPLACE VIEW cf10 AS SELECT * FROM cloudfront_logs
 limit 10" \
    --query-execution-context Database=cflogsdatabase
```

輸出：

```
{
"QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11113"
}
```

如需詳細資訊，請參閱 Amazon Athena 使用者指南中的使用 Amazon Athena 執行 SQL 查詢。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartQueryExecution。

stop-query-execution

以下程式碼範例顯示如何使用 stop-query-execution。

AWS CLI

停止執行中的查詢

下列stop-query-execution範例會停止具有指定查詢 ID 的查詢。

```
aws athena stop-query-execution \
    --query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Amazon Athena 使用者指南中的使用 Amazon Athena 執行 SQL 查詢。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopQueryExecution。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤加入資源

下列tag-resource範例會將三個標籤新增至dynamo_db_catalog資料目錄。

```
aws athena tag-resource \
    --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/
dynamo_db_catalog \
    --
tags Key=Organization,Value=Retail Key=Division,Value=Mountain Key=Product_Line,Value=Shoes
```

此命令不會產生輸出。若要查看結果，請使用 aws athena list-tags-for-resource --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/dynamo_db_catalog。

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的將標籤新增至資源：tag-resource。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源移除標籤

下列untag-resource範例會從dynamo_db_catalog資料目錄資源中移除 Specialization和Focus金鑰及其相關聯的值。

```
aws athena untag-resource \
    --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/
dynamo_db_catalog \
    --tag-keys Specialization Focus
```

此命令不會產生輸出。若要查看結果，請使用 list-tags-for-resource命令。

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的從資源移除標籤：untag-resource。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-data-catalog

以下程式碼範例顯示如何使用 update-data-catalog。

AWS CLI

更新資料目錄

下列update-data-catalog範例會更新 Lambda 函數和cw_logs_catalog資料目錄的描述。

```
aws athena update-data-catalog \
    --name cw_logs_catalog \
    --type LAMBDA \
    --description "New CloudWatch Logs Catalog" \
    --function=arn:aws:lambda:us-west-2:111122223333:function:new_cw_logs_lambda
```

此命令不會產生輸出。若要查看結果，請使用 aws athena get-data-catalog --name cw_logs_catalog。

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的更新目錄：update-data-catalog。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDataCatalog。

update-work-group

以下程式碼範例顯示如何使用 update-work-group。

AWS CLI

更新工作群組

下列update-work-group範例會停用Data_Analyst_Group工作群組。使用者無法在已停用的工作群組中執行或建立查詢，但仍可以檢視指標、資料用量限制控制、工作群組設定、查詢歷史記錄和已儲存的查詢。

```
aws athena update-work-group \
    --work-group Data_Analyst_Group \
    --state DISABLED
```

此命令不會產生輸出。若要驗證狀態變更，請使用 `aws athena get-work-group --work-group Data_Analyst_Group`並檢查輸出中的 `State` 屬性。

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的管理工作群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateWorkGroup。

使用 的 Auto Scaling 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Auto Scaling 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`attach-instances`

以下程式碼範例顯示如何使用 `attach-instances`。

AWS CLI

將執行個體連接至 Auto Scaling 群組

此範例會將指定的執行個體連接至指定的 Auto Scaling 群組。

```
aws autoscaling attach-instances \
    --instance-ids i-061c63c5eb45f0416 \
    --auto-scaling-group-name my-asg
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachInstances。

attach-load-balancer-target-groups

以下程式碼範例顯示如何使用 attach-load-balancer-target-groups。

AWS CLI

將目標群組連接至 Auto Scaling 群組

此範例會將指定的目標群組連接到指定的 Auto Scaling 群組。

```
aws autoscaling attach-load-balancer-target-groups \
    --auto-scaling-group-name my-asg \
    --target-group-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Elastic Load Balancing 和 Amazon EC2 Auto Scaling。 Amazon EC2 Auto Scaling

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachLoadBalancerTargetGroups。

attach-load-balancers

以下程式碼範例顯示如何使用 attach-load-balancers。

AWS CLI

將 Classic Load Balancer 連接至 Auto Scaling 群組

此範例會將指定的 Classic Load Balancer 連接至指定的 Auto Scaling 群組。

```
aws autoscaling attach-load-balancers \
```

```
    --load-balancer-names my-load-balancer \
    --auto-scaling-group-name my-asg
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Elastic Load Balancing 和 Amazon EC2 Auto Scaling。 Amazon EC2 Auto Scaling

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachLoadBalancers。

cancel-instance-refresh

以下程式碼範例顯示如何使用 cancel-instance-refresh。

AWS CLI

若要取消執行個體重新整理

下列cancel-instance-refresh範例會取消指定 Auto Scaling 群組的進行中執行個體重新整理。

```
aws autoscaling cancel-instance-refresh \
    --auto-scaling-group-name my-asg
```

輸出：

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的取消執行個體重新整理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelInstanceRefresh。

complete-lifecycle-action

以下程式碼範例顯示如何使用 complete-lifecycle-action。

AWS CLI

完成生命週期動作

此範例會通知 Amazon EC2 Auto Scaling 指定的生命週期動作已完成，以便完成啟動或終止執行個體。

```
aws autoscaling complete-lifecycle-action \
    --lifecycle-hook-name my-launch-hook \
    --auto-scaling-group-name my-asg \
    --lifecycle-action-result CONTINUE \
    --lifecycle-action-token bcd2f1b8-9a78-44d3-8a7a-4dd07d7cf635
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Amazon EC2 Auto Scaling lifecycle hook。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CompleteLifecycleAction。

`create-auto-scaling-group`

以下程式碼範例顯示如何使用 `create-auto-scaling-group`。

AWS CLI

範例 1：建立 Auto Scaling 群組

下列`create-auto-scaling-group`範例會在區域內多個可用區域中的子網路中建立 Auto Scaling 群組。執行個體會以指定啟動範本的預設版本啟動。請注意，預設值用於大多數其他設定，例如終止政策和運作狀態檢查組態。

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateId=lt-1234567890abcde12 \
    --min-size 1 \
    --max-size 5 \
    --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Auto Scaling 群組。

範例 2：連接 Application Load Balancer、Network Load Balancer 或 Gateway Load Balancer

此範例指定支援預期流量之負載平衡器的目標群組 ARN。運作狀態檢查類型會指定 ，ELB以便在 Elastic Load Balancing 將執行個體回報為運作狀態不佳時，Auto Scaling 群組會予以取代。命令也會定義600秒的運作狀態檢查寬限期。寬限期有助於防止新啟動的執行個體提早終止。

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateId=lt-1234567890abcde12 \
    --target-group-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/943f017f100becff \
    --health-check-type ELB \
    --health-check-grace-period 600 \
    --min-size 1 \
    --max-size 5 \
    --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Elastic Load Balancing 和 Amazon EC2 Auto Scaling。 Amazon EC2 Auto Scaling

範例 3：指定置放群組並使用最新版本的啟動範本

此範例會在單一可用區域內的置放群組中啟動執行個體。這對於具有 HPC 工作負載的低延遲群組很有用。此範例也會指定群組的大小下限、大小上限和所需容量。

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateId=lt-1234567890abcde12,Version='$Latest' \
    --min-size 1 \
    --max-size 5 \
    --desired-capacity 3 \
    --placement-group my-placement-group \
    --vpc-zone-identifier "subnet-6194ea3b"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Linux 執行個體使用者指南》中的置放群組。

範例 4：指定單一執行個體 Auto Scaling 群組並使用特定版本的啟動範本

此範例會建立容量下限和上限設為 的 Auto Scaling 群組1，以強制執行一個執行個體將執行。命令也會指定啟動範本的 v1，其中指定現有 ENI 的 ID。當您使用指定 eth0 現有 ENI 的啟動範本時，您必須為符合網路界面的 Auto Scaling 群組指定可用區域，而無需在請求中指定子網路 ID。

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg-single-instance \
    --launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='1' \
    --min-size 1 \
    --max-size 1 \
    --availability-zones us-west-2a
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Auto Scaling 群組。

範例 5：指定不同的終止政策

此範例使用啟動組態建立 Auto Scaling 群組，並將終止政策設定為先終止最舊的執行個體。命令也會將標籤套用至 群組及其執行個體，金鑰為 Role，值為 WebServer。

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-configuration-name my-lc \
    --min-size 1 \
    --max-size 5 \
    --termination-policies "OldestInstance" \
    --tags "ResourceId=my-asg,ResourceType=auto-scaling-
group,Key=Role,Value=WebServer,PropagateAtLaunch=true" \
    --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的使用 Amazon EC2 Auto Scaling 終止政策。 Amazon EC2 Auto Scaling

範例 6：指定啟動生命週期關聯

此範例會建立具有生命週期掛鉤的 Auto Scaling 群組，在執行個體啟動時支援自訂動作。

```
aws autoscaling create-auto-scaling-group \
    --cli-input-json file://~/config.json
```

config.json 檔案的內容：

```
{
    "AutoScalingGroupName": "my-asg",
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-1234567890abcde12"
    },
    "LifecycleHookSpecificationList": [{
        "LifecycleHookName": "my-launch-hook",
        "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING",
        "NotificationTargetARN": "arn:aws:sqs:us-west-2:123456789012:my-sqs-queue",
        "RoleARN": "arn:aws:iam::123456789012:role/my-notification-role",
        "NotificationMetadata": "SQS message metadata",
        "HeartbeatTimeout": 4800,
        "DefaultResult": "ABANDON"
    }],
    "MinSize": 1,
    "MaxSize": 5,
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
    "Tags": [{
        "ResourceType": "auto-scaling-group",
        "ResourceId": "my-asg",
        "PropagateAtLaunch": true,
        "Value": "test",
        "Key": "environment"
    }]
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Amazon EC2 Auto Scaling lifecycle hook。

範例 7：指定終止生命週期關聯

此範例會建立具有生命週期掛鉤的 Auto Scaling 群組，在執行個體終止時支援自訂動作。

```
aws autoscaling create-auto-scaling-group \
    --cli-input-json file://~/config.json
```

config.json 的內容：

```
{
```

```
    "AutoScalingGroupName": "my-asg",
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-1234567890abcde12"
    },
    "LifecycleHookSpecificationList": [{
        "LifecycleHookName": "my-termination-hook",
        "LifecycleTransition": "autoscaling:EC2_INSTANCE_TERMINATING",
        "HeartbeatTimeout": 120,
        "DefaultResult": "CONTINUE"
    }],
    "MinSize": 1,
    "MaxSize": 5,
    "TargetGroupARNs": [
        "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067"
    ],
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Amazon EC2 Auto Scaling lifecycle hook。

範例 8：指定自訂終止政策

此範例會建立 Auto Scaling 群組，指定自訂 Lambda 函數終止政策，告知 Amazon EC2 Auto Scaling 哪些執行個體可以安全地在縮減時終止。

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg-single-instance \
    --launch-template LaunchTemplateName=my-template-for-auto-scaling \
    --min-size 1 \
    --max-size 5 \
    --termination-policies "arn:aws:lambda:us-
west-2:123456789012:function:HelloFunction:prod" \
    --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的使用 Lambda 建立自訂終止政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAutoScalingGroup。

create-launch-configuration

以下程式碼範例顯示如何使用 create-launch-configuration。

AWS CLI

範例 1：建立啟動組態

此範例會建立簡單的啟動組態。

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的建立啟動組態。

範例 2：使用安全群組、金鑰對和引導指令碼建立啟動組態

此範例會使用使用者資料中包含的安全群組、金鑰對和引導指令碼來建立啟動組態。

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large \
    --security-groups sg-eb2af88example \
    --key-name my-key-pair \
    --user-data file://myuserdata.txt
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的建立啟動組態。

範例 3：使用 IAM 角色建立啟動組態

此範例會使用 IAM 角色的執行個體描述檔名稱建立啟動組態。

```
aws autoscaling create-launch-configuration \
```

```
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large \
    --iam-instance-profile my-autoscaling-role
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的在 Amazon EC2 執行個體上執行的應用程式 IAM 角色。

範例 4：建立啟用詳細監控的啟動組態

此範例會建立啟用 EC2 詳細監控的啟動組態，並在 1 分鐘內將 EC2 指標傳送至 CloudWatch。

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large \
    --instance-monitoring Enabled=true
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的設定 Auto Scaling 執行個體的監控。 Amazon EC2 Auto Scaling

範例 5：建立啟動 Spot 執行個體的啟動組態

此範例會建立使用 Spot 執行個體做為唯一購買選項的啟動組態。

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large \
    --spot-price "0.50"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的請求 Spot 執行個體。

範例 6：使用 EC2 執行個體建立啟動組態

此範例會根據現有執行個體的屬性建立啟動組態。它會覆寫置放租用，以及是否透過包含 `--placement-tenancy`和 `--no-associate-public-ip-address`選項來設定公有 IP 地址。

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc-from-instance \
    --instance-id i-0123a456700123456 \
    --instance-type m5.large \
    --no-associate-public-ip-address \
    --placement-tenancy dedicated
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的使用 EC2 執行個體建立啟動組態。 Amazon EC2 Auto Scaling

範例 7：使用 Amazon EBS 磁碟區的區塊型設備映射建立啟動組態

此範例會為裝置名稱/dev/sdh為 且磁碟區大小為 20 的 Amazon EBS gp3磁碟區建立具有區塊型裝置映射的啟動組態。

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large \
    --block-device-mappings '[{"DeviceName":"/dev/sdh","Ebs":
{"VolumeSize":20,"VolumeType":"gp3"}}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling API 參考》中的 EBS。

如需引用 JSON 格式參數值的語法資訊，請參閱《 AWS 命令列界面使用者指南》中的使用引號搭配 CLI AWS 中的字串。

範例 8：使用執行個體存放區磁碟區的區塊型設備映射建立啟動組態

此範例使用 建立啟動組態ephemeral1，做為裝置名稱為 的執行個體存放磁碟區/dev/sdc。

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large \
    --block-device-mappings '[{"DeviceName":"/dev/sdc","VirtualName":"ephemeral1"}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling API 參考》中的 BlockDeviceMapping。Amazon EC2 Auto Scaling

如需引用 JSON 格式參數值的語法資訊，請參閱《 AWS 命令列界面使用者指南》中的使用引號搭配 CLI AWS 中的字串。

範例 9：建立啟動組態並禁止區塊型設備在啟動時連接

此範例會建立啟動組態，以抑制 AMI 區塊型設備映射所指定的區塊型設備 （ 例如，/dev/sdf)。

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large \
    --block-device-mappings '[{"DeviceName":"/dev/sdf","NoDevice":""}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling API 參考》中的 BlockDeviceMapping。Amazon EC2 Auto Scaling

如需引用 JSON 格式參數值的語法資訊，請參閱《 AWS 命令列界面使用者指南》中的使用引號搭配 CLI AWS 中的字串。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLaunchConfiguration。

create-or-update-tags

以下程式碼範例顯示如何使用 create-or-update-tags。

AWS CLI

建立或更新 Auto Scaling 群組的標籤

此範例會將兩個標籤新增至指定的 Auto Scaling 群組。

```
aws autoscaling create-or-update-tags \
    --tags ResourceId=my-asg,ResourceType=auto-scaling-
group,Key=Role,Value=WebServer,PropagateAtLaunch=true ResourceId=my-
asg,ResourceType=auto-scaling-group,Key=Dept,Value=Research,PropagateAtLaunch=true
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的標記 Auto Scaling 群組和執行個體。 Amazon EC2 Auto Scaling

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateOrUpdateTags。

delete-auto-scaling-group

以下程式碼範例顯示如何使用 delete-auto-scaling-group。

AWS CLI

範例 1：刪除指定的 Auto Scaling 群組

此範例會刪除指定的 Auto Scaling 群組。

```
aws autoscaling delete-auto-scaling-group \
    --auto-scaling-group-name my-asg
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的刪除 Auto Scaling 基礎設施。 Amazon EC2 Auto Scaling

範例 2：強制刪除指定的 Auto Scaling 群組

若要刪除 Auto Scaling 群組，而不等待群組中的執行個體終止，請使用 --force-delete選項。

```
aws autoscaling delete-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --force-delete
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的刪除 Auto Scaling 基礎設施。 Amazon EC2 Auto Scaling

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAutoScalingGroup。

delete-launch-configuration

以下程式碼範例顯示如何使用 delete-launch-configuration。

AWS CLI

刪除啟動組態

此範例會刪除指定的啟動組態。

```
aws autoscaling delete-launch-configuration \
    --launch-configuration-name my-launch-config
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的刪除 Auto Scaling 基礎設施。 Amazon EC2 Auto Scaling

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLaunchConfiguration。

delete-lifecycle-hook

以下程式碼範例顯示如何使用 delete-lifecycle-hook。

AWS CLI

刪除生命週期掛鉤

此範例會刪除指定的生命週期關聯。

```
aws autoscaling delete-lifecycle-hook \
    --lifecycle-hook-name my-lifecycle-hook \
    --auto-scaling-group-name my-asg
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLifecycleHook。

delete-notification-configuration

以下程式碼範例顯示如何使用 delete-notification-configuration。

AWS CLI

刪除 Auto Scaling 通知

此範例會從指定的 Auto Scaling 群組刪除指定的通知。

```
aws autoscaling delete-notification-configuration \
    --auto-scaling-group-name my-asg \
    --topic-arn arn:aws:sns:us-west-2:123456789012:my-sns-topic
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的刪除通知組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNotificationConfiguration。

delete-policy

以下程式碼範例顯示如何使用 delete-policy。

AWS CLI

刪除擴展政策

此範例會刪除指定的擴展政策。

```
aws autoscaling delete-policy \
    --auto-scaling-group-name my-asg \
    --policy-name alb1000-target-tracking-scaling-policy
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeletePolicy。

delete-scheduled-action

以下程式碼範例顯示如何使用 delete-scheduled-action。

AWS CLI

從 Auto Scaling 群組刪除排程動作

此範例會從指定的 Auto Scaling 群組刪除指定的排程動作。

```
aws autoscaling delete-scheduled-action \
    --auto-scaling-group-name my-asg \
    --scheduled-action-name my-scheduled-action
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteScheduledAction。

delete-tags

以下程式碼範例顯示如何使用 delete-tags。

AWS CLI

從 Auto Scaling 群組刪除標籤

此範例會從指定的 Auto Scaling 群組刪除指定的標籤。

```
aws autoscaling delete-tags \
    --tags ResourceId=my-asg,ResourceType=auto-scaling-group,Key=Dept,Value=Research
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的標記 Auto Scaling 群組和執行個體。 Amazon EC2 Auto Scaling

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTags。

delete-warm-pool

以下程式碼範例顯示如何使用 delete-warm-pool。

AWS CLI

範例 1：刪除暖集區

下列範例會刪除指定 Auto Scaling 群組的暖集區。

```
aws autoscaling delete-warm-pool \
    --auto-scaling-group-name my-asg
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Amazon EC2 Auto Scaling 的暖集區。

範例 2：強制刪除暖集區

若要刪除暖集區而不等待其執行個體終止，請使用 `--force-delete`選項。

```
aws autoscaling delete-warm-pool \
    --auto-scaling-group-name my-asg \
    --force-delete
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Amazon EC2 Auto Scaling 的暖集區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteWarmPool。

describe-account-limits

以下程式碼範例顯示如何使用 `describe-account-limits`。

AWS CLI

描述您的 Amazon EC2 Auto Scaling 帳戶限制

此範例說明您 AWS 帳戶的 Amazon EC2 Auto Scaling 限制。

```
aws autoscaling describe-account-limits
```

輸出：

```
{
    "NumberOfLaunchConfigurations": 5,
    "MaxNumberOfLaunchConfigurations": 100,
    "NumberOfAutoScalingGroups": 3,
    "MaxNumberOfAutoScalingGroups": 20
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Amazon EC2 Auto Scaling 服務配額。 Amazon EC2 Auto Scaling

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAccountLimits。

describe-adjustment-types

以下程式碼範例顯示如何使用 `describe-adjustment-types`。

AWS CLI

描述可用的擴展調整類型

此範例說明可用的調整類型。

```
aws autoscaling describe-adjustment-types
```

輸出：

```
{
    "AdjustmentTypes": [
        {
            "AdjustmentType": "ChangeInCapacity"
        },
        {
            "AdjustmentType": "ExactCapacity"
        },
        {
            "AdjustmentType": "PercentChangeInCapacity"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的擴展調整類型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAdjustmentTypes。

describe-auto-scaling-groups

以下程式碼範例顯示如何使用 describe-auto-scaling-groups。

AWS CLI

範例 1：描述指定的 Auto Scaling 群組

此範例說明指定的 Auto Scaling 群組。

```
aws autoscaling describe-auto-scaling-groups \
    --auto-scaling-group-names my-asg
```

輸出：

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupName": "my-asg",
            "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:930d940e-891e-4781-
a11a-7b0acd480f03:autoScalingGroupName/my-asg",
            "LaunchTemplate": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1",
                "LaunchTemplateId": "lt-1234567890abcde12"
            },
            "MinSize": 0,
            "MaxSize": 1,
            "DesiredCapacity": 1,
            "DefaultCooldown": 300,
            "AvailabilityZones": [
                "us-west-2a",
                "us-west-2b",
                "us-west-2c"
            ],
            "LoadBalancerNames": [],
            "TargetGroupARNs": [],
            "HealthCheckType": "EC2",
            "HealthCheckGracePeriod": 0,
            "Instances": [
                {
                    "InstanceId": "i-06905f55584de02da",
                    "InstanceType": "t2.micro",
                    "AvailabilityZone": "us-west-2a",
                    "HealthStatus": "Healthy",
                    "LifecycleState": "InService",
                    "ProtectedFromScaleIn": false,
                    "LaunchTemplate": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "1",
                        "LaunchTemplateId": "lt-1234567890abcde12"
                    }
                }
            ],
            "CreatedTime": "2023-10-28T02:39:22.152Z",
            "SuspendedProcesses": [],
```

```
            "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
            "EnabledMetrics": [],
            "Tags": [],
            "TerminationPolicies": [
                "Default"
            ],
            "NewInstancesProtectedFromScaleIn": false,
            "ServiceLinkedRoleARN":"arn",
            "TrafficSources": []
        }
    ]
}
```

範例 2：描述前 100 個指定的 Auto Scaling 群組

此範例說明指定的 Auto Scaling 群組。它可讓您指定最多 100 個群組名稱。

```
aws autoscaling describe-auto-scaling-groups \
    --max-items 100 \
    --auto-scaling-group-names "group1" "group2" "group3" "group4"
```

如需範例輸出，請參閱範例 1。

範例 3：描述指定區域中的 Auto Scaling 群組

此範例說明指定區域中的 Auto Scaling 群組，最多 75 個群組。

```
aws autoscaling describe-auto-scaling-groups \
    --max-items 75 \
    --region us-east-1
```

如需範例輸出，請參閱範例 1。

範例 4：描述 Auto Scaling 群組的指定數量

若要傳回特定數量的 Auto Scaling 群組，請使用 `--max-items`選項。

```
aws autoscaling describe-auto-scaling-groups \
    --max-items 1
```

如需範例輸出，請參閱範例 1。

如果輸出包含 NextToken 欄位，則有更多群組。若要取得其他群組，請在後續呼叫中使用此欄位的值與 `--starting-token`選項，如下所示。

```
aws autoscaling describe-auto-scaling-groups \
    --starting-token Z3M3LMPEXAMPLE
```

如需範例輸出，請參閱範例 1。

範例 5：描述使用啟動組態的 Auto Scaling 群組

此範例使用 `--query`選項來描述使用啟動組態的 Auto Scaling 群組。

```
aws autoscaling describe-auto-scaling-groups \
    --query 'AutoScalingGroups[?LaunchConfigurationName!=`null`]'
```

輸出：

```
[
    {
        "AutoScalingGroupName": "my-asg",
        "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:930d940e-891e-4781-
a11a-7b0acd480f03:autoScalingGroupName/my-asg",
        "LaunchConfigurationName": "my-lc",
        "MinSize": 0,
        "MaxSize": 1,
        "DesiredCapacity": 1,
        "DefaultCooldown": 300,
        "AvailabilityZones": [
            "us-west-2a",
            "us-west-2b",
            "us-west-2c"
        ],
        "LoadBalancerNames": [],
        "TargetGroupARNs": [],
        "HealthCheckType": "EC2",
        "HealthCheckGracePeriod": 0,
        "Instances": [
            {
                "InstanceId": "i-088c57934a6449037",
                "InstanceType": "t2.micro",
                "AvailabilityZone": "us-west-2c",
```

```
                "HealthStatus": "Healthy",
                "LifecycleState": "InService",
                "LaunchConfigurationName": "my-lc",
                "ProtectedFromScaleIn": false
            }
        ],
        "CreatedTime": "2023-10-28T02:39:22.152Z",
        "SuspendedProcesses": [],
        "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
        "EnabledMetrics": [],
        "Tags": [],
        "TerminationPolicies": [
            "Default"
        ],
        "NewInstancesProtectedFromScaleIn": false,
        "ServiceLinkedRoleARN":"arn",
        "TrafficSources": []
    }
]
```

如需詳細資訊，請參閱《 AWS 命令列界面使用者指南》中的篩選 AWS CLI 輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAutoScalingGroups。

describe-auto-scaling-instances

以下程式碼範例顯示如何使用 describe-auto-scaling-instances。

AWS CLI

範例 1：描述一或多個執行個體

此範例說明指定的執行個體。

```
aws autoscaling describe-auto-scaling-instances \
    --instance-ids i-06905f55584de02da
```

輸出：

```
{
    "AutoScalingInstances": [
        {
            "InstanceId": "i-06905f55584de02da",
```

```
            "InstanceType": "t2.micro",
            "AutoScalingGroupName": "my-asg",
            "AvailabilityZone": "us-west-2b",
            "LifecycleState": "InService",
            "HealthStatus": "HEALTHY",
            "ProtectedFromScaleIn": false,
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-1234567890abcde12",
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            }
        }
    ]
}
```

範例 2：描述一或多個執行個體

此範例使用 `--max-items`選項來指定此呼叫要傳回的執行個體數量。

```
aws autoscaling describe-auto-scaling-instances \
    --max-items 1
```

如果輸出包含 NextToken 欄位，則有更多執行個體。若要取得其他執行個體，請在後續呼叫中使用此欄位的值與 `--starting-token`選項，如下所示。

```
aws autoscaling describe-auto-scaling-instances \
    --starting-token Z3M3LMPEXAMPLE
```

如需範例輸出，請參閱範例 1。

範例 3：描述使用啟動組態的執行個體

此範例使用 `--query`選項來描述使用啟動組態的執行個體。

```
aws autoscaling describe-auto-scaling-instances \
    --query 'AutoScalingInstances[?LaunchConfigurationName!=`null`]'
```

輸出：

```
[
    {
        "InstanceId": "i-088c57934a6449037",
```

```
        "InstanceType": "t2.micro",
        "AutoScalingGroupName": "my-asg",
        "AvailabilityZone": "us-west-2c",
        "LifecycleState": "InService",
        "HealthStatus": "HEALTHY",
        "LaunchConfigurationName": "my-lc",
        "ProtectedFromScaleIn": false
    }
]
```

如需詳細資訊，請參閱《 AWS 命令列界面使用者指南》中的篩選 AWS CLI 輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAutoScalingInstances。

describe-auto-scaling-notification-types

以下程式碼範例顯示如何使用 describe-auto-scaling-notification-types。

AWS CLI

描述可用的通知類型

此範例說明可用的通知類型。

```
aws autoscaling describe-auto-scaling-notification-types
```

輸出：

```
{
    "AutoScalingNotificationTypes": [
        "autoscaling:EC2_INSTANCE_LAUNCH",
        "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
        "autoscaling:EC2_INSTANCE_TERMINATE",
        "autoscaling:EC2_INSTANCE_TERMINATE_ERROR",
        "autoscaling:TEST_NOTIFICATION"
    ]
}
```

有關詳細資訊，請參閱 Amazon EC2 Auto Scaling 使用者指南中的取得 Auto Scaling 群組擴展時的 Amazon SNS 通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAutoScalingNotificationTypes。

describe-instance-refreshes

以下程式碼範例顯示如何使用 describe-instance-refreshes。

AWS CLI

描述執行個體重新整理

下列describe-instance-refreshes範例會傳回指定 Auto Scaling 群組所有執行個體重新整理請求的描述，包括狀態訊息和（如果可用）狀態原因。

```
aws autoscaling describe-instance-refreshes \
    --auto-scaling-group-name my-asg
```

輸出：

```
{
    "InstanceRefreshes": [
        {
            "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b",
            "AutoScalingGroupName": "my-asg",
            "Status": "InProgress",
            "StatusReason": "Waiting for instances to warm up before continuing. For
 example: 0e69cc3f05f825f4f is warming up.",
            "EndTime": "2023-03-23T16:42:55Z",
            "PercentageComplete": 0,
            "InstancesToUpdate": 0,
    "Preferences": {
                "MinHealthyPercentage": 100,
                "InstanceWarmup": 300,
                "CheckpointPercentages": [
                    50
                ],
                "CheckpointDelay": 3600,
                "SkipMatching": false,
                "AutoRollback": true,
                "ScaleInProtectedInstances": "Ignore",
                "StandbyInstances": "Ignore"
            }
        },
        {
            "InstanceRefreshId": "dd7728d0-5bc4-4575-96a3-1b2c52bf8bb1",
            "AutoScalingGroupName": "my-asg",
```

```
            "Status": "Successful",
            "EndTime": "2022-06-02T16:53:37Z",
            "PercentageComplete": 100,
            "InstancesToUpdate": 0,
    "Preferences": {
                "MinHealthyPercentage": 90,
                "InstanceWarmup": 300,
                "SkipMatching": true,
                "AutoRollback": true,
                "ScaleInProtectedInstances": "Ignore",
                "StandbyInstances": "Ignore"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的檢查執行個體重新整理的狀態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInstanceRefreshes。

describe-launch-configurations

以下程式碼範例顯示如何使用 describe-launch-configurations。

AWS CLI

範例 1：描述指定的啟動組態

此範例說明指定的啟動組態。

```
aws autoscaling describe-launch-configurations \
    --launch-configuration-names my-launch-config
```

輸出：

```
{
    "LaunchConfigurations": [
        {
            "LaunchConfigurationName": "my-launch-config",
            "LaunchConfigurationARN": "arn:aws:autoscaling:us-
west-2:123456789012:launchConfiguration:98d3b196-4cf9-4e88-8ca1-8547c24ced8b:launchConfigura
my-launch-config",
```

```
            "ImageId": "ami-0528a5175983e7f28",
            "KeyName": "my-key-pair-uswest2",
            "SecurityGroups": [
                "sg-05eaec502fcdadc2e"
            ],
            "ClassicLinkVPCSecurityGroups": [],
            "UserData": "",
            "InstanceType": "t2.micro",
            "KernelId": "",
            "RamdiskId": "",
            "BlockDeviceMappings": [
                {
                    "DeviceName": "/dev/xvda",
                    "Ebs": {
                        "SnapshotId": "snap-06c1606ba5ca274b1",
                        "VolumeSize": 8,
                        "VolumeType": "gp2",
                        "DeleteOnTermination": true,
                        "Encrypted": false
                    }
                }
            ],
            "InstanceMonitoring": {
                "Enabled": true
            },
            "CreatedTime": "2020-10-28T02:39:22.321Z",
            "EbsOptimized": false,
            "AssociatePublicIpAddress": true,
            "MetadataOptions": {
                "HttpTokens": "required",
                "HttpPutResponseHopLimit": 1,
                "HttpEndpoint": "disabled"
            }
        }
    ]
}
```

範例 2：描述指定數量的啟動組態

若要傳回特定數量的啟動組態，請使用 `--max-items`選項。

```
aws autoscaling describe-launch-configurations \
    --max-items 1
```

如果輸出包含 NextToken 欄位，則有更多啟動組態。若要取得其他啟動組態，請在後續呼叫中使用此欄位的值與 --starting-token選項，如下所示。

```
aws autoscaling describe-launch-configurations \
    --starting-token Z3M3LMPEXAMPLE
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLaunchConfigurations。

describe-lifecycle-hook-types

以下程式碼範例顯示如何使用 describe-lifecycle-hook-types。

AWS CLI

描述可用的生命週期關聯類型

此範例說明可用的生命週期關聯類型。

```
aws autoscaling describe-lifecycle-hook-types
```

輸出：

```
{
    "LifecycleHookTypes": [
        "autoscaling:EC2_INSTANCE_LAUNCHING",
        "autoscaling:EC2_INSTANCE_TERMINATING"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLifecycleHookTypes。

describe-lifecycle-hooks

以下程式碼範例顯示如何使用 describe-lifecycle-hooks。

AWS CLI

描述您的生命週期關聯

此範例說明指定 Auto Scaling 群組的生命週期關聯。

```
aws autoscaling describe-lifecycle-hooks \
    --auto-scaling-group-name my-asg
```

輸出：

```
{
    "LifecycleHooks": [
        {
            "GlobalTimeout": 3000,
            "HeartbeatTimeout": 30,
            "AutoScalingGroupName": "my-asg",
            "LifecycleHookName": "my-launch-hook",
            "DefaultResult": "ABANDON",
            "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING"
        },
        {
            "GlobalTimeout": 6000,
            "HeartbeatTimeout": 60,
            "AutoScalingGroupName": "my-asg",
            "LifecycleHookName": "my-termination-hook",
            "DefaultResult": "CONTINUE",
            "LifecycleTransition": "autoscaling:EC2_INSTANCE_TERMINATING"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLifecycleHooks。

describe-load-balancer-target-groups

以下程式碼範例顯示如何使用 describe-load-balancer-target-groups。

AWS CLI

描述 Auto Scaling 群組的負載平衡器目標群組

此範例說明連接至指定 Auto Scaling 群組的負載平衡器目標群組。

```
aws autoscaling describe-load-balancer-target-groups \
    --auto-scaling-group-name my-asg
```

輸出：

```
{
    "LoadBalancerTargetGroups": [
        {
            "LoadBalancerTargetGroupARN": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
            "State": "Added"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLoadBalancerTargetGroups。

describe-load-balancers

以下程式碼範例顯示如何使用 describe-load-balancers。

AWS CLI

描述 Auto Scaling 群組的 Classic Load Balancer

此範例說明指定 Auto Scaling 群組的 Classic Load Balancer。

```
aws autoscaling describe-load-balancers \
    --auto-scaling-group-name my-asg
```

輸出：

```
{
    "LoadBalancers": [
        {
            "State": "Added",
            "LoadBalancerName": "my-load-balancer"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLoadBalancers。

describe-metric-collection-types

以下程式碼範例顯示如何使用 describe-metric-collection-types。

AWS CLI

描述可用的指標集合類型

此範例說明可用的指標集合類型。

```
aws autoscaling describe-metric-collection-types
```

輸出：

```
{
    "Metrics": [
        {
            "Metric": "GroupMinSize"
        },
        {
            "Metric": "GroupMaxSize"
        },
        {
            "Metric": "GroupDesiredCapacity"
        },
        {
            "Metric": "GroupInServiceInstances"
        },
        {
            "Metric": "GroupInServiceCapacity"
        },
        {
            "Metric": "GroupPendingInstances"
        },
        {
            "Metric": "GroupPendingCapacity"
        },
        {
            "Metric": "GroupTerminatingInstances"
        },
        {
            "Metric": "GroupTerminatingCapacity"
        },
        {
            "Metric": "GroupStandbyInstances"
        },
        {
```

```
            "Metric": "GroupStandbyCapacity"
        },
        {
            "Metric": "GroupTotalInstances"
        },
        {
            "Metric": "GroupTotalCapacity"
        }
    ],
    "Granularities": [
        {
            "Granularity": "1Minute"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的Auto Scaling 群組指標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeMetricCollectionTypes。

describe-notification-configurations

以下程式碼範例顯示如何使用 describe-notification-configurations。

AWS CLI

範例 1：描述指定群組的通知組態

此範例說明指定 Auto Scaling 群組的通知組態。

```
aws autoscaling describe-notification-configurations \
    --auto-scaling-group-name my-asg
```

輸出：

```
{
    "NotificationConfigurations": [
        {
            "AutoScalingGroupName": "my-asg",
            "NotificationType": "autoscaling:TEST_NOTIFICATION",
            "TopicARN": "arn:aws:sns:us-west-2:123456789012:my-sns-topic-2"
        },
        {
```

```
            "AutoScalingGroupName": "my-asg",
            "NotificationType": "autoscaling:TEST_NOTIFICATION",
            "TopicARN": "arn:aws:sns:us-west-2:123456789012:my-sns-topic"
        }
    ]
}
```

有關詳細資訊，請參閱 Amazon EC2 Auto Scaling 使用者指南中的取得 Auto Scaling 群組擴展時的 Amazon SNS 通知。

範例 1：描述指定數量的通知組態

若要傳回特定數量的通知組態，請使用 `max-items` 參數。

```
aws autoscaling describe-notification-configurations \
    --auto-scaling-group-name my-auto-scaling-group \
    --max-items 1
```

輸出：

```
{
    "NotificationConfigurations": [
        {
            "AutoScalingGroupName": "my-asg",
            "NotificationType": "autoscaling:TEST_NOTIFICATION",
            "TopicARN": "arn:aws:sns:us-west-2:123456789012:my-sns-topic-2"
        },
        {
            "AutoScalingGroupName": "my-asg",
            "NotificationType": "autoscaling:TEST_NOTIFICATION",
            "TopicARN": "arn:aws:sns:us-west-2:123456789012:my-sns-topic"
        }
    ]
}
```

如果輸出包含 `NextToken` 欄位，則有更多通知組態。若要取得額外的通知組態，請在後續呼叫中使用此欄位的值搭配 `starting-token` 參數，如下所示。

```
aws autoscaling describe-notification-configurations \
    --auto-scaling-group-name my-asg \
    --starting-token Z3M3LMPEXAMPLE
```

有關詳細資訊，請參閱 Amazon EC2 Auto Scaling 使用者指南中的取得 Auto Scaling 群組擴展時的 Amazon SNS 通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeNotificationConfigurations。

describe-policies

以下程式碼範例顯示如何使用 describe-policies。

AWS CLI

範例 1：描述指定群組的擴展政策

此範例說明指定 Auto Scaling 群組的擴展政策。

```
aws autoscaling describe-policies \
    --auto-scaling-group-name my-asg
```

輸出：

```
{
    "ScalingPolicies": [
        {
            "AutoScalingGroupName": "my-asg",
            "PolicyName": "alb1000-target-tracking-scaling-policy",
            "PolicyARN": "arn:aws:autoscaling:us-
west-2:123456789012:scalingPolicy:3065d9c8-9969-4bec-
bb6a-3fbe5550fde6:autoScalingGroupName/my-asg:policyName/alb1000-target-tracking-
scaling-policy",
            "PolicyType": "TargetTrackingScaling",
            "StepAdjustments": [],
            "Alarms": [
                {
                    "AlarmName": "TargetTracking-my-asg-
AlarmHigh-924887a9-12d7-4e01-8686-6f844d13a196",
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-
AlarmHigh-924887a9-12d7-4e01-8686-6f844d13a196"
                },
                {
                    "AlarmName": "TargetTracking-my-asg-AlarmLow-f96f899d-b8e7-4d09-
a010-c1aaa35da296",
```

```
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-f96f899d-b8e7-4d09-a010-
c1aaa35da296"
                }
            ],
            "TargetTrackingConfiguration": {
                "PredefinedMetricSpecification": {
                    "PredefinedMetricType": "ALBRequestCountPerTarget",
                    "ResourceLabel": "app/my-alb/778d41231b141a0f/targetgroup/my-
alb-target-group/943f017f100becff"
                },
                "TargetValue": 1000.0,
                "DisableScaleIn": false
            },
            "Enabled": true
        },
        {
            "AutoScalingGroupName": "my-asg",
            "PolicyName": "cpu40-target-tracking-scaling-policy",
            "PolicyARN": "arn:aws:autoscaling:us-
west-2:123456789012:scalingPolicy:5fd26f71-39d4-4690-82a9-
b8515c45cdde:autoScalingGroupName/my-asg:policyName/cpu40-target-tracking-scaling-
policy",
            "PolicyType": "TargetTrackingScaling",
            "StepAdjustments": [],
            "Alarms": [
                {
                    "AlarmName": "TargetTracking-my-asg-
AlarmHigh-139f9789-37b9-42ad-bea5-b5b147d7f473",
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmHigh-139f9789-37b9-42ad-bea5-
b5b147d7f473"
                },
                {
                    "AlarmName": "TargetTracking-my-asg-AlarmLow-bd681c67-
fc18-4c56-8468-fb8e413009c9",
                    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-bd681c67-fc18-4c56-8468-
fb8e413009c9"
                }
            ],
            "TargetTrackingConfiguration": {
                "PredefinedMetricSpecification": {
                    "PredefinedMetricType": "ASGAverageCPUUtilization"
```

```
                },
                "TargetValue": 40.0,
                "DisableScaleIn": false
            },
            "Enabled": true
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的動態擴展。

範例 2：描述指定名稱的擴展政策

若要傳回特定擴展政策，請使用 `--policy-names`選項。

```
aws autoscaling describe-policies \
    --auto-scaling-group-name my-asg \
    --policy-names cpu40-target-tracking-scaling-policy
```

如需範例輸出，請參閱範例 1。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的動態擴展。

範例 3：描述數個擴展政策

若要傳回特定數量的政策，請使用 `--max-items`選項。

```
aws autoscaling describe-policies \
    --auto-scaling-group-name my-asg \
    --max-items 1
```

如需範例輸出，請參閱範例 1。

如果輸出包含 `NextToken` 欄位，請在後續呼叫中使用此欄位的值與 `--starting-token`選項，以取得其他政策。

```
aws autoscaling describe-policies --auto-scaling-group-name my-asg --starting-
token Z3M3LMPEXAMPLE
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的動態擴展。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePolicies。

describe-scaling-activities

以下程式碼範例顯示如何使用 describe-scaling-activities。

AWS CLI

範例 1：描述指定群組的擴展活動

此範例說明指定 Auto Scaling 群組的擴展活動。

```
aws autoscaling describe-scaling-activities \
    --auto-scaling-group-name my-asg
```

輸出：

```
{
    "Activities": [
        {
            "ActivityId": "f9f2d65b-f1f2-43e7-b46d-d86756459699",
            "Description": "Launching a new EC2 instance: i-0d44425630326060f",
            "AutoScalingGroupName": "my-asg",
            "Cause": "At 2020-10-30T19:35:51Z a user request update of
 AutoScalingGroup constraints to min: 0, max: 16, desired: 16 changing the desired
 capacity from 0 to 16.  At 2020-10-30T19:36:07Z an instance was started in response
 to a difference between desired and actual capacity, increasing the capacity from 0
 to 16.",
            "StartTime": "2020-10-30T19:36:09.766Z",
            "EndTime": "2020-10-30T19:36:41Z",
            "StatusCode": "Successful",
            "Progress": 100,
            "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":
\"us-west-2b\"}"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的驗證 Auto Scaling 群組的擴展活動。

範例 2：描述已刪除群組的擴展活動

若要描述已刪除 Auto Scaling 群組之後的擴展活動，請新增 --include-deleted-groups選項。

```
aws autoscaling describe-scaling-activities \
    --auto-scaling-group-name my-asg \
    --include-deleted-groups
```

輸出：

```
{
    "Activities": [
        {
            "ActivityId": "e1f5de0e-f93e-1417-34ac-092a76fba220",
            "Description": "Launching a new EC2 instance.  Status Reason: Your Spot
 request price of 0.001 is lower than the minimum required Spot request fulfillment
 price of 0.0031. Launching EC2 instance failed.",
            "AutoScalingGroupName": "my-asg",
            "Cause": "At 2021-01-13T20:47:24Z a user request update of
 AutoScalingGroup constraints to min: 1, max: 5, desired: 3 changing the desired
 capacity from 0 to 3.  At 2021-01-13T20:47:27Z an instance was started in response
 to a difference between desired and actual capacity, increasing the capacity from 0
 to 3.",
            "StartTime": "2021-01-13T20:47:30.094Z",
            "EndTime": "2021-01-13T20:47:30Z",
            "StatusCode": "Failed",
            "StatusMessage": "Your Spot request price of 0.001 is lower than the
 minimum required Spot request fulfillment price of 0.0031. Launching EC2 instance
 failed.",
            "Progress": 100,
            "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":
\"us-west-2b\"}",
            "AutoScalingGroupState": "Deleted",
            "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:283179a2-
f3ce-423d-93f6-66bb518232f7:autoScalingGroupName/my-asg"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的疑難排解 Amazon EC2 Auto Scaling。

範例 3：描述指定數量的擴展活動

若要傳回特定數量的活動，請使用 `--max-items`選項。

```
aws autoscaling describe-scaling-activities \
    --max-items 1
```

輸出：

```
{
    "Activities": [
        {
            "ActivityId": "f9f2d65b-f1f2-43e7-b46d-d86756459699",
            "Description": "Launching a new EC2 instance: i-0d44425630326060f",
            "AutoScalingGroupName": "my-asg",
            "Cause": "At 2020-10-30T19:35:51Z a user request update of
 AutoScalingGroup constraints to min: 0, max: 16, desired: 16 changing the desired
 capacity from 0 to 16.  At 2020-10-30T19:36:07Z an instance was started in response
 to a difference between desired and actual capacity, increasing the capacity from 0
 to 16.",
            "StartTime": "2020-10-30T19:36:09.766Z",
            "EndTime": "2020-10-30T19:36:41Z",
            "StatusCode": "Successful",
            "Progress": 100,
            "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":
\"us-west-2b\"}"
        }
    ]
}
```

如果輸出包含 NextToken 欄位，則有更多活動。若要取得其他活動，請在後續呼叫中使用此欄位的值與 `--starting-token`選項，如下所示。

```
aws autoscaling describe-scaling-activities \
    --starting-token Z3M3LMPEXAMPLE
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的驗證 Auto Scaling 群組的擴展活動。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeScalingActivities。

describe-scaling-process-types

以下程式碼範例顯示如何使用 `describe-scaling-process-types`。

AWS CLI

描述可用的程序類型

此範例說明可用的程序類型。

```
aws autoscaling describe-scaling-process-types
```

輸出：

```
{
    "Processes": [
        {
            "ProcessName": "AZRebalance"
        },
        {
            "ProcessName": "AddToLoadBalancer"
        },
        {
            "ProcessName": "AlarmNotification"
        },
        {
            "ProcessName": "HealthCheck"
        },
        {
            "ProcessName": "InstanceRefresh"
        },
        {
            "ProcessName": "Launch"
        },
        {
            "ProcessName": "ReplaceUnhealthy"
        },
        {
            "ProcessName": "ScheduledActions"
        },
        {
            "ProcessName": "Terminate"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的暫停和繼續擴展程序。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeScalingProcessTypes。

describe-scheduled-actions

以下程式碼範例顯示如何使用 describe-scheduled-actions。

AWS CLI

範例 1：描述所有排程動作

此範例說明所有排定的動作。

```
aws autoscaling describe-scheduled-actions
```

輸出：

```
{
    "ScheduledUpdateGroupActions": [
        {
            "AutoScalingGroupName": "my-asg",
            "ScheduledActionName": "my-recurring-action",
            "Recurrence": "30 0 1 1,6,12 *",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
            "StartTime": "2023-12-01T04:00:00Z",
            "Time": "2023-12-01T04:00:00Z",
            "MinSize": 1,
            "MaxSize": 6,
            "DesiredCapacity": 4,
            "TimeZone": "America/New_York"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的排程擴展。

範例 2：描述指定群組的排程動作

若要描述特定 Auto Scaling 群組的排程動作，請使用 --auto-scaling-group-name選項。

```
aws autoscaling describe-scheduled-actions \
```

```
    --auto-scaling-group-name my-asg
```

輸出：

```
{
    "ScheduledUpdateGroupActions": [
        {
            "AutoScalingGroupName": "my-asg",
            "ScheduledActionName": "my-recurring-action",
            "Recurrence": "30 0 1 1,6,12 *",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
            "StartTime": "2023-12-01T04:00:00Z",
            "Time": "2023-12-01T04:00:00Z",
            "MinSize": 1,
            "MaxSize": 6,
            "DesiredCapacity": 4,
            "TimeZone": "America/New_York"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的排程擴展。

範例 3：描述指定的排程動作

若要描述特定的排程動作，請使用 `--scheduled-action-names`選項。

```
aws autoscaling describe-scheduled-actions \
    --scheduled-action-names my-recurring-action
```

輸出：

```
{
    "ScheduledUpdateGroupActions": [
        {
            "AutoScalingGroupName": "my-asg",
            "ScheduledActionName": "my-recurring-action",
            "Recurrence": "30 0 1 1,6,12 *",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
```

```
            "StartTime": "2023-12-01T04:00:00Z",
            "Time": "2023-12-01T04:00:00Z",
            "MinSize": 1,
            "MaxSize": 6,
            "DesiredCapacity": 4,
            "TimeZone": "America/New_York"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的排程擴展。

範例 4：描述具有指定開始時間的排程動作

若要描述在特定時間開始的排程動作，請使用 `--start-time`選項。

```
aws autoscaling describe-scheduled-actions \
    --start-time "2023-12-01T04:00:00Z"
```

輸出：

```
{
    "ScheduledUpdateGroupActions": [
        {
            "AutoScalingGroupName": "my-asg",
            "ScheduledActionName": "my-recurring-action",
            "Recurrence": "30 0 1 1,6,12 *",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
            "StartTime": "2023-12-01T04:00:00Z",
            "Time": "2023-12-01T04:00:00Z",
            "MinSize": 1,
            "MaxSize": 6,
            "DesiredCapacity": 4,
            "TimeZone": "America/New_York"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的排程擴展。

範例 5：描述在指定時間結束的排程動作

若要描述在特定時間結束的排程動作，請使用 `--end-time`選項。

```
aws autoscaling describe-scheduled-actions \
    --end-time "2023-12-01T04:00:00Z"
```

輸出：

```
{
    "ScheduledUpdateGroupActions": [
        {
            "AutoScalingGroupName": "my-asg",
            "ScheduledActionName": "my-recurring-action",
            "Recurrence": "30 0 1 1,6,12 *",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
            "StartTime": "2023-12-01T04:00:00Z",
            "Time": "2023-12-01T04:00:00Z",
            "MinSize": 1,
            "MaxSize": 6,
            "DesiredCapacity": 4,
            "TimeZone": "America/New_York"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的排程擴展。

範例 6：描述指定數量的排程動作

若要傳回特定數量的排程動作，請使用 `--max-items`選項。

```
aws autoscaling describe-scheduled-actions \
    --auto-scaling-group-name my-asg \
    --max-items 1
```

輸出：

```
{
    "ScheduledUpdateGroupActions": [
        {
            "AutoScalingGroupName": "my-asg",
```

```
            "ScheduledActionName": "my-recurring-action",
            "Recurrence": "30 0 1 1,6,12 *",
            "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
            "StartTime": "2023-12-01T04:00:00Z",
            "Time": "2023-12-01T04:00:00Z",
            "MinSize": 1,
            "MaxSize": 6,
            "DesiredCapacity": 4,
            "TimeZone": "America/New_York"
        }
    ]
}
```

如果輸出包含 NextToken 欄位，則有更多排程動作。若要取得其他排程動作，請在後續呼叫中使用此欄位的值搭配 --starting-token選項，如下所示。

```
aws autoscaling describe-scheduled-actions \
    --auto-scaling-group-name my-asg \
    --starting-token Z3M3LMPEXAMPLE
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的排程擴展。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeScheduledActions。

describe-tags

以下程式碼範例顯示如何使用 describe-tags。

AWS CLI

描述所有標籤

此範例說明您的所有標籤。

```
aws autoscaling describe-tags
```

輸出：

```
{
    "Tags": [
```

```
        {
            "ResourceType": "auto-scaling-group",
            "ResourceId": "my-asg",
            "PropagateAtLaunch": true,
            "Value": "Research",
            "Key": "Dept"
        },
        {
            "ResourceType": "auto-scaling-group",
            "ResourceId": "my-asg",
            "PropagateAtLaunch": true,
            "Value": "WebServer",
            "Key": "Role"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的標記 Auto Scaling 群組和執行個體。 Amazon EC2 Auto Scaling

範例 2：描述指定群組的標籤

若要描述特定 Auto Scaling 群組的標籤，請使用 `--filters`選項。

```
aws autoscaling describe-tags --filters Name=auto-scaling-group,Values=my-asg
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的標記 Auto Scaling 群組和執行個體。 Amazon EC2 Auto Scaling

範例 3：描述指定的標籤數量

若要傳回特定數量的標籤，請使用 `--max-items`選項。

```
aws autoscaling describe-tags \
    --max-items 1
```

如果輸出包含 NextToken 欄位，則有更多標籤。若要取得其他標籤，請在後續呼叫中使用此欄位的值與 `--starting-token`選項，如下所示。

```
aws autoscaling describe-tags \
    --filters Name=auto-scaling-group,Values=my-asg \
    --starting-token Z3M3LMPEXAMPLE
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的標記 Auto Scaling 群組和執行個體。 Amazon EC2 Auto Scaling

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTags。

`describe-termination-policy-types`

以下程式碼範例顯示如何使用 `describe-termination-policy-types`。

AWS CLI

描述可用的終止政策類型

此範例說明可用的終止政策類型。

```
aws autoscaling describe-termination-policy-types
```

輸出：

```
{
    "TerminationPolicyTypes": [
        "AllocationStrategy",
        "ClosestToNextInstanceHour",
        "Default",
        "NewestInstance",
        "OldestInstance",
        "OldestLaunchConfiguration",
        "OldestLaunchTemplate"
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的控制哪些 Auto Scaling 執行個體在縮減期間終止。 Amazon EC2 Auto Scaling

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTerminationPolicyTypes。

`describe-warm-pool`

以下程式碼範例顯示如何使用 `describe-warm-pool`。

AWS CLI

描述暖集區

此範例說明指定 Auto Scaling 群組的暖集區。

```
aws autoscaling describe-warm-pool \
    --auto-scaling-group-name my-asg
```

輸出：

```
{
    "WarmPoolConfiguration": {
        "MinSize": 2,
        "PoolState": "Stopped"
    },
    "Instances": [
        {
            "InstanceId": "i-070a5bbc7e7f40dc5",
            "InstanceType": "t2.micro",
            "AvailabilityZone": "us-west-2c",
            "LifecycleState": "Warmed:Pending",
            "HealthStatus": "Healthy",
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-00a731f6e9fa48610",
                "LaunchTemplateName": "my-template-for-auto-scaling",
                "Version": "6"
            }
        },
        {
            "InstanceId": "i-0b52f061814d3bd2d",
            "InstanceType": "t2.micro",
            "AvailabilityZone": "us-west-2b",
            "LifecycleState": "Warmed:Pending",
            "HealthStatus": "Healthy",
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-00a731f6e9fa48610",
                "LaunchTemplateName": "my-template-for-auto-scaling",
                "Version": "6"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Amazon EC2 Auto Scaling 的暖集區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeWarmPool。

detach-instances

以下程式碼範例顯示如何使用 detach-instances。

AWS CLI

從 Auto Scaling 群組分離執行個體

此範例會將指定的執行個體與指定的 Auto Scaling 群組分離。

```
aws autoscaling detach-instances \
    --instance-ids i-030017cfa84b20135 \
    --auto-scaling-group-name my-asg \
    --should-decrement-desired-capacity
```

輸出：

```
{
    "Activities": [
        {
            "ActivityId": "5091cb52-547a-47ce-a236-c9ccbc2cb2c9",
            "AutoScalingGroupName": "my-asg",
            "Description": "Detaching EC2 instance: i-030017cfa84b20135",
            "Cause": "At 2020-10-31T17:35:04Z instance i-030017cfa84b20135 was
 detached in response to a user request, shrinking the capacity from 2 to 1.",
            "StartTime": "2020-04-12T15:02:16.179Z",
            "StatusCode": "InProgress",
            "Progress": 50,
            "Details": "{\"Subnet ID\":\"subnet-6194ea3b\",\"Availability Zone\":
\"us-west-2c\"}"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetachInstances。

detach-load-balancer-target-groups

以下程式碼範例顯示如何使用 detach-load-balancer-target-groups。

AWS CLI

從 Auto Scaling 群組分離負載平衡器目標群組

此範例會將指定的負載平衡器目標群組與指定的 Auto Scaling 群組分離。

```
aws autoscaling detach-load-balancer-target-groups \
    --auto-scaling-group-name my-asg \
    --target-group-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

此命令不會產生輸出

如需詳細資訊，請參閱 Amazon EC2 Auto Scaling User Guide 中的 Attaching a load balancer to your Auto Scaling group。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetachLoadBalancerTargetGroups。

`detach-load-balancers`

以下程式碼範例顯示如何使用 `detach-load-balancers`。

AWS CLI

從 Auto Scaling 群組分離 Classic Load Balancer

此範例會將指定的 Classic Load Balancer 從指定的 Auto Scaling 群組分離。

```
aws autoscaling detach-load-balancers \
    --load-balancer-names my-load-balancer \
    --auto-scaling-group-name my-asg
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Amazon EC2 Auto Scaling User Guide 中的 Attaching a load balancer to your Auto Scaling group。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetachLoadBalancers。

`disable-metrics-collection`

以下程式碼範例顯示如何使用 `disable-metrics-collection`。

AWS CLI

停用 Auto Scaling 群組的指標集合

此範例會停用指定 Auto Scaling `GroupDesiredCapacity` 群組的指標集合。

```
aws autoscaling disable-metrics-collection \
    --auto-scaling-group-name my-asg \
    --metrics GroupDesiredCapacity
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的監控 Auto Scaling 群組和執行個體的 CloudWatch 指標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableMetricsCollection。

`enable-metrics-collection`

以下程式碼範例顯示如何使用 `enable-metrics-collection`。

AWS CLI

範例 1：啟用 Auto Scaling 群組的指標集合

此範例會啟用指定 Auto Scaling 群組的資料收集。

```
aws autoscaling enable-metrics-collection \
    --auto-scaling-group-name my-asg \
    --granularity "1Minute"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的監控 Auto Scaling 群組和執行個體的 CloudWatch 指標。

範例 2：收集 Auto Scaling 群組指定指標的資料

若要收集特定指標的資料，請使用 `--metrics`選項。

```
aws autoscaling enable-metrics-collection \
    --auto-scaling-group-name my-asg \
```

```
    --metrics GroupDesiredCapacity --granularity "1Minute"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的監控 Auto Scaling 群組和執行個體的 CloudWatch 指標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableMetricsCollection。

enter-standby

以下程式碼範例顯示如何使用 enter-standby。

AWS CLI

將執行個體移至待命模式

此範例會將指定的執行個體置於待命模式。這對於更新或故障診斷目前服務中的執行個體非常有用。

```
aws autoscaling enter-standby \
    --instance-ids i-061c63c5eb45f0416 \
    --auto-scaling-group-name my-asg \
    --should-decrement-desired-capacity
```

輸出：

```
{
    "Activities": [
        {
            "ActivityId": "ffa056b4-6ed3-41ba-ae7c-249dfae6eba1",
            "AutoScalingGroupName": "my-asg",
            "Description": "Moving EC2 instance to Standby: i-061c63c5eb45f0416",
            "Cause": "At 2020-10-31T20:31:00Z instance i-061c63c5eb45f0416 was moved
 to standby in response to a user request, shrinking the capacity from 1 to 0.",
            "StartTime": "2020-10-31T20:31:00.949Z",
            "StatusCode": "InProgress",
            "Progress": 50,
            "Details": "{\"Subnet ID\":\"subnet-6194ea3b\",\"Availability Zone\":
\"us-west-2c\"}"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Amazon EC2 Auto Scaling 執行個體生命週期。 Amazon EC2 Auto Scaling

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnterStandby。

`execute-policy`

以下程式碼範例顯示如何使用 `execute-policy`。

AWS CLI

執行擴展政策

此範例會針對指定的 Auto Scaling 群組執行名為 `my-step-scale-out-policy`的擴展政策。

```
aws autoscaling execute-policy \
    --auto-scaling-group-name my-asg \
    --policy-name  my-step-scale-out-policy \
    --metric-value 95 \
    --breach-threshold 80
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的步驟和簡單擴展政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ExecutePolicy。

`exit-standby`

以下程式碼範例顯示如何使用 `exit-standby`。

AWS CLI

將執行個體移出待命模式

此範例會將指定的執行個體移出待命模式。

```
aws autoscaling exit-standby \
    --instance-ids i-061c63c5eb45f0416 \
    --auto-scaling-group-name my-asg
```

輸出：

```
{
    "Activities": [
        {
            "ActivityId": "142928e1-a2dc-453a-9b24-b85ad6735928",
            "AutoScalingGroupName": "my-asg",
            "Description": "Moving EC2 instance out of Standby:
 i-061c63c5eb45f0416",
            "Cause": "At 2020-10-31T20:32:50Z instance i-061c63c5eb45f0416 was moved
 out of standby in response to a user request, increasing the capacity from 0 to
 1.",
            "StartTime": "2020-10-31T20:32:50.222Z",
            "StatusCode": "PreInService",
            "Progress": 30,
            "Details": "{\"Subnet ID\":\"subnet-6194ea3b\",\"Availability Zone\":
\"us-west-2c\"}"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的暫時從 Auto Scaling 群組移除執行個體。 Amazon EC2 Auto Scaling

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ExitStandby。

put-lifecycle-hook

以下程式碼範例顯示如何使用 put-lifecycle-hook。

AWS CLI

範例 1：建立生命週期關聯

此範例會建立生命週期關聯，在任何新啟動的執行個體上調用，逾時為 4800 秒。這有助於將執行個體保持在等待狀態，直到使用者資料指令碼完成，或使用 EventBridge 叫用 AWS Lambda 函數。

```
aws autoscaling put-lifecycle-hook \
    --auto-scaling-group-name my-asg \
    --lifecycle-hook-name my-launch-hook \
    --lifecycle-transition autoscaling:EC2_INSTANCE_LAUNCHING \
```

```
    --heartbeat-timeout 4800
```

此命令不會產生輸出。如果具有相同名稱的生命週期掛鉤已存在，則會被新的生命週期掛鉤覆寫。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Amazon EC2 Auto Scaling lifecycle hook。

範例 2：傳送 Amazon SNS 電子郵件訊息，通知您執行個體狀態轉換

此範例會建立具有 Amazon SNS 主題和 IAM 角色的生命週期掛鉤，用於在執行個體啟動時接收通知。

```
aws autoscaling put-lifecycle-hook \
    --auto-scaling-group-name my-asg \
    --lifecycle-hook-name my-launch-hook \
    --lifecycle-transition autoscaling:EC2_INSTANCE_LAUNCHING \
    --notification-target-arn arn:aws:sns:us-west-2:123456789012:my-sns-topic \
    --role-arn arn:aws:iam::123456789012:role/my-auto-scaling-role
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Amazon EC2 Auto Scaling lifecycle hook。

範例 3：發佈訊息至 Amazon SQS 佇列

此範例會建立生命週期掛鉤，將具有中繼資料的訊息發佈至指定的 Amazon SQS 佇列。

```
aws autoscaling put-lifecycle-hook \
    --auto-scaling-group-name my-asg \
    --lifecycle-hook-name my-launch-hook \
    --lifecycle-transition autoscaling:EC2_INSTANCE_LAUNCHING \
    --notification-target-arn arn:aws:sqs:us-west-2:123456789012:my-sqs-queue \
    --role-arn arn:aws:iam::123456789012:role/my-notification-role \
    --notification-metadata "SQS message metadata"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Amazon EC2 Auto Scaling lifecycle hook。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutLifecycleHook。

put-notification-configuration

以下程式碼範例顯示如何使用 put-notification-configuration。

AWS CLI

新增通知

此範例會將指定的通知新增至指定的 Auto Scaling 群組。

```
aws autoscaling put-notification-configuration \
    --auto-scaling-group-name my-asg \
    --topic-arn arn:aws:sns:us-west-2:123456789012:my-sns-topic \
    --notification-type autoscaling:TEST_NOTIFICATION
```

此命令不會產生輸出。

有關詳細資訊，請參閱 Amazon EC2 Auto Scaling 使用者指南中的取得 Auto Scaling 群組擴展時的 Amazon SNS 通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutNotificationConfiguration。

put-scaling-policy

以下程式碼範例顯示如何使用 put-scaling-policy。

AWS CLI

將目標追蹤擴展政策新增至 Auto Scaling 群組

下列put-scaling-policy範例會將目標追蹤擴展政策套用至指定的 Auto Scaling 群組。輸出包含代表您建立的兩個 CloudWatch 警示的 ARNs 和名稱。如果具有相同名稱的擴展政策已存在，則新的擴展政策會覆寫該政策。

```
aws autoscaling put-scaling-policy --auto-scaling-group-name my-asg \
  --policy-name alb1000-target-tracking-scaling-policy \
  --policy-type TargetTrackingScaling \
  --target-tracking-configuration file://config.json
```

config.json 的內容：

```
{
```

```
    "TargetValue": 1000.0,
    "PredefinedMetricSpecification": {
         "PredefinedMetricType": "ALBRequestCountPerTarget",
         "ResourceLabel": "app/my-alb/778d41231b141a0f/targetgroup/my-alb-target-
group/943f017f100becff"
    }
}
```

輸出：

```
{
    "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:228f02c2-
c665-4bfd-aaac-8b04080bea3c:autoScalingGroupName/my-asg:policyName/alb1000-target-
tracking-scaling-policy",
    "Alarms": [
        {
            "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-
my-asg-AlarmHigh-fc0e4183-23ac-497e-9992-691c9980c38e",
            "AlarmName": "TargetTracking-my-asg-AlarmHigh-
fc0e4183-23ac-497e-9992-691c9980c38e"
        },
        {
            "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-
my-asg-AlarmLow-61a39305-ed0c-47af-bd9e-471a352ee1a2",
            "AlarmName": "TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2"
        }
    ]
 }
```

如需更多範例，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 AWS 命令列界面 (AWS CLI) 的範例擴展政策。 Amazon EC2 Auto Scaling

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutScalingPolicy。

put-scheduled-update-group-action

以下程式碼範例顯示如何使用 put-scheduled-update-group-action。

AWS CLI

範例 1：將排程動作新增至 Auto Scaling 群組

此範例會將指定的排程動作新增至指定的 Auto Scaling 群組。

```
aws autoscaling put-scheduled-update-group-action \
    --auto-scaling-group-name my-asg \
    --scheduled-action-name my-scheduled-action \
    --start-time "2023-05-12T08:00:00Z" \
    --min-size 2 \
    --max-size 6 \
    --desired-capacity 4
```

此命令不會產生輸出。如果具有相同名稱的排程動作已存在，則會被新的排程動作覆寫。

如需更多範例，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的排程擴展。

範例 2：指定週期性排程

此範例會建立排程動作，以根據排程於每年 1 月、6 月和 12 月 1 日 00：30 執行的週期性排程進行擴展。

```
aws autoscaling put-scheduled-update-group-action \
    --auto-scaling-group-name my-asg \
    --scheduled-action-name my-recurring-action \
    --recurrence "30 0 1 1,6,12 *" \
    --min-size 2 \
    --max-size 6 \
    --desired-capacity 4
```

此命令不會產生輸出。如果具有相同名稱的排程動作已存在，則會被新的排程動作覆寫。

如需更多範例，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的排程擴展。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutScheduledUpdateGroupAction。

put-warm-pool

以下程式碼範例顯示如何使用 `put-warm-pool`。

AWS CLI

建立暖集區

下列範例會為指定的 Auto Scaling 群組建立暖集區。

```
aws autoscaling put-warm-pool \
    --auto-scaling-group-name my-asg \
    --min-size 2
```

此命令不會產生輸出。如果暖集區已存在，將會更新。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Amazon EC2 Auto Scaling 的暖集區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutWarmPool。

record-lifecycle-action-heartbeat

以下程式碼範例顯示如何使用 `record-lifecycle-action-heartbeat`。

AWS CLI

記錄生命週期動作活動訊號

此範例會記錄生命週期動作活動訊號，讓執行個體保持待定狀態。

```
aws autoscaling record-lifecycle-action-heartbeat \
    --lifecycle-hook-name my-launch-hook \
    --auto-scaling-group-name my-asg \
    --lifecycle-action-token bcd2f1b8-9a78-44d3-8a7a-4dd07d7cf635
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Amazon EC2 Auto Scaling lifecycle hook。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RecordLifecycleActionHeartbeat。

resume-processes

以下程式碼範例顯示如何使用 `resume-processes`。

AWS CLI

若要繼續暫停的程序

此範例會繼續指定 Auto Scaling 群組的指定暫停擴展程序。

```
aws autoscaling resume-processes \
    --auto-scaling-group-name my-asg \
    --scaling-processes AlarmNotification
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的暫停和繼續擴展程序。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResumeProcesses。

rollback-instance-refresh

以下程式碼範例顯示如何使用 rollback-instance-refresh。

AWS CLI

復原執行個體重新整理

下列rollback-instance-refresh範例會復原指定 Auto Scaling 群組的進行中執行個體重新整理。

```
aws autoscaling rollback-instance-refresh \
    --auto-scaling-group-name my-asg
```

輸出：

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的使用復原功能來復原變更。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RollbackInstanceRefresh。

set-desired-capacity

以下程式碼範例顯示如何使用 set-desired-capacity。

AWS CLI

設定 Auto Scaling 群組所需的容量

此範例會為指定的 Auto Scaling 群組設定所需的容量。

```
aws autoscaling set-desired-capacity \
    --auto-scaling-group-name my-asg \
    --desired-capacity 2 \
    --honor-cooldown
```

如果成功，此命令會回到提示字元。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetDesiredCapacity。

set-instance-health

以下程式碼範例顯示如何使用 set-instance-health。

AWS CLI

設定執行個體的運作狀態

此範例會將指定執行個體的運作狀態設定為 Unhealthy。

```
aws autoscaling set-instance-health \
    --instance-id i-061c63c5eb45f0416 \
    --health-status Unhealthy
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetInstanceHealth。

set-instance-protection

以下程式碼範例顯示如何使用 set-instance-protection。

AWS CLI

範例 1：啟用執行個體的執行個體保護設定

此範例會啟用指定執行個體的執行個體保護。

```
aws autoscaling set-instance-protection \
```

```
    --instance-ids i-061c63c5eb45f0416 \
    --auto-scaling-group-name my-asg --protected-from-scale-in
```

此命令不會產生輸出。

範例 2：停用執行個體的執行個體保護設定

此範例會停用指定執行個體的執行個體保護。

```
aws autoscaling set-instance-protection \
    --instance-ids i-061c63c5eb45f0416 \
    --auto-scaling-group-name my-asg \
    --no-protected-from-scale-in
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetInstanceProtection。

start-instance-refresh

以下程式碼範例顯示如何使用 start-instance-refresh。

AWS CLI

範例 1：使用命令列參數啟動執行個體重新整理

下列start-instance-refresh範例使用命令列引數啟動執行個體重新整理。選用preferences參數會指定 60 秒InstanceWarmup的 和50百分比MinHealthyPercentage的 。

```
aws autoscaling start-instance-refresh \
    --auto-scaling-group-name my-asg \
    --preferences '{"InstanceWarmup": 60, "MinHealthyPercentage": 50}'
```

輸出：

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的啟動執行個體重新整理。

範例 2：使用 JSON 檔案啟動執行個體重新整理

下列start-instance-refresh範例使用 JSON 檔案啟動執行個體重新整理。您可以指定 Auto Scaling 群組，並在 JSON 檔案中定義所需的組態和偏好設定，如下列範例所示。

```
aws autoscaling start-instance-refresh \
    --cli-input-json file://config.json
```

config.json 的內容：

```
{
    "AutoScalingGroupName": "my-asg",
    "DesiredConfiguration": {
        "LaunchTemplate": {
            "LaunchTemplateId": "lt-068f72b729example",
            "Version": "$Default"
        }
    },
    "Preferences": {
        "InstanceWarmup": 60,
        "MinHealthyPercentage": 50,
        "AutoRollback": true,
        "ScaleInProtectedInstances": Ignore,
        "StandbyInstances": Terminate
    }
}
```

輸出：

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的啟動執行個體重新整理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartInstanceRefresh。

suspend-processes

以下程式碼範例顯示如何使用 suspend-processes。

AWS CLI

暫停 Auto Scaling 程序

此範例會暫停指定 Auto Scaling 群組的指定擴展程序。

```
aws autoscaling suspend-processes \
    --auto-scaling-group-name my-asg \
    --scaling-processes AlarmNotification
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的暫停和繼續擴展程序。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SuspendProcesses。

terminate-instance-in-auto-scaling-group

以下程式碼範例顯示如何使用 terminate-instance-in-auto-scaling-group。

AWS CLI

終止 Auto Scaling 群組中的執行個體

此範例會從指定的 Auto Scaling 群組終止指定的執行個體，而不更新群組的大小。Amazon EC2 Auto Scaling 會在指定的執行個體終止後啟動替代執行個體。

```
aws autoscaling terminate-instance-in-auto-scaling-group \
    --instance-id i-061c63c5eb45f0416 \
    --no-should-decrement-desired-capacity
```

輸出：

```
{
    "Activities": [
        {
            "ActivityId": "8c35d601-793c-400c-fcd0-f64a27530df7",
            "AutoScalingGroupName": "my-asg",
            "Description": "Terminating EC2 instance: i-061c63c5eb45f0416",
            "Cause": "",
            "StartTime": "2020-10-31T20:34:25.680Z",
```

```
            "StatusCode": "InProgress",
            "Progress": 0,
            "Details": "{\"Subnet ID\":\"subnet-6194ea3b\",\"Availability Zone\":
\"us-west-2c\"}"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TerminateInstanceInAutoScalingGroup。

update-auto-scaling-group

以下程式碼範例顯示如何使用 update-auto-scaling-group。

AWS CLI

範例 1：更新 Auto Scaling 群組的大小限制

此範例會更新指定的 Auto Scaling 群組，最小大小為 2，最大大小為 10。

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --min-size 2 \
    --max-size 10
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的設定 Auto Scaling 群組的容量限制。 Amazon EC2 Auto Scaling

範例 2：新增 Elastic Load Balancing 運作狀態檢查，並指定要使用的可用區域和子網路

此範例會更新指定的 Auto Scaling 群組，以新增 Elastic Load Balancing 運作狀態檢查。此命令也會--vpc-zone-identifier使用多個可用區域中的子網路 IDs 清單來更新 的值。

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --health-check-type ELB \
    --health-check-grace-period 600 \
    --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Elastic Load Balancing 和 Amazon EC2 Auto Scaling。 Amazon EC2 Auto Scaling

範例 3：更新置放群組和終止政策

此範例會更新要使用的置放群組和終止政策。

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --placement-group my-placement-group \
    --termination-policies "OldestInstance"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Auto Scaling 群組。

範例 4：使用最新版本的啟動範本

此範例會更新指定的 Auto Scaling 群組，以使用指定啟動範本的最新版本。

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateId=lt-1234567890abcde12,Version='$Latest'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Amazon EC2 Auto Scaling 使用者指南中的啟動範本。

範例 5：使用特定版本的啟動範本

此範例會更新指定的 Auto Scaling 群組，以使用啟動範本的特定版本，而非最新或預設版本。

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='2'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Amazon EC2 Auto Scaling 使用者指南中的啟動範本。

範例 6：定義混合執行個體政策並啟用容量重新平衡

此範例會更新指定的 Auto Scaling 群組，以使用混合執行個體政策並啟用容量重新平衡。此結構可讓您指定具有 Spot 和隨需容量的群組，並針對不同的架構使用不同的啟動範本。

```
aws autoscaling update-auto-scaling-group \
    --cli-input-json file://~/config.json
```

config.json 的內容：

```
{
    "AutoScalingGroupName": "my-asg",
    "CapacityRebalance": true,
    "MixedInstancesPolicy": {
        "LaunchTemplate": {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template-for-x86",
                "Version": "$Latest"
            },
            "Overrides": [
                {
                    "InstanceType": "c6g.large",
                    "LaunchTemplateSpecification": {
                        "LaunchTemplateName": "my-launch-template-for-arm",
                        "Version": "$Latest"
                    }
                },
                {
                    "InstanceType": "c5.large"
                },
                {
                    "InstanceType": "c5a.large"
                }
            ]
        },
        "InstancesDistribution": {
            "OnDemandPercentageAboveBaseCapacity": 50,
            "SpotAllocationStrategy": "capacity-optimized"
        }
    }
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱「Amazon EC2 Auto Scaling 使用者指南」中的具備多個執行個體類型及購買選項的 Auto Scaling 群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAutoScalingGroup。

使用 的 Auto Scaling Plans 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Auto Scaling Plans 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-scaling-plan`

以下程式碼範例顯示如何使用 `create-scaling-plan`。

AWS CLI

建立擴展計畫

下列`create-scaling-plan`範例`my-scaling-plan`使用已建立的 JSON 檔案 （ 名為 config.json) 建立名為 的擴展計劃。擴展計畫的結構包含名為 之 Auto Scaling 群組的擴展指示`my-asg`。它會將 `TagFilters` 屬性指定為應用程式來源，並啟用預測擴展和動態擴展。

```
aws autoscaling-plans create-scaling-plan \
    --scaling-plan-name my-scaling-plan \
    --cli-input-json file://~/config.json
```

`config.json` 檔案的內容：

```
{
    "ApplicationSource": {
        "TagFilters": [
```

```
            {
                "Key": "purpose",
                "Values": [
                    "my-application"
                ]
            }
        ]
    },
    "ScalingInstructions": [
        {
            "ServiceNamespace": "autoscaling",
            "ResourceId": "autoScalingGroup/my-asg",
            "ScalableDimension": "autoscaling:autoScalingGroup:DesiredCapacity",
            "ScheduledActionBufferTime": 300,
            "PredictiveScalingMaxCapacityBehavior":
 "SetForecastCapacityToMaxCapacity",
            "PredictiveScalingMode": "ForecastAndScale",
            "PredefinedLoadMetricSpecification": {
                "PredefinedLoadMetricType": "ASGTotalCPUUtilization"
            },
            "ScalingPolicyUpdateBehavior": "ReplaceExternalPolicies",
            "MinCapacity": 1,
            "MaxCapacity": 4,
            "TargetTrackingConfigurations": [
                {
                    "PredefinedScalingMetricSpecification": {
                        "PredefinedScalingMetricType": "ASGAverageCPUUtilization"
                    },
                    "TargetValue": 50
                }
            ]
        }
    ]
}
```

輸出：

```
{
"ScalingPlanVersion": 1
}
```

如需詳細資訊，請參閱 AWS Auto Scaling 使用者指南。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateScalingPlan。

delete-scaling-plan

以下程式碼範例顯示如何使用 delete-scaling-plan。

AWS CLI

刪除擴展計劃

下列delete-scaling-plan範例會刪除指定的擴展計劃。

```
aws autoscaling-plans delete-scaling-plan \
    --scaling-plan-name my-scaling-plan \
    --scaling-plan-version 1
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Auto Scaling 使用者指南。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteScalingPlan。

describe-scaling-plan-resources

以下程式碼範例顯示如何使用 describe-scaling-plan-resources。

AWS CLI

描述擴展計劃的可擴展資源

下列describe-scaling-plan-resources範例顯示與指定擴展計畫相關聯的單一可擴展資源 (Auto Scaling 群組) 的詳細資訊。

```
aws autoscaling-plans describe-scaling-plan-resources \
    --scaling-plan-name my-scaling-plan \
    --scaling-plan-version 1
```

輸出：

```
{
    "ScalingPlanResources": [
        {
            "ScalableDimension": "autoscaling:autoScalingGroup:DesiredCapacity",
            "ScalingPlanVersion": 1,
```

```
            "ResourceId": "autoScalingGroup/my-asg",
            "ScalingStatusCode": "Active",
            "ScalingStatusMessage": "Target tracking scaling policies have been
 applied to the resource.",
            "ScalingPolicies": [
                {
                    "PolicyName": "AutoScaling-my-asg-b1ab65ae-4be3-4634-bd64-
c7471662b251",
                    "PolicyType": "TargetTrackingScaling",
                    "TargetTrackingConfiguration": {
                        "PredefinedScalingMetricSpecification": {
                            "PredefinedScalingMetricType":
 "ALBRequestCountPerTarget",
                            "ResourceLabel": "app/my-alb/f37c06a68c1748aa/
targetgroup/my-target-group/6d4ea56ca2d6a18d"
                        },
                        "TargetValue": 40.0
                    }
                }
            ],
            "ServiceNamespace": "autoscaling",
            "ScalingPlanName": "my-scaling-plan"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Auto Scaling 使用者指南中的什麼是 Auto Scaling？。 AWS Auto Scaling

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeScalingPlanResources。

describe-scaling-plans

以下程式碼範例顯示如何使用 describe-scaling-plans。

AWS CLI

描述擴展計劃

下列describe-scaling-plans範例顯示指定擴展計劃的詳細資訊。

```
aws autoscaling-plans describe-scaling-plans \
    --scaling-plan-names scaling-plan-with-asg-and-ddb
```

輸出：

```
{
    "ScalingPlans": [
        {
            "LastMutatingRequestTime": 1565388443.963,
            "ScalingPlanVersion": 1,
            "CreationTime": 1565388443.963,
            "ScalingInstructions": [
                {
                    "ScalingPolicyUpdateBehavior": "ReplaceExternalPolicies",
                    "ScalableDimension":
 "autoscaling:autoScalingGroup:DesiredCapacity",
                    "TargetTrackingConfigurations": [
                        {
                            "PredefinedScalingMetricSpecification": {
                                "PredefinedScalingMetricType":
 "ASGAverageCPUUtilization"
                            },
                            "TargetValue": 50.0,
                            "EstimatedInstanceWarmup": 300,
                            "DisableScaleIn": false
                        }
                    ],
                    "ResourceId": "autoScalingGroup/my-asg",
                    "DisableDynamicScaling": false,
                    "MinCapacity": 1,
                    "ServiceNamespace": "autoscaling",
                    "MaxCapacity": 10
                },
                {
                    "ScalingPolicyUpdateBehavior": "ReplaceExternalPolicies",
                    "ScalableDimension": "dynamodb:table:ReadCapacityUnits",
                    "TargetTrackingConfigurations": [
                        {
                            "PredefinedScalingMetricSpecification": {
                                "PredefinedScalingMetricType":
 "DynamoDBReadCapacityUtilization"
                            },
                            "TargetValue": 50.0,
                            "ScaleInCooldown": 60,
                            "DisableScaleIn": false,
                            "ScaleOutCooldown": 60
                        }
```

```
                    ],
                    "ResourceId": "table/my-table",
                    "DisableDynamicScaling": false,
                    "MinCapacity": 5,
                    "ServiceNamespace": "dynamodb",
                    "MaxCapacity": 10000
                },
                {
                    "ScalingPolicyUpdateBehavior": "ReplaceExternalPolicies",
                    "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
                    "TargetTrackingConfigurations": [
                        {
                            "PredefinedScalingMetricSpecification": {
                                "PredefinedScalingMetricType":
"DynamoDBWriteCapacityUtilization"
                            },
                            "TargetValue": 50.0,
                            "ScaleInCooldown": 60,
                            "DisableScaleIn": false,
                            "ScaleOutCooldown": 60
                        }
                    ],
                    "ResourceId": "table/my-table",
                    "DisableDynamicScaling": false,
                    "MinCapacity": 5,
                    "ServiceNamespace": "dynamodb",
                    "MaxCapacity": 10000
                }
            ],
            "ApplicationSource": {
                "TagFilters": [
                    {
                        "Values": [
                            "my-application-id"
                        ],
                        "Key": "application"
                    }
                ]
            },
            "StatusStartTime": 1565388455.836,
            "ScalingPlanName": "scaling-plan-with-asg-and-ddb",
            "StatusMessage": "Scaling plan has been created and applied to all
resources.",
            "StatusCode": "Active"
```

```
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Auto Scaling 使用者指南中的什麼是 Auto Scaling？。 AWS Auto Scaling

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeScalingPlans。

get-scaling-plan-resource-forecast-data

以下程式碼範例顯示如何使用 get-scaling-plan-resource-forecast-data。

AWS CLI

擷取負載預測資料

此範例會擷取與指定擴展計畫相關聯之可擴展資源 (Auto Scaling 群組） 的負載預測資料。

```
aws autoscaling-plans get-scaling-plan-resource-forecast-data \
    --scaling-plan-name my-scaling-plan \
    --scaling-plan-version 1 \
    --service-namespace "autoscaling" \
    --resource-id autoScalingGroup/my-asg \
    --scalable-dimension "autoscaling:autoScalingGroup:DesiredCapacity" \
    --forecast-data-type "LoadForecast" \
    --start-time "2019-08-30T00:00:00Z" \
    --end-time "2019-09-06T00:00:00Z"
```

輸出：

```
{
    "Datapoints": [...]
}
```

如需詳細資訊，請參閱 AWS Auto Scaling 使用者指南中的什麼是 AWS Auto Scaling。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetScalingPlanResourceForecastData。

update-scaling-plan

以下程式碼範例顯示如何使用 update-scaling-plan。

AWS CLI

更新擴展計畫

下列update-scaling-plan範例會修改指定擴展計畫中 Auto Scaling 群組的擴展指標。

```
aws autoscaling-plans update-scaling-plan \
    --scaling-plan-name my-scaling-plan \
    --scaling-plan-version 1 \
    --scaling-instructions
 '{"ScalableDimension":"autoscaling:autoScalingGroup:DesiredCapacity","ResourceId":"autoScal
my-asg","ServiceNamespace":"autoscaling","TargetTrackingConfigurations":
[{"PredefinedScalingMetricSpecification":
 {"PredefinedScalingMetricType":"ALBRequestCountPerTarget","ResourceLabel":"app/my-
alb/f37c06a68c1748aa/targetgroup/my-target-
group/6d4ea56ca2d6a18d"},"TargetValue":40.0}],"MinCapacity": 1,"MaxCapacity": 10}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Auto Scaling 使用者指南中的什麼是自動擴展？。 AWS Auto Scaling

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateScalingPlan。

AWS Backup 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS Backup。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-backup-plan`

以下程式碼範例顯示如何使用 `create-backup-plan`。

AWS CLI

建立備份計劃

下列`create-backup-plan`範例會建立具有 35 天保留的指定備份計劃。

```
aws backup create-backup-plan \
--backup-plan "{\"BackupPlanName\":\"Example-Backup-Plan\",\"Rules\":[{\"RuleName\":
\"DailyBackups\",\"ScheduleExpression\":\"cron(0 5 ? * * *)\",\"StartWindowMinutes
\":480,\"TargetBackupVaultName\":\"Default\",\"Lifecycle\":{\"DeleteAfterDays
\":35}}]}"
```

輸出：

```
{
    "BackupPlanId": "1fa3895c-a7f5-484a-a371-2dd6a1a9f729",
    "BackupPlanArn": "arn:aws:backup:us-west-2:123456789012:backup-plan:1fa3895c-
a7f5-484a-a371-2dd6a1a9f729",
    "CreationDate": 1568928754.747,
    "VersionId": "ZjQ2ZTI5YWQtZDg5Yi00MzYzLWJmZTAtMDI1MzhlMDhjYjEz"
}
```

如需詳細資訊，請參閱《 Backup 開發人員指南》中的建立備份計劃。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateBackupPlan。

`create-backup-vault`

以下程式碼範例顯示如何使用 `create-backup-vault`。

AWS CLI

建立備份保存庫

下列`create-backup-vault`範例會建立具有指定名稱的備份保存庫。

```
aws backup create-backup-vault
    --backup-vault-name sample-vault
```

此命令不會產生輸出。輸出：

```
{
    "BackupVaultName": "sample-vault",
    "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-vault:sample-
vault",
    "CreationDate": 1568928338.385
}
```

如需詳細資訊，請參閱《 Backup 開發人員指南》中的建立備份保存庫。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateBackupVault。

get-backup-plan-from-template

以下程式碼範例顯示如何使用 get-backup-plan-from-template。

AWS CLI

從範本取得現有的備份計劃

下列get-backup-plan-from-template範例會從範本取得現有的備份計劃，該範本會指定保留 35 天的每日備份。

```
aws backup get-backup-plan-from-template \
    --backup-plan-template-id "87c0c1ef-254d-4180-8fef-2e76a2c38aaa"
```

輸出：

```
{
    "BackupPlanDocument": {
        "Rules": [
            {
                "RuleName": "DailyBackups",
                "ScheduleExpression": "cron(0 5 ? * * *)",
                "StartWindowMinutes": 480,
                "Lifecycle": {
```

```
                    "DeleteAfterDays": 35
                }
            }
        ]
    }
}
```

如需詳細資訊，請參閱《 Backup 開發人員指南》中的建立備份計劃。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBackupPlanFromTemplate。

get-backup-plan

以下程式碼範例顯示如何使用 get-backup-plan。

AWS CLI

取得備份計劃的詳細資訊

下列get-backup-plan範例顯示指定備份計劃的詳細資訊。

```
aws backup get-backup-plan \
    --backup-plan-id "fcbf5d8f-bd77-4f3a-9c97-f24fb3d373a5"
```

輸出：

```
{
    "BackupPlan": {
        "BackupPlanName": "Example-Backup-Plan",
        "Rules": [
            {
                "RuleName": "DailyBackups",
                "TargetBackupVaultName": "Default",
                "ScheduleExpression": "cron(0 5 ? * * *)",
                "StartWindowMinutes": 480,
                "CompletionWindowMinutes": 10080,
                "Lifecycle": {
                    "DeleteAfterDays": 35
                },
                "RuleId": "70e0ccdc-e9df-4e83-82ad-c1e5a9471cc3"
            }
        ]
```

```
    },
    "BackupPlanId": "fcbf5d8f-bd77-4f3a-9c97-f24fb3d373a5",
    "BackupPlanArn": "arn:aws:backup:us-west-2:123456789012:backup-plan:fcbf5d8f-
bd77-4f3a-9c97-f24fb3d373a5",
    "VersionId": "NjQ2ZTZkODktMGVhNy00MmQ0LWE4YjktZTkwNTQ3OTkyYTcw",
    "CreationDate": 1568926091.57
}
```

如需詳細資訊，請參閱《 Backup 開發人員指南》中的建立備份計劃。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBackupPlan。

list-backup-jobs

以下程式碼範例顯示如何使用 list-backup-jobs。

AWS CLI

範例 1：列出所有備份任務

下列list-backup-jobs範例會傳回您 AWS 帳戶中備份任務的相關中繼資料。

```
aws backup list-backup-jobs
```

輸出：

```
{
    "BackupJobs": [
        {
            "BackupJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "BackupVaultName": "Default",
            "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-
vault:Default",
            "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-12345678901234567",
            "CreationDate": 1600721892.929,
            "State": "CREATED",
            "PercentDone": "0.0",
            "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/
AWSBackupDefaultServiceRole",
            "StartBy": 1600725492.929,
            "ResourceType": "EC2"
```

```
        },
        {
            "BackupJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "BackupVaultName": "Default",
            "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-
vault:Default",
            "RecoveryPointArn": "arn:aws:backup:us-west-2:123456789012:recovery-
point:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "ResourceArn": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-
system/fs-12345678",
            "CreationDate": 1600721724.77,
            "CompletionDate": 1600721744.488,
            "State": "COMPLETED",
            "PercentDone": "100.0",
            "BackupSizeInBytes": 71,
            "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/
AWSBackupDefaultServiceRole",
            "StartBy": 1600725324.77,
            "ResourceType": "EFS"
        }
    ]
}
```

如需詳細資訊，請參閱 備份開發人員指南中的建立AWS 備份。

範例 2：列出已完成的備份任務

下列list-backup-jobs範例會傳回您 AWS 帳戶中已完成備份任務的中繼資料。

```
aws backup list-backup-jobs \
    --by-state COMPLETED
```

輸出：

```
{
    "BackupJobs": [
        {
            "BackupJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "BackupVaultName": "Default",
            "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-
vault:Default",
            "RecoveryPointArn": "arn:aws:backup:us-west-2:123456789012:recovery-
point:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
```

```
            "ResourceArn": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-
system/fs-12345678",
            "CreationDate": 1600721724.77,
            "CompletionDate": 1600721744.488,
            "State": "COMPLETED",
            "PercentDone": "100.0",
            "BackupSizeInBytes": 71,
            "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/
AWSBackupDefaultServiceRole",
            "StartBy": 1600725324.77,
            "ResourceType": "EFS"
        }
    ]
}
```

如需詳細資訊，請參閱 備份開發人員指南中的建立AWS 備份。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBackupJobs。

AWS Batch 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS Batch。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`cancel-job`

以下程式碼範例顯示如何使用 `cancel-job`。

AWS CLI

取消任務

此範例會取消具有指定任務 ID 的任務。

命令：

```
aws batch cancel-job --job-id bcf0b186-a532-4122-842e-2ccab8d54efb --
reason "Cancelling job."
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelJob。

create-compute-environment

以下程式碼範例顯示如何使用 create-compute-environment。

AWS CLI

使用隨需執行個體建立受管運算環境

此範例會建立受管運算環境，其中包含隨需啟動的特定 C4 執行個體類型。運算環境稱為 C4OnDemand。

命令：

```
aws batch create-compute-environment --cli-input-json file://<path_to_json_file>/
C4OnDemand.json
```

JSON 檔案格式：

```
{
  "computeEnvironmentName": "C4OnDemand",
  "type": "MANAGED",
  "state": "ENABLED",
  "computeResources": {
    "type": "EC2",
    "minvCpus": 0,
    "maxvCpus": 128,
    "desiredvCpus": 48,
    "instanceTypes": [
      "c4.large",
      "c4.xlarge",
      "c4.2xlarge",
      "c4.4xlarge",
```

```
      "c4.8xlarge"
    ],
    "subnets": [
      "subnet-220c0e0a",
      "subnet-1a95556d",
      "subnet-978f6dce"
    ],
    "securityGroupIds": [
      "sg-cf5093b2"
    ],
    "ec2KeyPair": "id_rsa",
    "instanceRole": "ecsInstanceRole",
    "tags": {
      "Name": "Batch Instance - C4OnDemand"
    }
  },
  "serviceRole": "arn:aws:iam::012345678910:role/AWSBatchServiceRole"
}
```

輸出：

```
{
    "computeEnvironmentName": "C4OnDemand",
    "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/C4OnDemand"
}
```

使用 Spot 執行個體建立受管運算環境

此範例會使用當 Spot 出價等於或低於執行個體類型隨需價格的 20% 時啟動的 M4 執行個體類型來建立受管運算環境。運算環境稱為 M4Spot。

命令：

```
aws batch create-compute-environment --cli-input-json file://<path_to_json_file>/
M4Spot.json
```

JSON 檔案格式：

```
{
  "computeEnvironmentName": "M4Spot",
```

```
  "type": "MANAGED",
  "state": "ENABLED",
  "computeResources": {
    "type": "SPOT",
    "spotIamFleetRole": "arn:aws:iam::012345678910:role/aws-ec2-spot-fleet-role",
    "minvCpus": 0,
    "maxvCpus": 128,
    "desiredvCpus": 4,
    "instanceTypes": [
      "m4"
    ],
    "bidPercentage": 20,
    "subnets": [
      "subnet-220c0e0a",
      "subnet-1a95556d",
      "subnet-978f6dce"
    ],
    "securityGroupIds": [
      "sg-cf5093b2"
    ],
    "ec2KeyPair": "id_rsa",
    "instanceRole": "ecsInstanceRole",
    "tags": {
      "Name": "Batch Instance - M4Spot"
    }
  },
  "serviceRole": "arn:aws:iam::012345678910:role/AWSBatchServiceRole"
}
```

輸出：

```
{
    "computeEnvironmentName": "M4Spot",
    "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/M4Spot"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateComputeEnvironment。

create-job-queue

以下程式碼範例顯示如何使用 create-job-queue。

AWS CLI

使用單一運算環境建立低優先順序任務佇列

此範例會建立名為 LowPriority 的任務佇列，使用 M4Spot 運算環境。

命令：

```
aws batch create-job-queue --cli-input-json file://<path_to_json_file>/
LowPriority.json
```

JSON 檔案格式：

```
{
  "jobQueueName": "LowPriority",
  "state": "ENABLED",
  "priority": 10,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "M4Spot"
    }
  ]
}
```

輸出：

```
{
    "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/LowPriority",
    "jobQueueName": "LowPriority"
}
```

建立具有兩個運算環境的高優先順序任務佇列

此範例會建立名為 HighPriority 的任務佇列，其使用順序為 1 的 C4OnDemand 運算環境，以及順序為 2 的 M4Spot 運算環境。排程器會先嘗試在 C4OnDemand 運算環境中放置任務。

命令：

```
aws batch create-job-queue --cli-input-json file://<path_to_json_file>/
HighPriority.json
```

JSON 檔案格式：

```
{
  "jobQueueName": "HighPriority",
  "state": "ENABLED",
  "priority": 1,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "C4OnDemand"
    },
    {
      "order": 2,
      "computeEnvironment": "M4Spot"
    }
  ]
}
```

輸出：

```
{
    "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/HighPriority",
    "jobQueueName": "HighPriority"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateJobQueue。

delete-compute-environment

以下程式碼範例顯示如何使用 delete-compute-environment。

AWS CLI

刪除運算環境

此範例會刪除 P2OnDemand 運算環境。

命令：

```
aws batch delete-compute-environment --compute-environment P2OnDemand
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteComputeEnvironment。

delete-job-queue

以下程式碼範例顯示如何使用 delete-job-queue。

AWS CLI

刪除任務佇列

此範例會刪除 GPGPU 任務佇列。

命令：

```
aws batch delete-job-queue --job-queue GPGPU
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteJobQueue。

deregister-job-definition

以下程式碼範例顯示如何使用 deregister-job-definition。

AWS CLI

取消註冊任務定義

此範例會取消註冊名為 sleep10 的任務定義。

命令：

```
aws batch deregister-job-definition --job-definition sleep10
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterJobDefinition。

describe-compute-environments

以下程式碼範例顯示如何使用 describe-compute-environments。

AWS CLI

描述運算環境

此範例說明 P2OnDemand 運算環境。

命令：

```
aws batch describe-compute-environments --compute-environments P2OnDemand
```

輸出：

```
{
    "computeEnvironments": [
        {
            "status": "VALID",
            "serviceRole": "arn:aws:iam::012345678910:role/AWSBatchServiceRole",
            "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/P2OnDemand",
            "computeResources": {
                "subnets": [
                    "subnet-220c0e0a",
                    "subnet-1a95556d",
                    "subnet-978f6dce"
                ],
                "tags": {
                    "Name": "Batch Instance - P2OnDemand"
                },
                "desiredvCpus": 48,
                "minvCpus": 0,
                "instanceTypes": [
                    "p2"
                ],
                "securityGroupIds": [
                    "sg-cf5093b2"
                ],
                "instanceRole": "ecsInstanceRole",
                "maxvCpus": 128,
                "type": "EC2",
                "ec2KeyPair": "id_rsa"
            },
            "statusReason": "ComputeEnvironment Healthy",
            "ecsClusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/
P2OnDemand_Batch_2c06f29d-d1fe-3a49-879d-42394c86effc",
            "state": "ENABLED",
            "computeEnvironmentName": "P2OnDemand",
            "type": "MANAGED"
```

```
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeComputeEnvironments。

describe-job-definitions

以下程式碼範例顯示如何使用 describe-job-definitions。

AWS CLI

描述作用中的任務定義

此範例說明所有作用中的任務定義。

命令：

```
aws batch describe-job-definitions --status ACTIVE
```

輸出：

```
{
    "jobDefinitions": [
        {
            "status": "ACTIVE",
            "jobDefinitionArn": "arn:aws:batch:us-east-1:012345678910:job-
definition/sleep60:1",
            "containerProperties": {
                "mountPoints": [],
                "parameters": {},
                "image": "busybox",
                "environment": {},
                "vcpus": 1,
                "command": [
                    "sleep",
                    "60"
                ],
                "volumes": [],
                "memory": 128,
                "ulimits": []
```

```
            },
            "type": "container",
            "jobDefinitionName": "sleep60",
            "revision": 1
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeJobDefinitions。

describe-job-queues

以下程式碼範例顯示如何使用 describe-job-queues。

AWS CLI

描述任務佇列

此範例說明 HighPriority 任務佇列。

命令：

```
aws batch describe-job-queues --job-queues HighPriority
```

輸出：

```
{
    "jobQueues": [
        {
            "status": "VALID",
            "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/
HighPriority",
            "computeEnvironmentOrder": [
                {
                    "computeEnvironment": "arn:aws:batch:us-
east-1:012345678910:compute-environment/C4OnDemand",
                    "order": 1
                }
            ],
            "statusReason": "JobQueue Healthy",
            "priority": 1,
            "state": "ENABLED",
```

```
            "jobQueueName": "HighPriority"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeJobQueues。

describe-jobs

以下程式碼範例顯示如何使用 describe-jobs。

AWS CLI

描述任務

下列describe-jobs範例說明具有指定任務 ID 的任務。

```
aws batch describe-jobs \
    --jobs bcf0b186-a532-4122-842e-2ccab8d54efb
```

輸出：

```
{
    "jobs": [
        {
            "status": "SUBMITTED",
            "container": {
                "mountPoints": [],
                "image": "busybox",
                "environment": [],
                "vcpus": 1,
                "command": [
                    "sleep",
                    "60"
                ],
                "volumes": [],
                "memory": 128,
                "ulimits": []
            },
            "parameters": {},
            "jobDefinition": "arn:aws:batch:us-east-1:012345678910:job-definition/
sleep60:1",
```

```
            "jobQueue": "arn:aws:batch:us-east-1:012345678910:job-queue/
HighPriority",
            "jobId": "bcf0b186-a532-4122-842e-2ccab8d54efb",
            "dependsOn": [],
            "jobName": "example",
            "createdAt": 1480483387803
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeJobs。

`list-jobs`

以下程式碼範例顯示如何使用 `list-jobs`。

AWS CLI

列出執行中的任務

此範例列出 HighPriority 任務佇列中的執行中任務。

命令：

```
aws batch list-jobs --job-queue HighPriority
```

輸出：

```
{
    "jobSummaryList": [
        {
            "jobName": "example",
            "jobId": "e66ff5fd-a1ff-4640-b1a2-0b0a142f49bb"
        }
    ]
}
```

列出提交的任務

此範例列出 HighPriority 任務佇列中處於 SUBMITTED 任務狀態的任務。

命令：

```
aws batch list-jobs --job-queue HighPriority --job-status SUBMITTED
```

輸出：

```
{
    "jobSummaryList": [
        {
            "jobName": "example",
            "jobId": "68f0c163-fbd4-44e6-9fd1-25b14a434786"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListJobs。

register-job-definition

以下程式碼範例顯示如何使用 register-job-definition。

AWS CLI

註冊任務定義

此範例會註冊簡單容器任務的任務定義。

命令：

```
aws batch register-job-definition --job-definition-name sleep30 --type container --
container-properties '{ "image": "busybox", "vcpus": 1, "memory": 128, "command":
 [ "sleep", "30"]}'
```

輸出：

```
{
    "jobDefinitionArn": "arn:aws:batch:us-east-1:012345678910:job-definition/
sleep30:1",
    "jobDefinitionName": "sleep30",
    "revision": 1
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterJobDefinition。

submit-job

以下程式碼範例顯示如何使用 submit-job。

AWS CLI

提交任務

此範例會將稱為範例的簡單容器任務提交至 HighPriority 任務佇列。

命令：

```
aws batch submit-job --job-name example --job-queue HighPriority  --job-
definition sleep60
```

輸出：

```
{
    "jobName": "example",
    "jobId": "876da822-4198-45f2-a252-6cea32512ea8"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SubmitJob。

terminate-job

以下程式碼範例顯示如何使用 terminate-job。

AWS CLI

終止任務

此範例會終止具有指定任務 ID 的任務。

命令：

```
aws batch terminate-job --job-id 61e743ed-35e4-48da-b2de-5c8333821c84 --
reason "Terminating job."
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TerminateJob。

update-compute-environment

以下程式碼範例顯示如何使用 update-compute-environment。

AWS CLI

更新運算環境

此範例會停用 P2OnDemand 運算環境，以便將其刪除。

命令：

```
aws batch update-compute-environment --compute-environment P2OnDemand --
state DISABLED
```

輸出：

```
{
    "computeEnvironmentName": "P2OnDemand",
    "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/P2OnDemand"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateComputeEnvironment。

update-job-queue

以下程式碼範例顯示如何使用 update-job-queue。

AWS CLI

更新任務佇列

此範例會停用任務佇列，以便將其刪除。

命令：

```
aws batch update-job-queue --job-queue GPGPU --state DISABLED
```

輸出：

```
{
    "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/GPGPU",
    "jobQueueName": "GPGPU"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateJobQueue。

AWS Budgets 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS Budgets。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-budget`

以下程式碼範例顯示如何使用 `create-budget`。

AWS CLI

建立成本和用量預算

下列`create-budget`命令會建立成本和用量預算。

```
aws budgets create-budget \
    --account-id 111122223333 \
    --budget file://budget.json \
    --notifications-with-subscribers file://notifications-with-subscribers.json
```

budget.json 的內容：

```
{
    "BudgetLimit": {
        "Amount": "100",
        "Unit": "USD"
    },
    "BudgetName": "Example Tag Budget",
    "BudgetType": "COST",
    "CostFilters": {
        "TagKeyValue": [
            "user:Key$value1",
            "user:Key$value2"
        ]
    },
    "CostTypes": {
        "IncludeCredit": true,
        "IncludeDiscount": true,
        "IncludeOtherSubscription": true,
        "IncludeRecurring": true,
        "IncludeRefund": true,
        "IncludeSubscription": true,
        "IncludeSupport": true,
        "IncludeTax": true,
        "IncludeUpfront": true,
        "UseBlended": false
    },
    "TimePeriod": {
        "Start": 1477958399,
        "End": 3706473600
    },
    "TimeUnit": "MONTHLY"
}
```

notifications-with-subscribers.json 的內容：

```
[
    {
        "Notification": {
            "ComparisonOperator": "GREATER_THAN",
            "NotificationType": "ACTUAL",
            "Threshold": 80,
            "ThresholdType": "PERCENTAGE"
        },
        "Subscribers": [
```

```
            {
                "Address": "example@example.com",
                "SubscriptionType": "EMAIL"
            }
        ]
    }
]
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateBudget。

create-notification

以下程式碼範例顯示如何使用 create-notification。

AWS CLI

為指定的成本和用量預算建立通知

此範例會為指定的成本和用量預算建立通知。

命令：

```
aws budgets create-notification --account-id 111122223333 --budget-name "Example
 Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
 --subscriber SubscriptionType=EMAIL,Address=example@example.com
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateNotification。

create-subscriber

以下程式碼範例顯示如何使用 create-subscriber。

AWS CLI

為與成本和用量預算相關聯的通知建立訂閱者

此範例會為指定的通知建立訂閱者。

命令：

```
aws budgets create-subscriber --account-id 111122223333 --budget-name "Example
 Budget" --
```

```
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
 --subscriber SubscriptionType=EMAIL,Address=example@example.com
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSubscriber。

delete-budget

以下程式碼範例顯示如何使用 delete-budget。

AWS CLI

刪除成本和用量預算

此範例會刪除指定的成本和用量預算。

命令：

```
aws budgets delete-budget --account-id 111122223333 --budget-name "Example Budget"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBudget。

delete-notification

以下程式碼範例顯示如何使用 delete-notification。

AWS CLI

從預算中刪除通知

此範例會從指定的預算中刪除指定的通知。

命令：

```
aws budgets delete-notification --account-id 111122223333 --budget-name "Example
 Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNotification。

delete-subscriber

以下程式碼範例顯示如何使用 delete-subscriber。

AWS CLI

從通知中刪除訂閱者

此範例會從指定的通知中刪除指定的訂閱者。

命令：

```
aws budgets delete-subscriber --account-id 111122223333 --budget-name "Example
 Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
 --subscriber SubscriptionType=EMAIL,Address=example@example.com
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSubscriber。

describe-budget

以下程式碼範例顯示如何使用 describe-budget。

AWS CLI

擷取與 帳戶相關聯的預算

此範例會擷取指定的成本和用量預算。

命令：

```
aws budgets describe-budget --account-id 111122223333 --budget-name "Example Budget"
```

輸出：

```
{
   "Budget": {
       "CalculatedSpend": {
           "ForecastedSpend": {
               "Amount": "2641.548000000000229192664846777915954589843 75",
               "Unit": "USD"
           },
           "ActualSpend": {
               "Amount": "604.456000000000017280399333685636520385742 1875",
               "Unit": "USD"
```

```
            }
        },
        "BudgetType": "COST",
        "BudgetLimit": {
            "Amount": "100",
            "Unit": "USD"
        },
        "BudgetName": "Example Budget",
        "CostTypes": {
            "IncludeOtherSubscription": true,
            "IncludeUpfront": true,
            "IncludeRefund": true,
            "UseBlended": false,
            "IncludeDiscount": true,
            "UseAmortized": false,
            "IncludeTax": true,
            "IncludeCredit": true,
            "IncludeSupport": true,
            "IncludeRecurring": true,
            "IncludeSubscription": true
        },
        "TimeUnit": "MONTHLY",
        "TimePeriod": {
            "Start": 1477958399.0,
            "End": 3706473600.0
        },
        "CostFilters": {
            "AZ": [
                "us-east-1"
            ]
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeBudget。

describe-budgets

以下程式碼範例顯示如何使用 describe-budgets。

AWS CLI

擷取與 帳戶相關聯的預算

此範例會擷取 帳戶的成本和用量預算。

命令：

```
aws budgets describe-budgets --account-id 111122223333 --max-results 20
```

輸出：

```
{
   "Budgets": [
       {
           "CalculatedSpend": {
               "ForecastedSpend": {
                   "Amount": "2641.548000000000229192664846777915954589843 75",
                   "Unit": "USD"
               },
               "ActualSpend": {
                   "Amount": "604.456000000000017280399333685636520385742 1875",
                   "Unit": "USD"
               }
           },
           "BudgetType": "COST",
           "BudgetLimit": {
               "Amount": "100",
               "Unit": "USD"
           },
           "BudgetName": "Example Budget",
           "CostTypes": {
               "IncludeOtherSubscription": true,
               "IncludeUpfront": true,
               "IncludeRefund": true,
               "UseBlended": false,
               "IncludeDiscount": true,
               "UseAmortized": false,
               "IncludeTax": true,
               "IncludeCredit": true,
               "IncludeSupport": true,
               "IncludeRecurring": true,
               "IncludeSubscription": true
           },
           "TimeUnit": "MONTHLY",
           "TimePeriod": {
               "Start": 1477958399.0,
```

```
                "End": 3706473600.0
            },
            "CostFilters": {
                "AZ": [
                    "us-east-1"
                ]
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeBudgets。

describe-notifications-for-budget

以下程式碼範例顯示如何使用 describe-notifications-for-budget。

AWS CLI

擷取預算的通知

此範例會擷取成本和用量預算的通知。

命令：

```
aws budgets describe-notifications-for-budget --account-id 111122223333 --budget-
name "Example Budget" --max-results 5
```

輸出：

```
{
    "Notifications": [
        {
            "Threshold": 80.0,
            "ComparisonOperator": "GREATER_THAN",
            "NotificationType": "ACTUAL"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeNotificationsForBudget。

describe-subscribers-for-notification

以下程式碼範例顯示如何使用 describe-subscribers-for-notification。

AWS CLI

擷取預算通知的訂閱者

此範例會擷取成本和用量預算通知的訂閱者。

命令：

```
aws budgets describe-subscribers-for-notification --
account-id 111122223333 --budget-name "Example Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
 --max-results 5
```

輸出：

```
{
   "Subscribers": [
       {
           "SubscriptionType": "EMAIL",
           "Address": "example2@example.com"
       },
       {
           "SubscriptionType": "EMAIL",
           "Address": "example@example.com"
       }
   ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSubscribersForNotification。

update-budget

以下程式碼範例顯示如何使用 update-budget。

AWS CLI

取代成本和用量預算的預算

此範例會將成本和用量預算取代為新的預算。

命令：

```
aws budgets update-budget --account-id 111122223333 --new-budget file://new-budget.json
```

new-budget.json：

```
{
    "BudgetLimit": {
       "Amount": "100",
       "Unit": "USD"
    },
    "BudgetName": "Example Budget",
    "BudgetType": "COST",
    "CostFilters": {
       "AZ" : [ "us-east-1" ]
    },
    "CostTypes": {
       "IncludeCredit": false,
       "IncludeDiscount": true,
       "IncludeOtherSubscription": true,
       "IncludeRecurring": true,
       "IncludeRefund": true,
       "IncludeSubscription": true,
       "IncludeSupport": true,
       "IncludeTax": true,
       "IncludeUpfront": true,
       "UseBlended": false,
       "UseAmortized": true
    },
    "TimePeriod": {
       "Start": 1477958399,
       "End": 3706473600
    },
    "TimeUnit": "MONTHLY"
 }
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateBudget。

update-notification

以下程式碼範例顯示如何使用 update-notification。

AWS CLI

取代成本和用量預算的通知

此範例會以 90% 的通知取代成本和用量預算的 80% 通知。

命令：

```
aws budgets update-notification --account-id 111122223333 --budget-name "Example
 Budget" --old-notification
  NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENT
 --new-notification
  NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=90,ThresholdType=PERCENT
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateNotification。

update-subscriber

以下程式碼範例顯示如何使用 update-subscriber。

AWS CLI

取代成本和用量預算的訂閱者

此範例會取代成本和用量預算的訂閱者。

命令：

```
aws budgets update-subscriber --account-id 111122223333 --budget-name "Example
 Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
 --old-subscriber SubscriptionType=EMAIL,Address=example@example.com --new-
subscriber SubscriptionType=EMAIL,Address=example2@example.com
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSubscriber。

使用 的 Amazon Chime 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon Chime 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-phone-number-with-user`

以下程式碼範例顯示如何使用 `associate-phone-number-with-user`。

AWS CLI

將電話號碼與使用者建立關聯

下列`associate-phone-number-with-user`範例會將指定的電話號碼與使用者建立關聯。

```
aws chime associate-phone-number-with-user \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --user-id 1ab2345c-67de-8901-f23g-45h678901j2k \
    --e164-phone-number "+12065550100"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的管理使用者電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociatePhoneNumberWithUser。

`associate-signin-delegate-groups-with-account`

以下程式碼範例顯示如何使用 `associate-signin-delegate-groups-with-account`。

AWS CLI

建立登入委派群組的關聯

下列`associate-signin-delegate-groups-with-account`範例會將指定的登入委派群組與指定的 Amazon Chime 帳戶建立關聯。

```
aws chime associate-signin-delegate-groups-with-account \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --signin-delegate-groups GroupName=my_users
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的管理使用者存取和許可。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateSigninDelegateGroupsWithAccount。

batch-create-room-membership

以下程式碼範例顯示如何使用 batch-create-room-membership。

AWS CLI

建立多個房間成員資格

下列batch-create-room-membership範例會將多個使用者新增至聊天室，做為聊天室成員。它也會將管理員和成員角色指派給使用者。

```
aws chime batch-create-room-membership \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \
    --membership-item-list "MemberId=1ab2345c-67de-8901-
f23g-45h678901j2k,Role=Administrator" "MemberId=2ab2345c-67de-8901-
f23g-45h678901j2k,Role=Member"
```

輸出：

```
{
    "ResponseMetadata": {
        "RequestId": "169ba401-d886-475f-8b3f-e01eac6fadfb",
        "HTTPStatusCode": 201,
        "HTTPHeaders": {
            "x-amzn-requestid": "169ba401-d886-475f-8b3f-e01eac6fadfb",
            "content-type": "application/json",
            "content-length": "13",
            "date": "Mon, 02 Dec 2019 22:46:58 GMT",
            "connection": "keep-alive"
        },
```

```
        "RetryAttempts": 0
    },
    "Errors": []
}
```

如需詳細資訊，請參閱《Amazon Chime 使用者指南》中的建立聊天室。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchCreateRoomMembership。

batch-delete-phone-number

以下程式碼範例顯示如何使用 batch-delete-phone-number。

AWS CLI

刪除多個電話號碼

下列batch-delete-phone-number範例會刪除所有指定的電話號碼。

```
aws chime batch-delete-phone-number \
    --phone-number-ids "%2B12065550100" "%2B12065550101"
```

此命令不會產生輸出。輸出：

```
{
    "PhoneNumberErrors": []
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的使用電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDeletePhoneNumber。

batch-suspend-user

以下程式碼範例顯示如何使用 batch-suspend-user。

AWS CLI

暫停多個使用者

下列batch-suspend-user範例會從指定的 Amazon Chime 帳戶暫停列出的使用者。

```
aws chime batch-suspend-user \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --user-id-list "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE" "a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE" "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE"
```

輸出：

```
{
    "UserErrors": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchSuspendUser。

batch-unsuspend-user

以下程式碼範例顯示如何使用 batch-unsuspend-user。

AWS CLI

取消暫停多個使用者

下列batch-unsuspend-user範例會針對指定 Amazon Chime 帳戶上列出的使用者移除任何先前的停用。

```
aws chime batch-unsuspend-user \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --user-id-list "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE" "a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE" "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE"
```

輸出：

```
{
    "UserErrors": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchUnsuspendUser。

batch-update-phone-number

以下程式碼範例顯示如何使用 batch-update-phone-number。

AWS CLI

同時更新數個電話號碼產品類型

下列batch-update-phone-number範例會更新所有指定電話號碼的產品類型。

```
aws chime batch-update-phone-number \
    --update-phone-number-request-items PhoneNumberId=
%2B12065550100,ProductType=BusinessCalling PhoneNumberId=
%2B12065550101,ProductType=BusinessCalling
```

輸出：

```
{
    "PhoneNumberErrors": []
}
```

同時更新數個電話號碼呼叫名稱

下列batch-update-phone-number範例會更新所有指定電話號碼的呼叫名稱。

```
aws chime batch-update-phone-number \
    --update-phone-number-request-items PhoneNumberId=
%2B14013143874,CallingName=phonenumber1 PhoneNumberId=
%2B14013144061,CallingName=phonenumber2
```

輸出：

```
{
    "PhoneNumberErrors": []
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的使用電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchUpdatePhoneNumber。

batch-update-user

以下程式碼範例顯示如何使用 batch-update-user。

AWS CLI

在單一命令中更新多個使用者

下列batch-update-user範例會LicenseType為指定 Amazon Chime 帳戶中列出的每個使用者更新 。

```
aws chime batch-update-user \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
    --update-user-request-items "UserId=a1b2c3d4-5678-90ab-
cdef-22222EXAMPLE,LicenseType=Basic" "UserId=a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE,LicenseType=Basic"
```

輸出：

```
{
    "UserErrors": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchUpdateUser。

create-account

以下程式碼範例顯示如何使用 create-account。

AWS CLI

建立 帳戶

下列create-account範例會在管理員的帳戶下建立 Amazon Chime AWS 帳戶。

```
aws chime create-account \
    --name MyChimeAccount
```

輸出：

```
{
    "Account": {
        "AwsAccountId": "111122223333",
        "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "Name": "MyChimeAccount",
```

```
        "AccountType": "Team",
        "CreatedTimestamp": "2019-01-04T17:11:22.003Z",
        "DefaultLicense": "Pro",
        "SupportedLicenses": [
            "Basic",
            "Pro"
        ],
        "SigninDelegateGroups": [
            {
                "GroupName": "myGroup"
            },
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAccount。

create-bot

以下程式碼範例顯示如何使用 create-bot。

AWS CLI

建立 Amazon Chime 機器人

下列create-bot範例會為指定的 Amazon Chime Enterprise 帳戶建立機器人。

```
aws chime create-bot \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --display-name "myBot" \
    --domain "example.com"
```

輸出：

```
{
    "Bot": {
        "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "DisplayName": "myBot (Bot)",
        "BotType": "ChatBot",
        "Disabled": false,
```

```
        "CreatedTimestamp": "2019-09-09T18:05:56.749Z",
        "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",
        "BotEmail": "myBot-chimebot@example.com",
        "SecurityToken": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 開發人員指南》中的將聊天機器人與 Amazon Chime 整合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateBot。

create-phone-number-order

以下程式碼範例顯示如何使用 create-phone-number-order。

AWS CLI

建立電話號碼順序

下列create-phone-number-order範例會為指定的電話號碼建立電話號碼順序。

```
aws chime create-phone-number-order \
    --product-type VoiceConnector \
    --e164-phone-numbers "+12065550100" "+12065550101" "+12065550102"
```

輸出：

```
{
    "PhoneNumberOrder": {
        "PhoneNumberOrderId": "abc12345-de67-89f0-123g-h45i678j9012",
        "ProductType": "VoiceConnector",
        "Status": "Processing",
        "OrderedPhoneNumbers": [
            {
                "E164PhoneNumber": "+12065550100",
                "Status": "Processing"
            },
            {
               "E164PhoneNumber": "+12065550101",
               "Status": "Processing"
            },
```

```
            {
              "E164PhoneNumber": "+12065550102",
              "Status": "Processing"
            }
        ],
        "CreatedTimestamp": "2019-08-09T21:35:21.427Z",
        "UpdatedTimestamp": "2019-08-09T21:35:22.408Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的使用電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePhoneNumberOrder。

create-room-membership

以下程式碼範例顯示如何使用 create-room-membership。

AWS CLI

建立房間成員資格

下列create-room-membership範例會將指定的使用者新增至聊天室，做為聊天室成員。

```
aws chime create-room-membership \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \
    --member-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

輸出：

```
{
    "RoomMembership": {
        "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
        "Member": {
            "MemberId": "1ab2345c-67de-8901-f23g-45h678901j2k",
            "MemberType": "User",
            "Email": "janed@example.com",
            "FullName": "Jane Doe",
            "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45"
        },
        "Role": "Member",
        "InvitedBy": "arn:aws:iam::111122223333:user/alejandro",
```

```
        "UpdatedTimestamp": "2019-12-02T22:36:41.969Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 使用者指南》中的建立聊天室。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRoomMembership。

create-room

以下程式碼範例顯示如何使用 create-room。

AWS CLI

建立聊天室

下列create-room範例會為指定的 Amazon Chime 帳戶建立聊天室。

```
aws chime create-room \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --name chatRoom
```

輸出：

```
{
    "Room": {
        "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
        "Name": "chatRoom",
        "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",
        "CreatedBy": "arn:aws:iam::111122223333:user/alejandro",
        "CreatedTimestamp": "2019-12-02T22:29:31.549Z",
        "UpdatedTimestamp": "2019-12-02T22:29:31.549Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 使用者指南》中的建立聊天室。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRoom。

create-user

以下程式碼範例顯示如何使用 create-user。

AWS CLI

為共用裝置建立使用者設定檔

下列create-user範例會為指定的電子郵件地址建立共用裝置設定檔。

```
aws chime create-user \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --email roomdevice@example.com \
    --user-type SharedDevice
```

輸出：

```
{
    "User": {
        "UserId": "1ab2345c-67de-8901-f23g-45h678901j2k",
        "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",
        "PrimaryEmail": "roomdevice@example.com",
        "DisplayName": "Room Device",
        "LicenseType": "Pro",
        "UserType": "SharedDevice",
        "UserRegistrationStatus": "Registered",
        "RegisteredOn": "2020-01-15T22:38:09.806Z",
        "AlexaForBusinessMetadata": {
            "IsAlexaForBusinessEnabled": false
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的準備設定。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateUser。

delete-account

以下程式碼範例顯示如何使用 delete-account。

AWS CLI

刪除 帳戶

下列delete-account範例會刪除指定的帳戶。

```
aws chime delete-account --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的刪除您的帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAccount。

delete-phone-number

以下程式碼範例顯示如何使用 delete-phone-number。

AWS CLI

刪除電話號碼

下列delete-phone-number範例會將指定的電話號碼移至刪除佇列。

```
aws chime delete-phone-number \
    --phone-number-id "+12065550100"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的使用電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePhoneNumber。

delete-room-membership

以下程式碼範例顯示如何使用 delete-room-membership。

AWS CLI

將使用者移除為聊天室的成員

下列delete-room-membership範例會從指定的聊天室移除指定的成員。

```
aws chime delete-room-membership \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \
    --member-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Chime 使用者指南》中的建立聊天室。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRoomMembership。

delete-room

以下程式碼範例顯示如何使用 delete-room。

AWS CLI

刪除聊天室

下列delete-room範例會刪除指定的聊天室，並移除聊天室成員資格。

```
aws chime delete-room \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Chime 使用者指南》中的建立聊天室。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRoom。

disassociate-phone-number-from-user

以下程式碼範例顯示如何使用 disassociate-phone-number-from-user。

AWS CLI

取消電話號碼與使用者的關聯

下列disassociate-phone-number-from-user範例會取消電話號碼與指定使用者的關聯。

```
aws chime disassociate-phone-number-from-user \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --user-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的管理使用者電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociatePhoneNumberFromUser。

disassociate-signin-delegate-groups-from-account

以下程式碼範例顯示如何使用 disassociate-signin-delegate-groups-from-account。

AWS CLI

取消與登入委派群組的關聯

下列disassociate-signin-delegate-groups-from-account範例會將指定的登入委派群組與指定的 Amazon Chime 帳戶取消關聯。

```
aws chime disassociate-signin-delegate-groups-from-account \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --group-names "my_users"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的管理使用者存取和許可。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateSigninDelegateGroupsFromAccount。

get-account-settings

以下程式碼範例顯示如何使用 get-account-settings。

AWS CLI

擷取 帳戶的設定

下列get-account-settings範例會擷取指定帳戶的帳戶設定。

```
aws chime get-account-settings --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

輸出：

```
{
    "AccountSettings": {
        "DisableRemoteControl": false,
```

```
        "EnableDialOut": false
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的管理您的 Amazon Chime 帳戶。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetAccountSettings。

get-account

以下程式碼範例顯示如何使用 get-account。

AWS CLI

擷取帳戶的詳細資訊

下列get-account範例會擷取指定 Amazon Chime 帳戶的詳細資訊。

```
aws chime get-account \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

輸出：

```
{
    "Account": {
        "AwsAccountId": "111122223333",
        "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "Name": "EnterpriseDirectory",
        "AccountType": "EnterpriseDirectory",
        "CreatedTimestamp": "2018-12-20T18:38:02.181Z",
        "DefaultLicense": "Pro",
        "SupportedLicenses": [
            "Basic",
            "Pro"
        ],
        "SigninDelegateGroups": [
            {
                "GroupName": "myGroup"
            },
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的管理您的 Amazon Chime 帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAccount。

get-bot

以下程式碼範例顯示如何使用 get-bot。

AWS CLI

擷取機器人的詳細資訊

下列get-bot範例顯示指定機器人的詳細資訊。

```
aws chime get-bot \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k
```

輸出：

```
{
    "Bot": {
        "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "DisplayName": "myBot (Bot)",
        "BotType": "ChatBot",
        "Disabled": false,
        "CreatedTimestamp": "2019-09-09T18:05:56.749Z",
        "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",
        "BotEmail": "myBot-chimebot@example.com",
        "SecurityToken": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 開發人員指南》中的更新聊天機器人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBot。

get-global-settings

以下程式碼範例顯示如何使用 get-global-settings。

AWS CLI

取得全域設定

下列get-global-settings範例會擷取用於存放與管理員 AWS 帳戶相關聯之 Amazon Chime Business Calling 和 Amazon Chime Voice Connectors 的通話詳細資訊記錄的 S3 儲存貯體名稱。

```
aws chime get-global-settings
```

輸出：

```
{
    "BusinessCalling": {
        "CdrBucket": "s3bucket"
    },
    "VoiceConnector": {
        "CdrBucket": "s3bucket"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的管理全域設定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetGlobalSettings。

get-phone-number-order

以下程式碼範例顯示如何使用 get-phone-number-order。

AWS CLI

取得電話號碼順序的詳細資訊

下列get-phone-number-order範例顯示指定電話號碼順序的詳細資訊。

```
aws chime get-phone-number-order \
    --phone-number-order-id abc12345-de67-89f0-123g-h45i678j9012
```

輸出：

```
{
    "PhoneNumberOrder": {
        "PhoneNumberOrderId": "abc12345-de67-89f0-123g-h45i678j9012",
```

```
        "ProductType": "VoiceConnector",
        "Status": "Partial",
        "OrderedPhoneNumbers": [
            {
              "E164PhoneNumber": "+12065550100",
              "Status": "Acquired"
            },
            {
                "E164PhoneNumber": "+12065550101",
                "Status": "Acquired"
            },
            {
                "E164PhoneNumber": "+12065550102",
                "Status": "Failed"
            }
        ],
        "CreatedTimestamp": "2019-08-09T21:35:21.427Z",
        "UpdatedTimestamp": "2019-08-09T21:35:31.926Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的使用電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPhoneNumberOrder。

get-phone-number-settings

以下程式碼範例顯示如何使用 get-phone-number-settings。

AWS CLI

擷取外撥通話名稱

下列get-phone-number-settings範例會擷取呼叫使用者帳戶的預設外撥呼叫名稱 AWS 。

```
aws chime get-phone-number-settings
```

此命令不會產生輸出。輸出：

```
{
    "CallingName": "myName",
    "CallingNameUpdatedTimestamp": "2019-10-28T18:56:42.911Z"
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的使用電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPhoneNumberSettings。

get-phone-number

以下程式碼範例顯示如何使用 get-phone-number。

AWS CLI

取得電話號碼詳細資訊

下列get-phone-number範例顯示指定電話號碼的詳細資訊。

```
aws chime get-phone-number \
    --phone-number-id +12065550100
```

輸出：

```
{
    "PhoneNumber": {
        "PhoneNumberId": "%2B12065550100",
        "E164PhoneNumber": "+12065550100",
        "Type": "Local",
        "ProductType": "VoiceConnector",
        "Status": "Unassigned",
        "Capabilities": {
            "InboundCall": true,
            "OutboundCall": true,
            "InboundSMS": true,
            "OutboundSMS": true,
            "InboundMMS": true,
            "OutboundMMS": true
        },
       "Associations": [
            {
                "Value": "abcdef1ghij2klmno3pqr4",
                "Name": "VoiceConnectorId",
                "AssociatedTimestamp": "2019-10-28T18:40:37.453Z"
            }
        ],
        "CallingNameStatus": "UpdateInProgress",
        "CreatedTimestamp": "2019-08-09T21:35:21.445Z",
```

```
        "UpdatedTimestamp": "2019-08-09T21:35:31.745Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的使用電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPhoneNumber。

get-room

以下程式碼範例顯示如何使用 get-room。

AWS CLI

取得聊天室的詳細資訊

下列get-room範例顯示指定聊天室的詳細資訊。

```
aws chime get-room \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

輸出：

```
{
    "Room": {
        "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
        "Name": "chatRoom",
        "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",
        "CreatedBy": "arn:aws:iam::111122223333:user/alejandro",
        "CreatedTimestamp": "2019-12-02T22:29:31.549Z",
        "UpdatedTimestamp": "2019-12-02T22:29:31.549Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 使用者指南》中的建立聊天室。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRoom。

get-user-settings

以下程式碼範例顯示如何使用 get-user-settings。

AWS CLI

擷取使用者設定

下列get-user-settings範例顯示指定的使用者設定。

```
aws chime get-user-settings \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --user-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

輸出：

```
{
    "UserSettings": {
        "Telephony": {
            "InboundCalling": true,
            "OutboundCalling": true,
            "SMS": true
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的管理使用者電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetUserSettings。

get-user

以下程式碼範例顯示如何使用 get-user。

AWS CLI

取得使用者的詳細資訊

下列get-user範例會擷取指定使用者的詳細資訊。

```
aws chime get-user \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

輸出：

```
{
    "User": {
        "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
        "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "PrimaryEmail": "marthar@example.com",
        "DisplayName": "Martha Rivera",
        "LicenseType": "Pro",
        "UserRegistrationStatus": "Registered",
        "RegisteredOn": "2018-12-20T18:45:25.231Z",
        "InvitedOn": "2018-12-20T18:45:25.231Z",
        "AlexaForBusinessMetadata": {
            "IsAlexaForBusinessEnabled": False,
            "AlexaForBusinessRoomArn": "null"
        },
        "PersonalPIN": "XXXXXXXXXX"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的管理使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetUser。

invite-users

以下程式碼範例顯示如何使用 `invite-users`。

AWS CLI

邀請使用者加入 Amazon Chime

下列`invite-users`範例會傳送電子郵件，邀請使用者加入指定的 Amazon Chime 帳戶。

```
aws chime invite-users \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --user-email-list "alejandror@example.com" "janed@example.com"
```

輸出：

```
{
    "Invites": [
        {
            "InviteId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
```

```
            "Status": "Pending",
            "EmailAddress": "alejandror@example.com",
            "EmailStatus": "Sent"
        }
        {
            "InviteId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
            "Status": "Pending",
            "EmailAddress": "janed@example.com",
            "EmailStatus": "Sent"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的邀請和暫停使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 InviteUsers。

`list-accounts`

以下程式碼範例顯示如何使用 `list-accounts`。

AWS CLI

取得帳戶清單

下列`list-accounts`範例會擷取管理員帳戶中的 Amazon Chime AWS 帳戶清單。

```
aws chime list-accounts
```

輸出：

```
{
    "Accounts": [
        {
            "AwsAccountId": "111122223333",
            "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "Name": "First Chime Account",
            "AccountType": "EnterpriseDirectory",
            "CreatedTimestamp": "2018-12-20T18:38:02.181Z",
            "DefaultLicense": "Pro",
            "SupportedLicenses": [
                "Basic",
                "Pro"
```

```
            ],
            "SigninDelegateGroups": [
                {
                    "GroupName": "myGroup"
                },
            ]
        },
        {
            "AwsAccountId": "111122223333",
            "AccountId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
            "Name": "Second Chime Account",
            "AccountType": "Team",
            "CreatedTimestamp": "2018-09-04T21:44:22.292Z",
            "DefaultLicense": "Pro",
            "SupportedLicenses": [
                "Basic",
                "Pro"
            ],
            "SigninDelegateGroups": [
                {
                    "GroupName": "myGroup"
                },
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的管理您的 Amazon Chime 帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAccounts。

list-bots

以下程式碼範例顯示如何使用 `list-bots`。

AWS CLI

擷取機器人清單

下列`list-bots`範例列出與指定 Amazon Chime Enterprise 帳戶相關聯的機器人。

```
aws chime list-bots \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45
```

輸出：

```
{
    "Bot": {
        "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "DisplayName": "myBot (Bot)",
        "BotType": "ChatBot",
        "Disabled": false,
        "CreatedTimestamp": "2019-09-09T18:05:56.749Z",
        "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",
        "BotEmail": "myBot-chimebot@example.com",
        "SecurityToken": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 開發人員指南》中的搭配使用聊天機器人與 Amazon Chime。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBots。

`list-phone-number-orders`

以下程式碼範例顯示如何使用 `list-phone-number-orders`。

AWS CLI

列出電話號碼訂單

下列`list-phone-number-orders`範例列出與 Amazon Chime 管理員帳戶相關聯的電話號碼訂單。

```
aws chime list-phone-number-orders
```

輸出：

```
{
    "PhoneNumberOrders": [
        {
            "PhoneNumberOrderId": "abc12345-de67-89f0-123g-h45i678j9012",
            "ProductType": "VoiceConnector",
            "Status": "Partial",
```

```
            "OrderedPhoneNumbers": [
                {
                    "E164PhoneNumber": "+12065550100",
                    "Status": "Acquired"
                },
                {
                    "E164PhoneNumber": "+12065550101",
                    "Status": "Acquired"
                },
                {
                    "E164PhoneNumber": "+12065550102",
                    "Status": "Failed"
                }
            ],
            "CreatedTimestamp": "2019-08-09T21:35:21.427Z",
            "UpdatedTimestamp": "2019-08-09T21:35:31.926Z"
        }
        {
            "PhoneNumberOrderId": "cba54321-ed76-09f5-321g-h54i876j2109",
            "ProductType": "BusinessCalling",
            "Status": "Partial",
            "OrderedPhoneNumbers": [
                {
                    "E164PhoneNumber": "+12065550103",
                    "Status": "Acquired"
                },
                {
                    "E164PhoneNumber": "+12065550104",
                    "Status": "Acquired"
                },
                {
                    "E164PhoneNumber": "+12065550105",
                    "Status": "Failed"
                }
            ],
            "CreatedTimestamp": "2019-08-09T21:35:21.427Z",
            "UpdatedTimestamp": "2019-08-09T21:35:31.926Z"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的使用電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPhoneNumberOrders。

list-phone-numbers

以下程式碼範例顯示如何使用 list-phone-numbers。

AWS CLI

列出 Amazon Chime 帳戶的電話號碼

下列list-phone-numbers範例列出與管理員 Amazon Chime 帳戶相關聯的電話號碼。

```
aws chime list-phone-numbers
```

此命令不會產生輸出。輸出：

```
{
    "PhoneNumbers": [
        {
            "PhoneNumberId": "%2B12065550100",
            "E164PhoneNumber": "+12065550100",
            "Type": "Local",
            "ProductType": "VoiceConnector",
            "Status": "Assigned",
            "Capabilities": {
                "InboundCall": true,
                "OutboundCall": true,
                "InboundSMS": true,
                "OutboundSMS": true,
                "InboundMMS": true,
                "OutboundMMS": true
            },
            "Associations": [
                {
                    "Value": "abcdef1ghij2klmno3pqr4",
                    "Name": "VoiceConnectorId",
                    "AssociatedTimestamp": "2019-10-28T18:40:37.453Z"
                }
            ],
            "CallingNameStatus": "UpdateInProgress",
            "CreatedTimestamp": "2019-08-12T22:10:20.521Z",
            "UpdatedTimestamp": "2019-10-28T18:42:07.964Z"
        },
        {
            "PhoneNumberId": "%2B12065550101",
```

```
            "E164PhoneNumber": "+12065550101",
            "Type": "Local",
            "ProductType": "VoiceConnector",
            "Status": "Assigned",
            "Capabilities": {
                "InboundCall": true,
                "OutboundCall": true,
                "InboundSMS": true,
                "OutboundSMS": true,
                "InboundMMS": true,
                "OutboundMMS": true
            },
            "Associations": [
                {
                    "Value": "abcdef1ghij2klmno3pqr4",
                    "Name": "VoiceConnectorId",
                    "AssociatedTimestamp": "2019-10-28T18:40:37.511Z"
                }
            ],
            "CallingNameStatus": "UpdateInProgress",
            "CreatedTimestamp": "2019-08-12T22:10:20.521Z",
            "UpdatedTimestamp": "2019-10-28T18:42:07.960Z"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的使用電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPhoneNumbers。

`list-room-memberships`

以下程式碼範例顯示如何使用 `list-room-memberships`。

AWS CLI

列出房間成員資格

下列`list-room-memberships`範例顯示指定聊天室的成員資格詳細資訊清單。

```
aws chime list-room-memberships \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

輸出：

```
{
    "RoomMemberships": [
        {
            "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
            "Member": {
                "MemberId": "2ab2345c-67de-8901-f23g-45h678901j2k",
                "MemberType": "User",
                "Email": "zhangw@example.com",
                "FullName": "Zhang Wei",
                "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45"
            },
            "Role": "Member",
            "InvitedBy": "arn:aws:iam::111122223333:user/alejandro",
            "UpdatedTimestamp": "2019-12-02T22:46:58.532Z"
        },
        {
            "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
            "Member": {
                "MemberId": "1ab2345c-67de-8901-f23g-45h678901j2k",
                "MemberType": "User",
                "Email": "janed@example.com",
                "FullName": "Jane Doe",
                "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45"
            },
            "Role": "Administrator",
            "InvitedBy": "arn:aws:iam::111122223333:user/alejandro",
            "UpdatedTimestamp": "2019-12-02T22:46:58.532Z"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Chime 使用者指南》中的建立聊天室。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRoomMemberships。

`list-rooms`

以下程式碼範例顯示如何使用 `list-rooms`。

AWS CLI

列出聊天室

下列list-rooms範例顯示指定帳戶中聊天室的清單。清單只會篩選為指定成員所屬的聊天室。

```
aws chime list-rooms \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --member-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

輸出：

```
{
    "Room": {
        "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
        "Name": "teamRoom",
        "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",
        "CreatedBy": "arn:aws:iam::111122223333:user/alejandro",
        "CreatedTimestamp": "2019-12-02T22:29:31.549Z",
        "UpdatedTimestamp": "2019-12-02T22:33:19.310Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 使用者指南》中的建立聊天室。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRooms。

list-users

以下程式碼範例顯示如何使用 list-users。

AWS CLI

列出帳戶中的使用者

下列list-users範例列出指定 Amazon Chime 帳戶的使用者。

```
aws chime list-users --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

輸出：

```
{
    "Users": [
```

```
        {
            "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
            "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "PrimaryEmail": "mariag@example.com",
            "DisplayName": "Maria Garcia",
            "LicenseType": "Pro",
            "UserType": "PrivateUser",
            "UserRegistrationStatus": "Registered",
            "RegisteredOn": "2018-12-20T18:45:25.231Z"
            "AlexaForBusinessMetadata": {
                "IsAlexaForBusinessEnabled": false
            }
        },
        {
            "UserId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
            "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "PrimaryEmail": "richardr@example.com",
            "DisplayName": "Richard Roe",
            "LicenseType": "Pro",
            "UserType": "PrivateUser",
            "UserRegistrationStatus": "Registered",
            "RegisteredOn": "2018-12-20T18:45:45.415Z"
            "AlexaForBusinessMetadata": {
                "IsAlexaForBusinessEnabled": false
            }
        },
        {
            "UserId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
            "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "PrimaryEmail": "saanvis@example.com",
            "DisplayName": "Saanvi Sarkar",
            "LicenseType": "Basic",
            "UserType": "PrivateUser",
            "UserRegistrationStatus": "Registered",
            "RegisteredOn": "2018-12-20T18:46:57.747Z"
            "AlexaForBusinessMetadata": {
                "IsAlexaForBusinessEnabled": false
            }
        },
        {
            "UserId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE",
            "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "PrimaryEmail": "wxiulan@example.com",
            "DisplayName": "Wang Xiulan",
```

```
            "LicenseType": "Basic",
            "UserType": "PrivateUser",
            "UserRegistrationStatus": "Registered",
            "RegisteredOn": "2018-12-20T18:47:15.390Z"
            "AlexaForBusinessMetadata": {
                "IsAlexaForBusinessEnabled": false
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的管理使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListUsers。

`logout-user`

以下程式碼範例顯示如何使用 `logout-user`。

AWS CLI

登出使用者

下列`logout-user`範例會將指定的使用者登出。

```
aws chime logout-user \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 LogoutUser。

`regenerate-security-token`

以下程式碼範例顯示如何使用 `regenerate-security-token`。

AWS CLI

重新產生安全字符

下列`regenerate-security-token`範例會為指定的機器人重新產生安全字符。

```
aws chime regenerate-security-token \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k
```

輸出：

```
{
    "Bot": {
        "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "DisplayName": "myBot (Bot)",
        "BotType": "ChatBot",
        "Disabled": false,
        "CreatedTimestamp": "2019-09-09T18:05:56.749Z",
        "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",
        "BotEmail": "myBot-chimebot@example.com",
        "SecurityToken": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 開發人員指南》中的驗證聊天機器人請求。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegenerateSecurityToken。

reset-personal-pin

以下程式碼範例顯示如何使用 `reset-personal-pin`。

AWS CLI

重設使用者的個人會議 PIN 碼

下列`reset-personal-pin`範例會重設指定使用者的個人會議 PIN 碼。

```
aws chime reset-personal-pin \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
    --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

輸出：

```
{
```

```
    "User": {
        "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
        "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "PrimaryEmail": "mateo@example.com",
        "DisplayName": "Mateo Jackson",
        "LicenseType": "Pro",
        "UserType": "PrivateUser",
        "UserRegistrationStatus": "Registered",
        "RegisteredOn": "2018-12-20T18:45:25.231Z",
        "AlexaForBusinessMetadata": {
            "IsAlexaForBusinessEnabled": False,
            "AlexaForBusinessRoomArn": "null"
        },
        "PersonalPIN": "XXXXXXXXXX"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的變更個人會議 PINs 碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetPersonalPin。

`restore-phone-number`

以下程式碼範例顯示如何使用 `restore-phone-number`。

AWS CLI

還原電話號碼

下列`restore-phone-number`範例會從刪除佇列還原指定的電話號碼。

```
aws chime restore-phone-number \
    --phone-number-id "+12065550100"
```

輸出：

```
{
    "PhoneNumber": {
        "PhoneNumberId": "%2B12065550100",
        "E164PhoneNumber": "+12065550100",
        "Type": "Local",
        "ProductType": "BusinessCalling",
        "Status": "Unassigned",
```

```
        "Capabilities": {
            "InboundCall": true,
            "OutboundCall": true,
            "InboundSMS": true,
            "OutboundSMS": true,
            "InboundMMS": true,
            "OutboundMMS": true
        },
        "Associations": [],
        "CreatedTimestamp": "2019-08-09T21:35:21.445Z",
        "UpdatedTimestamp": "2019-08-12T22:06:36.355Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的使用電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestorePhoneNumber。

search-available-phone-numbers

以下程式碼範例顯示如何使用 search-available-phone-numbers。

AWS CLI

搜尋可用的電話號碼

下列search-available-phone-numbers範例會依區域碼搜尋可用的電話號碼。

```
aws chime search-available-phone-numbers \
    --area-code "206"
```

輸出：

```
{
    "E164PhoneNumbers": [
        "+12065550100",
        "+12065550101",
        "+12065550102",
        "+12065550103",
        "+12065550104",
        "+12065550105",
        "+12065550106",
        "+12065550107",
```

```
        "+12065550108",
        "+12065550109",
    ]
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的使用電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchAvailablePhoneNumbers。

update-account-settings

以下程式碼範例顯示如何使用 update-account-settings。

AWS CLI

更新您帳戶的設定

下列update-account-settings範例會停用指定 Amazon Chime 帳戶的共用畫面的遠端控制。

```
aws chime update-account-settings \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --account-settings DisableRemoteControl=true
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAccountSettings。

update-account

以下程式碼範例顯示如何使用 update-account。

AWS CLI

更新 帳戶

下列update-account範例會更新指定的帳戶名稱。

```
aws chime update-account \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --name MyAccountName
```

輸出：

```
{
    "Account": {
        "AwsAccountId": "111122223333",
        "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "Name": "MyAccountName",
        "AccountType": "Team",
        "CreatedTimestamp": "2018-09-04T21:44:22.292Z",
        "DefaultLicense": "Pro",
        "SupportedLicenses": [
            "Basic",
            "Pro"
        ],
        "SigninDelegateGroups": [
            {
                "GroupName": "myGroup"
            },
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的重新命名您的帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAccount。

update-bot

以下程式碼範例顯示如何使用 update-bot。

AWS CLI

更新機器人

下列update-bot範例會更新指定機器人的狀態，使其停止執行。

```
aws chime update-bot \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k \
    --disabled
```

輸出：

```
{
```

```
    "Bot": {
        "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",
        "DisplayName": "myBot (Bot)",
        "BotType": "ChatBot",
        "Disabled": true,
        "CreatedTimestamp": "2019-09-09T18:05:56.749Z",
        "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",
        "BotEmail": "myBot-chimebot@example.com",
        "SecurityToken": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 開發人員指南》中的更新聊天機器人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateBot。

update-global-settings

以下程式碼範例顯示如何使用 update-global-settings。

AWS CLI

更新全域設定

下列update-global-settings範例會更新用於存放與管理員 AWS 帳戶相關聯之 Amazon Chime Business Calling 和 Amazon Chime Voice Connectors 的通話詳細資訊記錄的 S3 儲存貯體。

```
aws chime update-global-settings \
    --business-calling CdrBucket="s3bucket" \
    --voice-connector CdrBucket="s3bucket"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的管理全域設定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateGlobalSettings。

update-phone-number-settings

以下程式碼範例顯示如何使用 update-phone-number-settings。

AWS CLI

更新外撥通話名稱

下列update-phone-number-settings範例會更新管理員 AWS 帳戶的預設外撥通話名稱。

```
aws chime update-phone-number-settings \
    --calling-name "myName"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的使用電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePhoneNumberSettings。

update-phone-number

以下程式碼範例顯示如何使用 update-phone-number。

AWS CLI

範例 1：更新電話號碼的產品類型

下列update-phone-number範例會更新指定電話號碼的產品類型。

```
aws chime update-phone-number \
    --phone-number-id "+12065550100" \
    --product-type "BusinessCalling"
```

輸出：

```
{
    "PhoneNumber": {
        "PhoneNumberId": "%2B12065550100",
        "E164PhoneNumber": "+12065550100",
        "Type": "Local",
        "ProductType": "BusinessCalling",
        "Status": "Unassigned",
        "Capabilities": {
            "InboundCall": true,
            "OutboundCall": true,
            "InboundSMS": true,
            "OutboundSMS": true,
```

```
            "InboundMMS": true,
            "OutboundMMS": true
        },
        "Associations": [],
        "CallingName": "phonenumber1",
        "CreatedTimestamp": "2019-08-09T21:35:21.445Z",
        "UpdatedTimestamp": "2019-08-12T21:44:07.591Z"
    }
}
```

範例 2：更新電話號碼的外撥通話名稱

下列update-phone-number範例會更新指定電話號碼的外撥通話名稱。

aws chime update-phone-number --phone-number-id "+12065550100" --calling-name "phonenumber2"

輸出：

```
{
    "PhoneNumber": {
        "PhoneNumberId": "%2B12065550100",
        "E164PhoneNumber": "+12065550100",
        "Type": "Local",
        "ProductType": "BusinessCalling",
        "Status": "Unassigned",
        "Capabilities": {
            "InboundCall": true,
            "OutboundCall": true,
            "InboundSMS": true,
            "OutboundSMS": true,
            "InboundMMS": true,
            "OutboundMMS": true
        },
        "Associations": [],
        "CallingName": "phonenumber2",
        "CreatedTimestamp": "2019-08-09T21:35:21.445Z",
        "UpdatedTimestamp": "2019-08-12T21:44:07.591Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的使用電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePhoneNumber。

update-room-membership

以下程式碼範例顯示如何使用 update-room-membership。

AWS CLI

更新房間成員資格

下列update-room-membership範例會將指定聊天室成員的角色修改為 Administrator。

```
aws chime update-room-membership \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \
    --member-id 1ab2345c-67de-8901-f23g-45h678901j2k \
    --role Administrator
```

輸出：

```
{
    "RoomMembership": {
        "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
        "Member": {
            "MemberId": "1ab2345c-67de-8901-f23g-45h678901j2k",
            "MemberType": "User",
            "Email": "sofiamartinez@example.com",
            "FullName": "Sofia Martinez",
            "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45"
        },
        "Role": "Administrator",
        "InvitedBy": "arn:aws:iam::111122223333:user/admin",
        "UpdatedTimestamp": "2019-12-02T22:40:22.931Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 使用者指南》中的建立聊天室。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRoomMembership。

update-room

以下程式碼範例顯示如何使用 update-room。

AWS CLI

更新聊天室

下列update-room範例會修改指定聊天室的名稱。

```
aws chime update-room \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \
    --name teamRoom
```

輸出：

```
{
    "Room": {
        "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
        "Name": "teamRoom",
        "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",
        "CreatedBy": "arn:aws:iam::111122223333:user/alejandro",
        "CreatedTimestamp": "2019-12-02T22:29:31.549Z",
        "UpdatedTimestamp": "2019-12-02T22:33:19.310Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Chime 使用者指南》中的建立聊天室。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRoom。

update-user-settings

以下程式碼範例顯示如何使用 update-user-settings。

AWS CLI

更新使用者設定

下列update-user-settings範例可讓指定的使用者進行傳入和傳出通話，以及傳送和接收簡訊。

```
aws chime update-user-settings \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
```

```
    --user-id 1ab2345c-67de-8901-f23g-45h678901j2k \
    --user-settings "Telephony={InboundCalling=true,OutboundCalling=true,SMS=true}"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Chime 管理指南》中的管理使用者電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateUserSettings。

update-user

以下程式碼範例顯示如何使用 update-user。

AWS CLI

更新使用者詳細資訊

此範例會更新指定使用者的指定詳細資訊。

命令：

```
aws chime update-user \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE \
    --license-type "Basic"
```

輸出：

```
{
    "User": {
        "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateUser。

使用 的雲端控制 API 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Cloud Control API 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-resource`

以下程式碼範例顯示如何使用 `create-resource`。

AWS CLI

建立資源

下列`create-resource`範例會建立名為 ResourceExample 的 AWS：：Kinesis：：Stream 資源，其保留期間為 168 小時，碎片計數為 3。

```
aws cloudcontrol create-resource \
    --type-name AWS::Kinesis::Stream \
    --desired-state "{\"Name\": \"ResourceExample\",\"RetentionPeriodHours\":168,
 \"ShardCount\":3}"
```

輸出：

```
{
    "ProgressEvent": {
        "EventTime": 1632506656.706,
        "TypeName": "AWS::Kinesis::Stream",
        "OperationStatus": "IN_PROGRESS",
        "Operation": "CREATE",
        "Identifier": "ResourceExample",
        "RequestToken": "20999d87-e304-4725-ad84-832dcbfd7fc5"
    }
}
```

如需詳細資訊，請參閱《雲端控制 API 使用者指南》中的建立資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateResource。

delete-resource

以下程式碼範例顯示如何使用 delete-resource。

AWS CLI

刪除資源

下列delete-resource範例會從您的帳戶刪除識別碼為 ResourceExample 的 a AWS：：Kinesis：：Stream 資源 AWS 。

```
aws cloudcontrol delete-resource \
    --type-name AWS::Kinesis::Stream \
    --identifier ResourceExample
```

輸出：

```
{
    "ProgressEvent": {
        "TypeName": "AWS::Kinesis::Stream",
        "Identifier": "ResourceExample",
        "RequestToken": "e48f26ff-d0f9-4ab8-a878-120db1edf111",
        "Operation": "DELETE",
        "OperationStatus": "IN_PROGRESS",
        "EventTime": 1632950300.14
    }
}
```

如需詳細資訊，請參閱《雲端控制 API 使用者指南》中的刪除資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteResource。

get-resource-request-status

以下程式碼範例顯示如何使用 get-resource-request-status。

AWS CLI

取得資源請求的狀態資訊

下列get-resource-request-status範例會傳回指定資源請求的狀態資訊。

```
aws cloudcontrol get-resource-request-status \
```

```
    --request-token "e1a6b86e-46bd-41ac-bfba-001234567890"
```

輸出：

```
{
    "ProgressEvent": {
        "TypeName": "AWS::Kinesis::Stream",
        "Identifier": "Demo",
        "RequestToken": "e1a6b86e-46bd-41ac-bfba-001234567890",
        "Operation": "CREATE",
        "OperationStatus": "FAILED",
        "EventTime": 1632950268.481,
        "StatusMessage": "Resource of type 'AWS::Kinesis::Stream' with identifier
 'Demo' already exists.",
        "ErrorCode": "AlreadyExists"
    }
}
```

如需詳細資訊，請參閱《雲端控制 API 使用者指南》中的管理資源操作請求。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetResourceRequestStatus。

get-resource

以下程式碼範例顯示如何使用 get-resource。

AWS CLI

若要取得資源的目前狀態

下列get-resource範例會傳回名為 ResourceExample 的 AWS：：Kinesis：：Stream 資源的目前狀態。

```
aws cloudcontrol get-resource \
    --type-name AWS::Kinesis::Stream \
    --identifier ResourceExample
```

輸出：

```
{
    "TypeName": "AWS::Kinesis::Stream",
```

```
    "ResourceDescription": {
        "Identifier": "ResourceExample",
        "Properties": "{\"Arn\":\"arn:aws:kinesis:us-west-2:099908667365:stream/
ResourceExample\",\"RetentionPeriodHours\":168,\"Name\":\"ResourceExample\",
\"ShardCount\":3}"
    }
}
```

如需詳細資訊，請參閱《 雲端控制 API 使用者指南》中的讀取資源的目前狀態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResource。

list-resource-requests

以下程式碼範例顯示如何使用 `list-resource-requests`。

AWS CLI

列出作用中資源操作請求

下列`list-resource-requests`範例列出您 AWS 帳戶中失敗的 CREATE 和 UPDATE 操作的資源請求。

```
aws cloudcontrol list-resource-requests \
    --resource-request-status-filter Operations=CREATE,OperationStatuses=FAILED
```

輸出：

```
{
    "ResourceRequestStatusSummaries": [
        {
            "TypeName": "AWS::Kinesis::Stream",
            "Identifier": "Demo",
            "RequestToken": "e1a6b86e-46bd-41ac-bfba-633abcdfdbd7",
            "Operation": "CREATE",
            "OperationStatus": "FAILED",
            "EventTime": 1632950268.481,
            "StatusMessage": "Resource of type 'AWS::Kinesis::Stream' with
 identifier 'Demo' already exists.",
            "ErrorCode": "AlreadyExists"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Cloud Control API 使用者指南》中的管理資源操作請求。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourceRequests。

`list-resources`

以下程式碼範例顯示如何使用 `list-resources`。

AWS CLI

列出指定類型的資源

下列`list-resources`範例列出您 AWS 帳戶中佈建的 AWS：：Kinesis：：Stream 資源。

```
aws cloudcontrol list-resources \
    --type-name AWS::Kinesis::Stream
```

輸出：

```
{
    "TypeName": "AWS::Kinesis::Stream",
    "ResourceDescriptions": [
        {
            "Identifier": "MyKinesisStream",
            "Properties": "{\"Name\":\"MyKinesisStream\"}"
        },
        {
            "Identifier": "AnotherStream",
            "Properties": "{\"Name\":\"AnotherStream\"}"
        }
    ]
}
```

如需詳細資訊，請參閱《 雲端控制 API 使用者指南》中的探索資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResources。

`update-resource`

以下程式碼範例顯示如何使用 `update-resource`。

AWS CLI

更新現有資源的屬性

下列update-resource範例會將名為 ExampleLogGroup 之 AWS：：Logs：：LogGroup 資源的保留政策更新為 90 天。 ExampleLogGroup

```
aws cloudcontrol update-resource \
    --type-name AWS::Logs::LogGroup \
    --identifier ExampleLogGroup \
    --patch-document "[{\"op\":\"replace\",\"path\":\"/RetentionInDays\",\"value
\":90}]"
```

輸出：

```
{
    "ProgressEvent": {
        "EventTime": "2021-08-09T18:17:15.219Z",
        "TypeName": "AWS::Logs::LogGroup",
        "OperationStatus": "IN_PROGRESS",
        "Operation": "UPDATE",
        "Identifier": "ExampleLogGroup",
        "RequestToken": "5f40c577-3534-4b20-9599-0b0123456789"
    }
}
```

如需詳細資訊，請參閱《雲端控制 API 使用者指南》中的更新資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateResource。

AWS Cloud Map 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS Cloud Map。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-http-namespace`

以下程式碼範例顯示如何使用 `create-http-namespace`。

AWS CLI

建立 HTTP 命名空間

下列`create-http-namespace`範例會建立 HTTP 命名空間 `example.com`。

```
aws servicediscovery create-http-namespace \
    --name example.com \
    --creator-request-id example-request-id
```

輸出：

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

若要確認操作成功，您可以執行 `get-operation`。如需詳細資訊，請參閱 get-operation 。

如需建立命名空間的詳細資訊，請參閱《AWS Cloud Map 開發人員指南》中的建立 Cloud Map 命名空間以將應用程式服務分組。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateHttpNamespace。

`create-private-dns-namespace`

以下程式碼範例顯示如何使用 `create-private-dns-namespace`。

AWS CLI

建立私有 DNS 命名空間

下列`create-private-dns-namespace`範例會建立私有 DNS 命名空間。

```
aws servicediscovery create-private-dns-namespace \
```

```
    --name example.com \
    --vpc vpc-1c56417b
```

輸出：

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

若要確認操作成功，您可以執行 get-operation。如需詳細資訊，請參閱 get-operation 。

如需詳細資訊，請參閱《AWS Cloud Map 開發人員指南》中的建立命名空間。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePrivateDnsNamespace。

create-public-dns-namespace

以下程式碼範例顯示如何使用 create-public-dns-namespace。

AWS CLI

建立公有 DNS 命名空間

下列create-public-dns-namespace範例會建立公有 DNS 命名空間 example.com。

```
aws servicediscovery create-public-dns-namespace \
    --name example-public-dns.com \
    --creator-request-id example-public-request-id \
    --properties DnsProperties={SOA={TTL=60}}
```

輸出：

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

若要確認操作成功，您可以執行 get-operation。

如需建立命名空間的詳細資訊，請參閱《AWS Cloud Map 開發人員指南》中的建立 Cloud Map 命名空間以將應用程式服務分組。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePublicDnsNamespace。

create-service

以下程式碼範例顯示如何使用 create-service。

AWS CLI

建立服務

下列create-service範例會建立 服務。

```
aws servicediscovery create-service \
    --name myservice \
    --namespace-id  ns-ylexjili4cdxy3xm \
    --dns-config "NamespaceId=ns-
ylexjili4cdxy3xm,RoutingPolicy=MULTIVALUE,DnsRecords=[{Type=A,TTL=60}]"
```

輸出：

```
{
        "Service": {
        "Id": "srv-p5zdwlg5uvvzjita",
        "Arn": "arn:aws:servicediscovery:us-west-2:803642222207:service/srv-
p5zdwlg5uvvzjita",
        "Name": "myservice",
        "NamespaceId": "ns-ylexjili4cdxy3xm",
        "DnsConfig": {
            "NamespaceId": "ns-ylexjili4cdxy3xm",
            "RoutingPolicy": "MULTIVALUE",
            "DnsRecords": [
                {
                    "Type": "A",
                    "TTL": 60
                }
            ]
        },
        "CreateDate": 1587081768.334,
        "CreatorRequestId": "567c1193-6b00-4308-bd57-ad38a8822d25"
    }
}
```

如需詳細資訊，請參閱《AWS Cloud Map 開發人員指南》中的建立服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateService。

delete-namespace

以下程式碼範例顯示如何使用 delete-namespace。

AWS CLI

刪除命名空間

下列delete-namespace範例會刪除命名空間。

```
aws servicediscovery delete-namespace \
    --id ns-ylexjili4cdxy3xm
```

輸出：

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k98y6drk"
}
```

若要確認操作成功，您可以執行 get-operation。如需詳細資訊，請參閱 get-operation 。

如需詳細資訊，請參閱《AWS Cloud Map 開發人員指南》中的刪除命名空間。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNamespace。

delete-service-attributes

以下程式碼範例顯示如何使用 delete-service-attributes。

AWS CLI

刪除服務屬性

下列delete-service-attributes範例會刪除具有與指定服務Port相關聯之金鑰的服務屬性。

```
aws servicediscovery delete-service-attributes \
    --service-id srv-e4anhexample0004 \
    --attributes Port
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Cloud Map 開發人員指南》中的刪除命名空間。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteServiceAttributes。

delete-service

以下程式碼範例顯示如何使用 delete-service。

AWS CLI

刪除服務

下列delete-service範例會刪除服務。

```
aws servicediscovery delete-service \
    --id srv-p5zdwlg5uvvzjita
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Cloud Map 開發人員指南》中的刪除服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteService。

deregister-instance

以下程式碼範例顯示如何使用 deregister-instance。

AWS CLI

取消註冊服務執行個體

下列deregister-instance範例會取消註冊服務執行個體。

```
aws servicediscovery deregister-instance \
    --service-id srv-p5zdwlg5uvvzjita \
    --instance-id myservice-53
```

輸出：

```
{
    "OperationId": "4yejorelbukcjzpnr6tlmrghsjwpngf4-k98rnaiq"
}
```

若要確認操作成功，您可以執行 get-operation。如需詳細資訊，請參閱 get-operation 。

如需詳細資訊，請參閱《AWS Cloud Map 開發人員指南》中的取消註冊服務執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterInstance。

discover-instances-revision

以下程式碼範例顯示如何使用 discover-instances-revision。

AWS CLI

探索執行個體的修訂

下列discover-instances-revision範例會探索執行個體不斷增加的修訂。

```
aws servicediscovery discover-instances-revision \
    --namespace-name example.com \
    --service-name myservice
```

輸出：

```
{
    "InstancesRevision": 123456
}
```

如需詳細資訊，請參閱AWS 《Cloud Map 開發人員指南》中的 Cloud Map 服務執行個體。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DiscoverInstancesRevision。

discover-instances

以下程式碼範例顯示如何使用 discover-instances。

AWS CLI

探索已註冊的執行個體

下列discover-instances範例會探索已註冊的執行個體。

```
aws servicediscovery discover-instances \
    --namespace-name example.com \
    --service-name myservice \
    --max-results 10 \
    --health-status ALL
```

輸出：

```
{
    "Instances": [
        {
            "InstanceId": "myservice-53",
            "NamespaceName": "example.com",
            "ServiceName": "myservice",
            "HealthStatus": "UNKNOWN",
            "Attributes": {
                "AWS_INSTANCE_IPV4": "172.2.1.3",
                "AWS_INSTANCE_PORT": "808"
            }
        }
    ]
}
```

如需詳細資訊，請參閱AWS 《Cloud Map 開發人員指南》中的 Cloud Map 服務執行個體。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DiscoverInstances。

get-instance

以下程式碼範例顯示如何使用 get-instance。

AWS CLI

取得執行個體的詳細資訊

下列get-instance範例會取得服務的屬性。

```
aws servicediscovery get-instance \
    --service-id srv-e4anhexample0004
    --instance-id i-abcd1234
```

輸出：

```
{
    "Instances": {
        "Id": "arn:aws:servicediscovery:us-west-2:111122223333;:service/srv-
e4anhexample0004",
        "Attributes": {
```

```
            "AWS_INSTANCE_IPV4": "192.0.2.44",
            "AWS_INSTANCE_PORT": "80",
            "color": "green",
            "region": "us-west-2",
            "stage": "beta"
        }
    }
}
```

如需詳細資訊，請參閱AWS 《Cloud Map 開發人員指南》中的 Cloud Map 服務執行個體。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInstance。

get-instances-health-status

以下程式碼範例顯示如何使用 get-instances-health-status。

AWS CLI

取得與服務相關聯之執行個體的運作狀態

下列get-instances-health-status範例會取得與指定服務相關聯之執行個體的運作狀態。

```
aws servicediscovery get-instances-health-status \
    --service-id srv-e4anhexample0004
```

輸出：

```
{
    "Status": {
        "i-abcd1234": "HEALTHY",
        "i-abcd1235": "UNHEALTHY"
    }
}
```

如需詳細資訊，請參閱AWS 《Cloud Map 開發人員指南》中的 Cloud Map 服務執行個體。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInstancesHealthStatus。

get-namespace

以下程式碼範例顯示如何使用 get-namespace。

AWS CLI

取得命名空間的詳細資訊

下列get-namespace範例會擷取指定命名空間的相關資訊。

```
aws servicediscovery get-namespace \
    --id ns-e4anhexample0004
```

輸出：

```
{
    "Namespaces": {
        "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
e4anhexample0004",
        "CreateDate": "20181118T211712Z",
        "CreatorRequestId": "example-creator-request-id-0001",
        "Description": "Example.com AWS Cloud Map HTTP Namespace",
        "Id": "ns-e4anhexample0004",
        "Name": "example-http.com",
        "Properties": {
            "DnsProperties": {},
            "HttpProperties": {
                "HttpName": "example-http.com"
            }
        },
        "Type": "HTTP"
    }
}
```

如需詳細資訊，請參閱AWS 《Cloud Map 開發人員指南》中的 Cloud Map 命名空間。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetNamespace。

get-operation

以下程式碼範例顯示如何使用 get-operation。

AWS CLI

取得 操作的結果

下列get-operation範例取得命名空間建立操作的結果。

```
aws servicediscovery get-operation \
    --operation-id gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd
```

輸出：

```
{
    "Operation": {
        "Id": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd",
        "Type": "CREATE_NAMESPACE",
        "Status": "SUCCESS",
        "CreateDate": 1587055860.121,
        "UpdateDate": 1587055900.469,
        "Targets": {
            "NAMESPACE": "ns-ylexjili4cdxy3xm"
        }
    }
}
```

如需詳細資訊，請參閱《AWS Cloud Map 開發人員指南》中的建立 Cloud Map 命名空間以將應用程式服務分組。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetOperation。

get-service-attributes

以下程式碼範例顯示如何使用 get-service-attributes。

AWS CLI

取得 服務的屬性

下列get-service-attributes範例會取得服務的屬性。

```
aws servicediscovery get-service-attributes \
    --service-id srv-e4anhexample0004
```

輸出：

```
{
    "ServiceAttributes": {
        "ServiceArn": "arn:aws:servicediscovery:us-west-2:111122223333;:service/srv-
e4anhexample0004",
```

```
        "Attributes": {
            "Port": "80"
        }
    }
}
```

如需詳細資訊，請參閱AWS 《Cloud Map 開發人員指南》中的 Cloud Map 服務。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetServiceAttributes。

get-service

以下程式碼範例顯示如何使用 get-service。

AWS CLI

取得服務的設定

下列get-service範例會取得指定服務的設定。

```
aws servicediscovery get-service \
    --id srv-e4anhexample0004
```

輸出：

```
{
    "Service": {
        "Id": "srv-e4anhexample0004",
        "Arn": "arn:aws:servicediscovery:us-west-2:111122223333:service/srv-
e4anhexample0004",
        "Name": "test-service",
        "NamespaceId": "ns-e4anhexample0004",
        "DnsConfig": {},
        "Type": "HTTP",
        "CreateDate": "2025-02-24T10:59:02.905000-06:00",
        "CreatorRequestId": "3f50f9d9-b14c-482e-a556-d2a22fe6106d"
    }
}
```

如需詳細資訊，請參閱AWS 《Cloud Map 開發人員指南》中的 Cloud Map 服務。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetService。

list-instances

以下程式碼範例顯示如何使用 list-instances。

AWS CLI

列出服務執行個體

下列list-instances範例列出服務執行個體。

```
aws servicediscovery list-instances \
    --service-id srv-qzpwvt2tfqcegapy
```

輸出：

```
{
    "Instances": [
        {
            "Id": "i-06bdabbae60f65a4e",
            "Attributes": {
                "AWS_INSTANCE_IPV4": "172.2.1.3",
                "AWS_INSTANCE_PORT": "808"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Cloud Map 開發人員指南》中的檢視服務執行個體清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListInstances。

list-namespaces

以下程式碼範例顯示如何使用 list-namespaces。

AWS CLI

列出命名空間

下列list-namespaces範例列出命名空間。

```
aws servicediscovery list-namespaces
```

輸出：

```
{
    "Namespaces": [
        {
            "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
a3ccy2e7e3a7rile",
            "CreateDate": 1585354387.357,
            "Id": "ns-a3ccy2e7e3a7rile",
            "Name": "local",
            "Properties": {
                "DnsProperties": {
                    "HostedZoneId": "Z06752353VBUDTC32S84S"
                },
                "HttpProperties": {
                    "HttpName": "local"
                 }
            },
            "Type": "DNS_PRIVATE"
        },
        {
            "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
pocfyjtrsmwtvcxx",
            "CreateDate": 1586468974.698,
            "Description": "My second namespace",
            "Id": "ns-pocfyjtrsmwtvcxx",
            "Name": "My-second-namespace",
            "Properties": {
                "DnsProperties": {},
                "HttpProperties": {
                    "HttpName": "My-second-namespace"
                }
            },
            "Type": "HTTP"
        },
        {
            "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
ylexjili4cdxy3xm",
            "CreateDate": 1587055896.798,
            "Id": "ns-ylexjili4cdxy3xm",
            "Name": "example.com",
            "Properties": {
                "DnsProperties": {
                    "HostedZoneId": "Z09983722P0QME1B3KC8I"
```

```
                },
                 "HttpProperties": {
                     "HttpName": "example.com"
                }
            },
            "Type": "DNS_PRIVATE"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Cloud Map 開發人員指南》中的檢視命名空間清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListNamespaces。

`list-operations`

以下程式碼範例顯示如何使用 `list-operations`。

AWS CLI

列出符合指定條件的操作

下列`list-operations`範例列出狀態為 PENDING或 的操作SUCCESS。

```
aws servicediscovery list-operations \
    --service-id srv-e4anhexample0004 \
    --filters Name=STATUS,Condition=IN,Values=PENDING,SUCCESS
```

輸出：

```
{
    "Operations": [
        {
            "Id": "76yy8ovhpdz0plmjzbsnqgnrqvpv2qdt-kexample",
            "Status": "SUCCESS"
        },
        {
            "Id": "prysnyzpji3u2ciy45nke83x2zanl7yk-dexample",
            "Status": "SUCCESS"
        },
        {
            "Id": "ko4ekftir7kzlbechsh7xvcdgcpk66gh-7example",
```

```
            "Status": "PENDING"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Cloud Map 開發人員指南》中的什麼是 AWS Cloud Map？。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListOperations。

list-services

以下程式碼範例顯示如何使用 list-services。

AWS CLI

列出服務

下列list-services範例列出 服務。

```
aws servicediscovery list-services
```

輸出：

```
{
    "Services": [
        {
            "Id": "srv-p5zdwlg5uvvzjita",
            "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:service/srv-
p5zdwlg5uvvzjita",
            "Name": "myservice",
            "DnsConfig": {
                "RoutingPolicy": "MULTIVALUE",
                "DnsRecords": [
                    {
                        "Type": "A",
                        "TTL": 60
                    }
                ]
            },
            "CreateDate": 1587081768.334
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《AWS Cloud Map 開發人員指南》中的檢視服務清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListServices。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出與指定資源相關聯的標籤

下列`list-tags-for-resource`範例列出指定資源的標籤。

```
aws servicediscovery list-tags-for-resource \
    --resource-arn arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-e4anhexample0004
```

輸出：

```
{
    "Tags": [
        {
            "Key": "Project",
            "Value": "Zeta"
        },
        {
            "Key": "Department",
            "Value": "Engineering"
        }
    ]
}
```

如需詳細資訊，請參閱AWS 《Cloud Map 開發人員指南》中的標記您的 Cloud Map 資源。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`register-instance`

以下程式碼範例顯示如何使用 `register-instance`。

AWS CLI

註冊服務執行個體

下列register-instance範例會註冊服務執行個體。

```
aws servicediscovery register-instance \
    --service-id srv-p5zdwlg5uvvzjita \
    --instance-id myservice-53 \
    --attributes=AWS_INSTANCE_IPV4=172.2.1.3,AWS_INSTANCE_PORT=808
```

輸出：

```
{
    "OperationId": "4yejorelbukcjzpnr6tlmrghsjwpngf4-k95yg2u7"
}
```

若要確認操作成功，您可以執行 get-operation。如需詳細資訊，請參閱 get-operation 。

如需詳細資訊，請參閱《AWS Cloud Map 開發人員指南》中的註冊執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterInstance。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤與指定的資源建立關聯

下列tag-resource範例會將Department標籤與 值Engineering與指定的命名空間建立關聯。

```
aws servicediscovery tag-resource \
    --resource-arn arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
e4anhexample0004 \
    --tags Key=Department, Value=Engineering
```

此命令不會產生輸出。

如需詳細資訊，請參閱AWS 《Cloud Map 開發人員指南》中的標記您的 Cloud Map 資源。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從指定的資源移除標籤

下列untag-resource範例會從指定的命名空間移除Department標籤。

```
aws servicediscovery untag-resource \
    --resource-arn arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-e4anhexample0004 \
    --tags Key=Department, Value=Engineering
```

此命令不會產生輸出。

如需詳細資訊，請參閱AWS 《Cloud Map 開發人員指南》中的標記您的 Cloud Map 資源。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-http-namespace

以下程式碼範例顯示如何使用 update-http-namespace。

AWS CLI

更新 HTTP 命名空間

下列update-http-namespace範例會更新指定的 HTTP 命名空間描述。

```
aws servicediscovery update-http-namespace \
    --id ns-vh4nbmEXAMPLE \
    --updater-request-id example-request-id \
    --namespace Description="The updated namespace description."
```

輸出：

```
{
```

```
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

若要確認操作成功，您可以執行 get-operation。如需詳細資訊，請參閱 get-operation 。

如需詳細資訊，請參閱AWS 《Cloud Map 開發人員指南》中的 Cloud Map 命名空間。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateHttpNamespace。

update-instance-custom-health-status

以下程式碼範例顯示如何使用 update-instance-custom-health-status。

AWS CLI

更新自訂運作狀態檢查

下列update-instance-custom-health-status範例會將指定服務和範例服務執行個體的自訂運作狀態檢查狀態更新為 HEALTHY。

```
aws servicediscovery update-instance-custom-health-status \
    --service-id srv-e4anhexample0004 \
    --instance-id example \
    --status HEALTHY
```

此命令不會產生輸出。

如需詳細資訊，請參閱AWS 《Cloud Map 開發人員指南》中的 Cloud Map 服務運作狀態檢查組態。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateInstanceCustomHealthStatus。

update-private-dns-namespace

以下程式碼範例顯示如何使用 update-private-dns-namespace。

AWS CLI

更新私有 DNS 命名空間

下列update-private-dns-namespace範例會更新私有 DNS 命名空間的描述。

```
aws servicediscovery update-private-dns-namespace \
    --id ns-bk3aEXAMPLE \
    --updater-request-id example-private-request-id \
    --namespace Description="The updated namespace description."
```

輸出：

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

若要確認操作成功，您可以執行 get-operation。

如需詳細資訊，請參閱AWS 《Cloud Map 開發人員指南》中的 Cloud Map 命名空間。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePrivateDnsNamespace。

update-public-dns-namespace

以下程式碼範例顯示如何使用 update-public-dns-namespace。

AWS CLI

更新公有 DNS 命名空間

下列update-public-dns-namespace範例會更新公有 DNS 命名空間的描述。

```
aws servicediscovery update-public-dns-namespace \
    --id ns-bk3aEXAMPLE \
    --updater-request-id example-public-request-id \
    --namespace Description="The updated namespace description."
```

輸出：

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

若要確認操作成功，您可以執行 get-operation。

如需詳細資訊，請參閱AWS 《Cloud Map 開發人員指南》中的 Cloud Map 命名空間。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePublicDnsNamespace。

update-service-attributes

以下程式碼範例顯示如何使用 update-service-attributes。

AWS CLI

更新服務以新增屬性

下列update-service-attributes範例會更新指定的服務，以新增具有索引鍵 Port和值 的服務屬性80。

```
aws servicediscovery update-service-attributes \
    --service-id srv-e4anhexample0004 \
    --attributes Port=80
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Cloud Map 開發人員指南》中的 Cloud Map 服務。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateServiceAttributes。

update-service

以下程式碼範例顯示如何使用 update-service。

AWS CLI

更新服務

下列update-service範例會更新服務以更新 DnsConfig和 HealthCheckConfig設定。

```
aws servicediscovery update-service \
    --id srv-e4anhexample0004 \
    --service
 "DnsConfig={DnsRecords=[{"Type"="A","TTL"=60}]},HealthCheckConfig={"Type"="HTTP","ResourceP
```

輸出：

```
{
```

```
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

若要確認操作成功，您可以執行 `get-operation`。

如需更新服務的詳細資訊，請參閱《AWS Cloud Map 開發人員指南》中的更新 Cloud Map 服務。AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateService。

AWS Cloud9 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS Cloud9。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-environment-ec2`

以下程式碼範例顯示如何使用 `create-environment-ec2`。

AWS CLI

建立 an AWS Cloud9 EC2 開發環境

以下`create-environment-ec2`範例會使用指定的設定建立 an AWS Cloud9 開發環境、啟動 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體，然後從執行個體連線至環境。

```
aws cloud9 create-environment-ec2 \
    --name my-demo-env \
    --description "My demonstration development environment." \
    --instance-type t2.micro --image-id amazonlinux-2023-x86_64 \
    --subnet-id subnet-1fab8aEX \
```

```
    --automatic-stop-time-minutes 60 \
    --owner-arn arn:aws:iam::123456789012:user/MyDemoUser
```

輸出：

```
{
    "environmentId": "8a34f51ce1e04a08882f1e811bd706EX"
}
```

如需詳細資訊，請參閱《AWS Cloud9 使用者指南》中的建立 EC2 環境。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateEnvironmentEc2。

`create-environment-membership`

以下程式碼範例顯示如何使用 `create-environment-membership`。

AWS CLI

將環境成員新增至 an AWS Cloud9 開發環境

此範例會將指定的環境成員新增至指定的 AWS Cloud9 開發環境。

命令：

```
aws cloud9 create-environment-membership --environment-
id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/
AnotherDemoUser --permissions read-write
```

輸出：

```
{
  "membership": {
    "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
    "userId": "AIDAJ3LOROMOUXTBSU6EX",
    "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",
    "permissions": "read-write"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateEnvironmentMembership。

delete-environment-membership

以下程式碼範例顯示如何使用 delete-environment-membership。

AWS CLI

從 an AWS Cloud9 開發環境刪除環境成員

此範例會從指定的 AWS Cloud9 開發環境刪除指定的環境成員。

命令：

```
aws cloud9 delete-environment-membership --environment-
id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/
AnotherDemoUser
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteEnvironmentMembership。

delete-environment

以下程式碼範例顯示如何使用 delete-environment。

AWS CLI

刪除 an AWS Cloud9 開發環境

此範例會刪除指定的 AWS Cloud9 開發環境。如果 Amazon EC2 執行個體連線到環境， 也會終止執行個體。

命令：

```
aws cloud9 delete-environment --environment-id 8a34f51ce1e04a08882f1e811bd706EX
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteEnvironment。

describe-environment-memberships

以下程式碼範例顯示如何使用 describe-environment-memberships。

AWS CLI

取得有關 an AWS Cloud9 開發環境的環境成員資訊

此範例會取得指定 AWS Cloud9 開發環境的環境成員相關資訊。

命令：

```
aws cloud9 describe-environment-memberships --environment-
id 8a34f51ce1e04a08882f1e811bd706EX
```

輸出：

```
{
  "memberships": [
    {
      "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
      "userId": "AIDAJ3LOROMOUXTBSU6EX",
      "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",
      "permissions": "read-write"
    },
    {
      "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
      "userId": "AIDAJNUEDQAQWFELJDLEX",
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "permissions": "owner"
    }
  ]
}
```

取得有關 an AWS Cloud9 開發環境擁有者的資訊

此範例會取得指定 AWS Cloud9 開發環境擁有者的相關資訊。

命令：

```
aws cloud9 describe-environment-memberships --environment-
id 8a34f51ce1e04a08882f1e811bd706EX --permissions owner
```

輸出：

```
{
  "memberships": [
    {
      "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
      "userId": "AIDAJNUEDQAQWFELJDLEX",
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "permissions": "owner"
    }
  ]
}
```

取得多個 AWS Cloud9 開發環境的環境成員相關資訊

此範例會取得多個 AWS Cloud9 開發環境之指定環境成員的相關資訊。

命令：

```
aws cloud9 describe-environment-memberships --user-
arn arn:aws:iam::123456789012:user/MyDemoUser
```

輸出：

```
{
  "memberships": [
    {
      "environmentId": "10a75714bd494714929e7f5ec4125aEX",
      "lastAccess": 1516213427.0,
      "userId": "AIDAJNUEDQAQWFELJDLEX",
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "permissions": "owner"
    },
    {
      "environmentId": "1980b80e5f584920801c09086667f0EX",
      "lastAccess": 1516144884.0,
      "userId": "AIDAJNUEDQAQWFELJDLEX",
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "permissions": "owner"
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEnvironmentMemberships。

describe-environment-status

以下程式碼範例顯示如何使用 describe-environment-status。

AWS CLI

取得 an AWS Cloud9 開發環境的狀態資訊

此範例會取得指定 AWS Cloud9 開發環境的狀態資訊。

命令：

```
aws cloud9 describe-environment-status --environment-
id 685f892f431b45c2b28cb69eadcdb0EX
```

輸出：

```
{
  "status": "ready",
  "message": "Environment is ready to use"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEnvironmentStatus。

describe-environments

以下程式碼範例顯示如何使用 describe-environments。

AWS CLI

取得 AWS Cloud9 開發環境的相關資訊

此範例會取得指定 AWS Cloud9 開發環境的相關資訊。

命令：

```
aws cloud9 describe-environments --environment-
ids 685f892f431b45c2b28cb69eadcdb0EX 349c86d4579e4e7298d500ff57a6b2EX
```

輸出：

```
{
  "environments": [
    {
      "id": "685f892f431b45c2b28cb69eadcdb0EX",
      "name": "my-demo-ec2-env",
      "description": "Created from CodeStar.",
      "type": "ec2",
      "arn": "arn:aws:cloud9:us-
east-1:123456789012:environment:685f892f431b45c2b28cb69eadcdb0EX",
      "ownerArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "lifecycle": {
        "status": "CREATED"
      }
    },
    {
      "id": "349c86d4579e4e7298d500ff57a6b2EX",
      "name": my-demo-ssh-env",
      "description": "",
      "type": "ssh",
      "arn": "arn:aws:cloud9:us-
east-1:123456789012:environment:349c86d4579e4e7298d500ff57a6b2EX",
      "ownerArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "lifecycle": {
        "status": "CREATED"
      }
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEnvironments。

`list-environments`

以下程式碼範例顯示如何使用 `list-environments`。

AWS CLI

取得 available AWS Cloud9 開發環境識別符的清單

此範例會取得 available AWS Cloud9 開發環境識別符的清單。

命令：

```
aws cloud9 list-environments
```

輸出：

```
{
  "environmentIds": [
    "685f892f431b45c2b28cb69eadcdb0EX",
    "1980b80e5f584920801c09086667f0EX"
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEnvironments。

update-environment-membership

以下程式碼範例顯示如何使用 update-environment-membership。

AWS CLI

變更 an AWS Cloud9 開發環境現有環境成員的設定

此範例會變更指定 AWS Cloud9 開發環境之指定現有環境成員的設定。

命令：

```
aws cloud9 update-environment-membership --environment-
id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/
AnotherDemoUser --permissions read-only
```

輸出：

```
{
  "membership": {
    "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
    "userId": "AIDAJ3LOROMOUXTBSU6EX",
    "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",
    "permissions": "read-only"
  }
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateEnvironmentMembership。

`update-environment`

以下程式碼範例顯示如何使用 `update-environment`。

AWS CLI

變更現有 AWS Cloud9 開發環境的設定

此範例會變更指定之 existing AWS Cloud9 開發環境的指定設定。

命令：

```
aws cloud9 update-environment --environment-id 8a34f51ce1e04a08882f1e811bd706EX
 --name my-changed-demo-env --description "My changed demonstration development
 environment."
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateEnvironment。

AWS CloudFormation 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS CloudFormation。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`activate-type`

以下程式碼範例顯示如何使用 `activate-type`。

AWS CLI

啟用類型

下列`activate-type`範例會啟用公有第三方延伸模組，使其可用於堆疊範本。

```
aws cloudformation activate-type \
    --region us-west-2 \
    --type RESOURCE \
    --type-name Example::Test::1234567890abcdef0 \
    --type-name-alias Example::Test::Alias
```

輸出：

```
{
    "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/Example-
Test-Alias"
}
```

如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的使用 CloudFormation 登錄檔。AWS CloudFormation

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ActivateType。

`batch-describe-type-configurations`

以下程式碼範例顯示如何使用 `batch-describe-type-configurations`。

AWS CLI

批次描述類型組態

下列`batch-describe-type-configurations`範例會設定 類型的資料。

```
aws cloudformation batch-describe-type-configurations \
    --region us-west-2 \
```

```
    --type-configuration-identifiers TypeArn="arn:aws:cloudformation:us-
west-2:123456789012:type/resource/Example-Test-
Type,TypeConfigurationAlias=MyConfiguration"
```

輸出：

```
{
    "Errors": [],
    "UnprocessedTypeConfigurations": [],
    "TypeConfigurations": [
        {
            "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/
Example-Test-Type",
            "Alias": "MyConfiguration",
            "Configuration": "{\n        \"Example\": {\n            \"ApiKey\":
 \"examplekey\",\n            \"ApplicationKey\": \"examplekey1\",\n
 \"ApiURL\": \"exampleurl\"\n            }\n}",
            "LastUpdated": "2021-10-01T15:25:46.210000+00:00",
            "TypeArn": "arn:aws:cloudformation:us-east-1:123456789012:type/resource/
Example-Test-Type"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的使用 CloudFormation 登錄檔。AWS CloudFormation

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDescribeTypeConfigurations。

cancel-update-stack

以下程式碼範例顯示如何使用 cancel-update-stack。

AWS CLI

取消進行中的堆疊更新

下列cancel-update-stack命令會取消堆疊上的myteststack堆疊更新：

```
aws cloudformation cancel-update-stack --stack-name myteststack
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelUpdateStack。

continue-update-rollback

以下程式碼範例顯示如何使用 continue-update-rollback。

AWS CLI

重試更新轉返

下列continue-update-rollback範例會從先前失敗的堆疊更新繼續復原操作。

```
aws cloudformation continue-update-rollback \
    --stack-name my-stack
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ContinueUpdateRollback。

create-change-set

以下程式碼範例顯示如何使用 create-change-set。

AWS CLI

建立變更集

下列create-change-set範例會使用 CAPABILITY_IAM功能建立變更集。檔案template.yaml是目前資料夾中的 AWS CloudFormation 範本，定義包含 IAM 資源的堆疊。

```
aws cloudformation create-change-set \
    --stack-name my-application \
    --change-set-name my-change-set \
    --template-body file://template.yaml \
    --capabilities CAPABILITY_IAM
```

輸出：

```
{
    "Id": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-change-set/
bc9555ba-a949-xmpl-bfb8-f41d04ec5784",
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-application/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateChangeSet。

create-stack-instances

以下程式碼範例顯示如何使用 create-stack-instances。

AWS CLI

建立堆疊執行個體

下列create-stack-instances範例會在兩個帳戶和四個區域中建立堆疊集的執行個體。容錯能力設定可確保在所有帳戶和區域中嘗試更新，即使無法建立某些堆疊。

```
aws cloudformation create-stack-instances \
    --stack-set-name my-stack-set \
    --accounts 123456789012 223456789012 \
    --regions us-east-1 us-east-2 us-west-1 us-west-2 \
    --operation-preferences FailureToleranceCount=7
```

輸出：

```
{
    "OperationId": "d7995c31-83c2-xmpl-a3d4-e9ca2811563f"
}
```

若要建立堆疊集，請使用 create-stack-set命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateStackInstances。

create-stack-set

以下程式碼範例顯示如何使用 create-stack-set。

AWS CLI

建立堆疊集

下列create-stack-set範例會使用指定的 YAML 檔案樣板建立堆疊集。 template.yaml是定義堆疊之目前資料夾中的 AWS CloudFormation 範本。

```
aws cloudformation create-stack-set \
    --stack-set-name my-stack-set \
```

```
    --template-body file://template.yaml \
    --description "SNS topic"
```

輸出：

```
{
    "StackSetId": "my-stack-set:8d0f160b-d157-xmpl-a8e6-c0ce8e5d8cc1"
}
```

若要將堆疊執行個體新增至堆疊集，請使用 create-stack-instances命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateStackSet。

create-stack

以下程式碼範例顯示如何使用 create-stack。

AWS CLI

建立 an AWS CloudFormation 堆疊

下列create-stacks命令myteststack會使用 sampletemplate.json 範本建立名稱為 的堆疊：

```
aws cloudformation create-stack --stack-name myteststack --template-body file://
sampletemplate.json --parameters ParameterKey=KeyPairName,ParameterValue=TestKey
 ParameterKey=SubnetIDs,ParameterValue=SubnetID1\\,SubnetID2
```

輸出：

```
{
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

如需詳細資訊，請參閱 AWS CloudFormation 使用者指南中的 Stacks。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateStack。

deactivate-type

以下程式碼範例顯示如何使用 deactivate-type。

AWS CLI

停用類型

下列deactivate-type範例會停用先前在此帳戶和區域中啟用的公有擴充功能。

```
aws cloudformation deactivate-type \
    --region us-west-2 \
    --type MODULE \
    --type-name Example::Test::Type::MODULE
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的使用 CloudFormation 登錄檔。AWS CloudFormation

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeactivateType。

delete-change-set

以下程式碼範例顯示如何使用 delete-change-set。

AWS CLI

刪除變更集

下列delete-change-set範例會指定變更集名稱和堆疊名稱，以刪除變更集。

```
aws cloudformation delete-change-set \
    --stack-name my-stack \
    --change-set-name my-change-set
```

此命令不會產生輸出。

下列delete-change-set範例會透過指定變更集的完整 ARN 來刪除變更集。

```
aws cloudformation delete-change-set \
    --change-set-name arn:aws:cloudformation:us-east-2:123456789012:changeSet/my-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteChangeSet。

delete-stack-instances

以下程式碼範例顯示如何使用 delete-stack-instances。

AWS CLI

刪除堆疊執行個體

下列delete-stack-instances範例會刪除兩個區域中兩個帳戶中設定的堆疊執行個體，並終止堆疊。

```
aws cloudformation delete-stack-instances \
    --stack-set-name my-stack-set \
    --accounts 123456789012 567890123456 \
    --regions us-east-1 us-west-1 \
    --no-retain-stacks
```

輸出：

```
{
    "OperationId": "ad49f10c-fd1d-413f-a20a-8de6e2fa8f27"
}
```

若要刪除空堆疊集，請使用 delete-stack-set命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteStackInstances。

delete-stack-set

以下程式碼範例顯示如何使用 delete-stack-set。

AWS CLI

刪除堆疊集

下列命令會刪除指定的空堆疊集。堆疊集必須是空的。

```
aws cloudformation delete-stack-set \
    --stack-set-name my-stack-set
```

此命令不會產生輸出。

若要從堆疊集刪除執行個體，請使用 delete-stack-instances命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteStackSet。

delete-stack

以下程式碼範例顯示如何使用 delete-stack。

AWS CLI

刪除堆疊

下列delete-stack範例會刪除指定的堆疊。

```
aws cloudformation delete-stack \
    --stack-name my-stack
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteStack。

deploy

以下程式碼範例顯示如何使用 deploy。

AWS CLI

下列命令會將名為 的範本部署template.json至名為 的堆疊my-new-stack：

```
aws cloudformation deploy --template-file /path_to_template/template.json
 --stack-name my-new-stack --parameter-overrides Key1=Value1 Key2=Value2 --
tags Key1=Value1 Key2=Value2
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的部署。

deregister-type

以下程式碼範例顯示如何使用 deregister-type。

AWS CLI

取消註冊類型版本

下列deregister-type範例會將指定的類型版本從 CloudFormation 登錄檔中的作用中使用中移除，使其無法再用於 CloudFormation 操作。

```
aws cloudformation deregister-type \
    --type RESOURCE \
    --type-name My::Logs::LogGroup \
    --version-id 00000002
```

此命令不會產生輸出。

如需詳細資訊，請參閱《CloudFormation 使用者指南》中的使用 CloudFormation 登錄檔。 AWS CloudFormation

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterType。

describe-account-limits

以下程式碼範例顯示如何使用 describe-account-limits。

AWS CLI

取得您帳戶限制的相關資訊

下列命令會擷取目前帳戶的區域限制清單。

```
aws cloudformation describe-account-limits
```

輸出：

```
{
    "AccountLimits": [
        {
            "Name": "StackLimit",
            "Value": 200
        },
        {
            "Name": "StackOutputsLimit",
            "Value": 60
        },
        {
            "Name": "ConcurrentResourcesLimit",
            "Value": 2500
        }
```

```
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAccountLimits。

describe-change-set

以下程式碼範例顯示如何使用 describe-change-set。

AWS CLI

取得變更集的相關資訊

下列describe-change-set範例顯示變更集名稱和堆疊名稱所指定的變更集詳細資訊。

```
aws cloudformation describe-change-set \
    --change-set-name my-change-set \
    --stack-name my-stack
```

下列describe-change-set範例顯示變更集的完整 ARN 所指定變更集的詳細資訊：

```
aws cloudformation describe-change-set \
    --change-set-name arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-change-set/bc9555ba-a949-xmpl-bfb8-f41d04ec5784
```

輸出：

```
{
    "Changes": [
        {
            "Type": "Resource",
            "ResourceChange": {
                "Action": "Modify",
                "LogicalResourceId": "function",
                "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
                "ResourceType": "AWS::Lambda::Function",
                "Replacement": "False",
                "Scope": [
                    "Properties"
                ],
                "Details": [
                    {
```

```
                        "Target": {
                            "Attribute": "Properties",
                            "Name": "Timeout",
                            "RequiresRecreation": "Never"
                        },
                        "Evaluation": "Static",
                        "ChangeSource": "DirectModification"
                    }
                ]
            }
        }
    ],
    "ChangeSetName": "my-change-set",
    "ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-
change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
    "StackName": "my-stack",
    "Description": null,
    "Parameters": null,
    "CreationTime": "2019-10-02T05:20:56.651Z",
    "ExecutionStatus": "AVAILABLE",
    "Status": "CREATE_COMPLETE",
    "StatusReason": null,
    "NotificationARNs": [],
    "RollbackConfiguration": {},
    "Capabilities": [
        "CAPABILITY_IAM"
    ],
    "Tags": null
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeChangeSet。

describe-publisher

以下程式碼範例顯示如何使用 describe-publisher。

AWS CLI

描述發佈者

下列describe-publisher範例會設定發佈者的資訊。

```
aws cloudformation describe-publisher \
    --region us-west-2 \
    --publisher-id 000q6TfUovXsEMmgKowxDZLlwqr2QUsh
```

輸出：

```
{
    "PublisherId": "000q6TfUovXsEMmgKowxDZLlwqr2QUshd2e75c8c",
    "PublisherStatus": "VERIFIED",
    "IdentityProvider": "AWS_Marketplace",
    "PublisherProfile": "https://aws.amazon.com/marketplace/seller-profile?
id=2c5dc1f0-17cd-4259-8e46-822a83gdtegd"
}
```

如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的使用 CloudFormation 登錄檔。AWS CloudFormation

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePublisher。

describe-stack-drift-detection-status

以下程式碼範例顯示如何使用 describe-stack-drift-detection-status。

AWS CLI

檢查偏離偵測操作的狀態

下列describe-stack-drift-detection-status範例顯示偏離偵測操作的狀態。依執行detect-stack-drift命令的 ID 取得 。

```
aws cloudformation describe-stack-drift-detection-status \
    --stack-drift-detection-id 1a229160-e4d9-xmpl-ab67-0a4f93df83d4
```

輸出：

```
{
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
    "StackDriftDetectionId": "1a229160-e4d9-xmpl-ab67-0a4f93df83d4",
    "StackDriftStatus": "DRIFTED",
    "DetectionStatus": "DETECTION_COMPLETE",
    "DriftedStackResourceCount": 1,
```

```
    "Timestamp": "2019-10-02T05:54:30.902Z"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStackDriftDetectionStatus。

describe-stack-events

以下程式碼範例顯示如何使用 describe-stack-events。

AWS CLI

描述堆疊事件

下列describe-stack-events範例顯示指定堆疊的 2 個最近事件。

```
aws cloudformation describe-stack-events \
    --stack-name my-stack \
    --max-items 2

{
    "StackEvents": [
        {
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "EventId": "4e1516d0-e4d6-xmpl-b94f-0a51958a168c",
            "StackName": "my-stack",
            "LogicalResourceId": "my-stack",
            "PhysicalResourceId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "ResourceType": "AWS::CloudFormation::Stack",
            "Timestamp": "2019-10-02T05:34:29.556Z",
            "ResourceStatus": "UPDATE_COMPLETE"
        },
        {
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "EventId": "4dd3c810-e4d6-xmpl-bade-0aaf8b31ab7a",
            "StackName": "my-stack",
            "LogicalResourceId": "my-stack",
            "PhysicalResourceId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "ResourceType": "AWS::CloudFormation::Stack",
            "Timestamp": "2019-10-02T05:34:29.127Z",
```

```
            "ResourceStatus": "UPDATE_COMPLETE_CLEANUP_IN_PROGRESS"
        }
    ],

 "NextToken": "eyJOZXh0VG9XMPLiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStackEvents。

describe-stack-instance

以下程式碼範例顯示如何使用 describe-stack-instance。

AWS CLI

描述堆疊執行個體

下列命令說明指定帳戶和區域中指定堆疊集的執行個體。堆疊集位於目前區域和帳戶中，而執行個體位於帳戶 中的us-west-2區域123456789012。：

```
aws cloudformation describe-stack-instance \
    --stack-set-name my-stack-set \
    --stack-instance-account 123456789012 \
    --stack-instance-region us-west-2
```

輸出：

```
{
    "StackInstance": {
        "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
        "Region": "us-west-2",
        "Account": "123456789012",
        "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/
StackSet-enable-config-e6cac20f-xmpl-46e9-8314-53e0d4591532/4287f9a0-e615-
xmpl-894a-12b31d3117be",
        "ParameterOverrides": [],
        "Status": "OUTDATED",
        "StatusReason": "ResourceLogicalId:ConfigBucket,
 ResourceType:AWS::S3::Bucket, ResourceStatusReason:You have attempted to create
 more buckets than allowed (Service: Amazon S3; Status Code: 400; Error Code:
 TooManyBuckets; Request ID: F7F21CXMPL580224; S3 Extended Request ID: egd/
Fdt89BXMPLyiqbMNljVk55Yqqvi3NYW2nKLUVWhUGEhNfCmZdyj967lhriaG/dWMobSO40o=)."
```

```
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStackInstance。

describe-stack-resource-drifts

以下程式碼範例顯示如何使用 describe-stack-resource-drifts。

AWS CLI

取得偏離堆疊定義之資源的相關資訊

下列命令會顯示指定堆疊之漂移資源的相關資訊。若要啟動偏離偵測，請使用 detect-stack-drift命令：

```
aws cloudformation describe-stack-resource-drifts \
    --stack-name my-stack
```

輸出顯示已修改out-of-band的 AWS Lambda 函數：

```
{
    "StackResourceDrifts": [
        {
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "LogicalResourceId": "function",
            "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
            "ResourceType": "AWS::Lambda::Function",
            "ExpectedProperties": "{\"Description\":\"Write a file to S3.\",
\"Environment\":{\"Variables\":{\"bucket\":\"my-stack-bucket-1vc62xmplgguf
\"}},\"Handler\":\"index.handler\",\"MemorySize\":128,\"Role\":
\"arn:aws:iam::123456789012:role/my-functionRole-HIZXMPLEOM9E\",\"Runtime\":
\"nodejs10.x\",\"Tags\":[{\"Key\":\"lambda:createdBy\",\"Value\":\"SAM\"}],\"Timeout
\":900,\"TracingConfig\":{\"Mode\":\"Active\"}}",
            "ActualProperties": "{\"Description\":\"Write a file to S3.\",
\"Environment\":{\"Variables\":{\"bucket\":\"my-stack-bucket-1vc62xmplgguf
\"}},\"Handler\":\"index.handler\",\"MemorySize\":256,\"Role\":
\"arn:aws:iam::123456789012:role/my-functionRole-HIZXMPLEOM9E\",\"Runtime\":
\"nodejs10.x\",\"Tags\":[{\"Key\":\"lambda:createdBy\",\"Value\":\"SAM\"}],\"Timeout
\":22,\"TracingConfig\":{\"Mode\":\"Active\"}}",
            "PropertyDifferences": [
```

```
                {
                    "PropertyPath": "/MemorySize",
                    "ExpectedValue": "128",
                    "ActualValue": "256",
                    "DifferenceType": "NOT_EQUAL"
                },
                {
                    "PropertyPath": "/Timeout",
                    "ExpectedValue": "900",
                    "ActualValue": "22",
                    "DifferenceType": "NOT_EQUAL"
                }
            ],
            "StackResourceDriftStatus": "MODIFIED",
            "Timestamp": "2019-10-02T05:54:44.064Z"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStackResourceDrifts。

describe-stack-resource

以下程式碼範例顯示如何使用 describe-stack-resource。

AWS CLI

取得堆疊資源的相關資訊

下列describe-stack-resource範例顯示指定堆疊MyFunction中名為 之資源的詳細資訊。

```
aws cloudformation describe-stack-resource \
    --stack-name MyStack \
    --logical-resource-id MyFunction
```

輸出：

```
{
    "StackResourceDetail": {
        "StackName": "MyStack",
        "StackId": "arn:aws:cloudformation:us-east-2:123456789012:stack/MyStack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
```

```
        "LogicalResourceId": "MyFunction",
        "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
        "ResourceType": "AWS::Lambda::Function",
        "LastUpdatedTimestamp": "2019-10-02T05:34:27.989Z",
        "ResourceStatus": "UPDATE_COMPLETE",
        "Metadata": "{}",
        "DriftInformation": {
            "StackResourceDriftStatus": "IN_SYNC"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStackResource。

describe-stack-resources

以下程式碼範例顯示如何使用 describe-stack-resources。

AWS CLI

取得堆疊資源的相關資訊

下列describe-stack-resources範例顯示指定堆疊中資源的詳細資訊。

```
aws cloudformation describe-stack-resources \
    --stack-name my-stack
```

輸出：

```
{
    "StackResources": [
        {
            "StackName": "my-stack",
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "LogicalResourceId": "bucket",
            "PhysicalResourceId": "my-stack-bucket-1vc62xmplgguf",
            "ResourceType": "AWS::S3::Bucket",
            "Timestamp": "2019-10-02T04:34:11.345Z",
            "ResourceStatus": "CREATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
```

```
        },
        {
            "StackName": "my-stack",
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "LogicalResourceId": "function",
            "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
            "ResourceType": "AWS::Lambda::Function",
            "Timestamp": "2019-10-02T05:34:27.989Z",
            "ResourceStatus": "UPDATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        },
        {
            "StackName": "my-stack",
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "LogicalResourceId": "functionRole",
            "PhysicalResourceId": "my-functionRole-HIZXMPLEOM9E",
            "ResourceType": "AWS::IAM::Role",
            "Timestamp": "2019-10-02T04:34:06.350Z",
            "ResourceStatus": "CREATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStackResources。

describe-stack-set-operation

以下程式碼範例顯示如何使用 describe-stack-set-operation。

AWS CLI

取得堆疊集操作的相關資訊

下列 describe-stack-set-operation` 範例顯示指定堆疊集上更新操作的詳細資訊。

```
aws cloudformation describe-stack-set-operation \
```

```
    --stack-set-name enable-config \
    --operation-id 35d45ebc-ed88-xmpl-ab59-0197a1fc83a0
```

輸出：

```
{
    "StackSetOperation": {
        "OperationId": "35d45ebc-ed88-xmpl-ab59-0197a1fc83a0",
        "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
        "Action": "UPDATE",
        "Status": "SUCCEEDED",
        "OperationPreferences": {
            "RegionOrder": [
                "us-east-1",
                "us-west-2",
                "eu-west-1",
                "us-west-1"
            ],
            "FailureToleranceCount": 7,
            "MaxConcurrentCount": 2
        },
        "AdministrationRoleARN": "arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole",
        "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",
        "CreationTimestamp": "2019-10-03T16:28:44.377Z",
        "EndTimestamp": "2019-10-03T16:42:08.607Z"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStackSetOperation。

describe-stack-set

以下程式碼範例顯示如何使用 describe-stack-set。

AWS CLI

取得堆疊集的相關資訊

下列 describe-stack-set` 範例會顯示指定堆疊集的詳細資訊。

```
aws cloudformation describe-stack-set \
```

```
    --stack-set-name my-stack-set
```

輸出：

```
{
    "StackSet": {
        "StackSetName": "my-stack-set",
        "StackSetId": "my-stack-set:296a3360-xmpl-40af-be78-9341e95bf743",
        "Description": "Create an Amazon SNS topic",
        "Status": "ACTIVE",
        "TemplateBody": "AWSTemplateFormatVersion: '2010-09-09'\nDescription: An AWS
 SNS topic\nResources:\n  topic:\n    Type: AWS::SNS::Topic",
        "Parameters": [],
        "Capabilities": [],
        "Tags": [],
        "StackSetARN": "arn:aws:cloudformation:us-west-2:123456789012:stackset/
enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
        "AdministrationRoleARN": "arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole",
        "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStackSet。

describe-stacks

以下程式碼範例顯示如何使用 `describe-stacks`。

AWS CLI

描述 AWS CloudFormation 堆疊

下列`describe-stacks`命令顯示`myteststack`堆疊的摘要資訊：

```
aws cloudformation describe-stacks --stack-name myteststack
```

輸出：

```
{
    "Stacks":  [
```

```
        {
            "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
            "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
 template showing how to create a publicly accessible S3 bucket. **WARNING** This
 template creates an S3 bucket. You will be billed for the AWS resources used if you
 create a stack from this template.",
            "Tags": [],
            "Outputs": [
                {
                    "Description": "Name of S3 bucket to hold website content",
                    "OutputKey": "BucketName",
                    "OutputValue": "myteststack-s3bucket-jssofi1zie2w"
                }
            ],
            "StackStatusReason": null,
            "CreationTime": "2013-08-23T01:02:15.422Z",
            "Capabilities": [],
            "StackName": "myteststack",
            "StackStatus": "CREATE_COMPLETE",
            "DisableRollback": false
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CloudFormation 使用者指南中的 Stacks。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStacks。

describe-type-registration

以下程式碼範例顯示如何使用 describe-type-registration。

AWS CLI

顯示類型註冊資訊

下列describe-type-registration範例顯示指定類型註冊的相關資訊，包括該類型的目前狀態、類型和版本。

```
aws cloudformation describe-type-registration \
    --registration-token a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "ProgressStatus": "COMPLETE",
    "TypeArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Logs-LogGroup",
    "Description": "Deployment is currently in DEPLOY_STAGE of status COMPLETED; ",
    "TypeVersionArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Logs-LogGroup/00000001"
}
```

如需詳細資訊，請參閱《CloudFormation 使用者指南》中的使用 CloudFormation 登錄檔。 AWS CloudFormation

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTypeRegistration。

describe-type

以下程式碼範例顯示如何使用 describe-type。

AWS CLI

顯示類型資訊

下列describe-type範例顯示指定類型的資訊。

```
aws cloudformation describe-type \
    --type-name My::Logs::LogGroup \
    --type RESOURCE
```

輸出：

```
{
    "SourceUrl": "https://github.com/aws-cloudformation/aws-cloudformation-resource-providers-logs.git",
    "Description": "Customized resource derived from AWS::Logs::LogGroup",
    "TimeCreated": "2019-12-03T23:29:33.321Z",
    "Visibility": "PRIVATE",
    "TypeName": "My::Logs::LogGroup",
    "LastUpdated": "2019-12-03T23:29:33.321Z",
    "DeprecatedStatus": "LIVE",
    "ProvisioningType": "FULLY_MUTABLE",
```

```
    "Type": "RESOURCE",
    "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Logs-
LogGroup/00000001",
    "Schema": "[details omitted]"
}
```

如需詳細資訊，請參閱《CloudFormation 使用者指南》中的使用 CloudFormation 登錄檔。 AWS CloudFormation

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeType。

detect-stack-drift

以下程式碼範例顯示如何使用 detect-stack-drift。

AWS CLI

偵測漂移的資源

下列detect-stack-drift範例會啟動指定堆疊的偏離偵測。

```
aws cloudformation detect-stack-drift \
    --stack-name my-stack
```

輸出：

```
{
    "StackDriftDetectionId": "1a229160-e4d9-xmpl-ab67-0a4f93df83d4"
}
```

然後，您可以使用此 ID 搭配 describe-stack-resource-drifts命令來描述漂移的資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetectStackDrift。

detect-stack-resource-drift

以下程式碼範例顯示如何使用 detect-stack-resource-drift。

AWS CLI

偵測資源的偏離

下列detect-stack-resource-drift範例會檢查名為 的堆疊MyFunction中名為 的資源，MyStack以用於偏離：

```
aws cloudformation detect-stack-resource-drift \
   --stack-name MyStack \
   --logical-resource-id MyFunction
```

輸出顯示已修改out-of-band的 AWS Lambda 函數：

```
{
    "StackResourceDrift": {
        "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/MyStack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
        "LogicalResourceId": "MyFunction",
        "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
        "ResourceType": "AWS::Lambda::Function",
        "ExpectedProperties": "{\"Description\":\"Write a file to S3.\",
\"Environment\":{\"Variables\":{\"bucket\":\"my-stack-bucket-1vc62xmplgguf
\"}},\"Handler\":\"index.handler\",\"MemorySize\":128,\"Role\":
\"arn:aws:iam::123456789012:role/my-functionRole-HIZXMPLEOM9E\",\"Runtime\":
\"nodejs10.x\",\"Tags\":[{\"Key\":\"lambda:createdBy\",\"Value\":\"SAM\"}],\"Timeout
\":900,\"TracingConfig\":{\"Mode\":\"Active\"}}",
        "ActualProperties": "{\"Description\":\"Write a file to S3.\",\"Environment
\":{\"Variables\":{\"bucket\":\"my-stack-bucket-1vc62xmplgguf\"}},\"Handler\":
\"index.handler\",\"MemorySize\":256,\"Role\":\"arn:aws:iam::123456789012:role/
my-functionRole-HIZXMPLEOM9E\",\"Runtime\":\"nodejs10.x\",\"Tags\":[{\"Key\":
\"lambda:createdBy\",\"Value\":\"SAM\"}],\"Timeout\":22,\"TracingConfig\":{\"Mode\":
\"Active\"}}",
        "PropertyDifferences": [
            {
                "PropertyPath": "/MemorySize",
                "ExpectedValue": "128",
                "ActualValue": "256",
                "DifferenceType": "NOT_EQUAL"
            },
            {
                "PropertyPath": "/Timeout",
                "ExpectedValue": "900",
                "ActualValue": "22",
                "DifferenceType": "NOT_EQUAL"
            }
        ],
        "StackResourceDriftStatus": "MODIFIED",
```

```
        "Timestamp": "2019-10-02T05:58:47.433Z"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetectStackResourceDrift。

detect-stack-set-drift

以下程式碼範例顯示如何使用 detect-stack-set-drift。

AWS CLI

偵測堆疊集和所有相關堆疊執行個體上的偏離

下列detect-stack-set-drift範例會在指定的堆疊集上啟動偏離偵測操作，包括與該堆疊集相關聯的所有堆疊執行個體，並傳回可用於追蹤偏離操作狀態的操作 ID。

```
aws cloudformation detect-stack-set-drift \
    --stack-set-name stack-set-drift-example
```

輸出：

```
{
    "OperationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的偵測堆疊集中的未受管組態變更。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetectStackSetDrift。

estimate-template-cost

以下程式碼範例顯示如何使用 estimate-template-cost。

AWS CLI

估算範本成本

下列estimate-template-cost範例會為目前資料夾中名為 template.yaml的範本產生成本估算。

```
aws cloudformation estimate-template-cost \
    --template-body file://template.yaml
```

輸出：

```
{
    "Url": "http://calculator.s3.amazonaws.com/calc5.html?
key=cloudformation/7870825a-xmpl-4def-92e7-c4f8dd360cca"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EstimateTemplateCost。

execute-change-set

以下程式碼範例顯示如何使用 execute-change-set。

AWS CLI

執行變更集

下列execute-change-set範例會執行由變更集名稱和堆疊名稱指定的變更集。

```
aws cloudformation execute-change-set \
    --change-set-name my-change-set \
    --stack-name my-stack
```

下列execute-change-set範例會執行變更集的完整 ARN 所指定的變更集。

```
aws cloudformation execute-change-set \
    --change-set-name arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-
change-set/bc9555ba-a949-xmpl-bfb8-f41d04ec5784
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ExecuteChangeSet。

get-stack-policy

以下程式碼範例顯示如何使用 get-stack-policy。

AWS CLI

檢視堆疊政策

下列get-stack-policy範例顯示指定堆疊的堆疊政策。若要將政策連接至堆疊，請使用 set-stack-policy命令。

```
aws cloudformation get-stack-policy \
    --stack-name my-stack
```

輸出：

```
{
    "StackPolicyBody": "{\n  \"Statement\" : [\n    {\n      \"Effect\" :
 \"Allow\",\n      \"Action\" : \"Update:*\",\n      \"Principal\": \"*\",\n
     \"Resource\" : \"*\"\n    },\n    {\n      \"Effect\" : \"Deny\",\n
 \"Action\" : \"Update:*\",\n      \"Principal\": \"*\",\n      \"Resource\" :
 \"LogicalResourceId/bucket\"\n    }\n  ]\n}\n"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetStackPolicy。

get-template-summary

以下程式碼範例顯示如何使用 get-template-summary。

AWS CLI

顯示範本摘要

下列命令會顯示指定範本檔案的資源和中繼資料的摘要資訊。

```
aws cloudformation get-template-summary \
   --template-body file://template.yaml
```

輸出：

```
{
    "Parameters": [],
    "Description": "A VPC and subnets.",
    "ResourceTypes": [
        "AWS::EC2::VPC",
        "AWS::EC2::Subnet",
        "AWS::EC2::Subnet",
        "AWS::EC2::RouteTable",
```

```
        "AWS::EC2::VPCEndpoint",
        "AWS::EC2::SubnetRouteTableAssociation",
        "AWS::EC2::SubnetRouteTableAssociation",
        "AWS::EC2::VPCEndpoint"
    ],
    "Version": "2010-09-09"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTemplateSummary。

get-template

以下程式碼範例顯示如何使用 get-template。

AWS CLI

檢視 an AWS CloudFormation 堆疊的範本內文

下列get-template命令顯示myteststack堆疊的 範本：

```
aws cloudformation get-template --stack-name myteststack
```

輸出：

```
{
    "TemplateBody": {
        "AWSTemplateFormatVersion": "2010-09-09",
        "Outputs": {
            "BucketName": {
                "Description": "Name of S3 bucket to hold website content",
                "Value": {
                    "Ref": "S3Bucket"
                }
            }
        },
        "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
 template showing how to create a publicly accessible S3 bucket. **WARNING** This
 template creates an S3 bucket. You will be billed for the AWS resources used if you
 create a stack from this template.",
        "Resources": {
            "S3Bucket": {
                "Type": "AWS::S3::Bucket",
```

```
                "Properties": {
                    "AccessControl": "PublicRead"
                }
            }
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTemplate。

`list-change-sets`

以下程式碼範例顯示如何使用 `list-change-sets`。

AWS CLI

列出變更集

下列`list-change-sets`範例顯示指定堆疊的待處理變更集清單。

```
aws cloudformation list-change-sets \
   --stack-name my-stack
```

輸出：

```
{
    "Summaries": [
        {
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "StackName": "my-stack",
            "ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/
my-change-set/70160340-7914-xmpl-bcbf-128a1fa78b5d",
            "ChangeSetName": "my-change-set",
            "ExecutionStatus": "AVAILABLE",
            "Status": "CREATE_COMPLETE",
            "CreationTime": "2019-10-02T05:38:54.297Z"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListChangeSets。

list-exports

以下程式碼範例顯示如何使用 list-exports。

AWS CLI

列出匯出

下列list-exports範例顯示目前區域中堆疊的匯出清單。

```
aws cloudformation list-exports
```

輸出：

```
{
    "Exports": [
        {
            "ExportingStackId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
            "Name": "private-vpc-subnet-a",
            "Value": "subnet-07b410xmplddcfa03"
        },
        {
            "ExportingStackId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
            "Name": "private-vpc-subnet-b",
            "Value": "subnet-075ed3xmplebd2fb1"
        },
        {
            "ExportingStackId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
            "Name": "private-vpc-vpcid",
            "Value": "vpc-011d7xmpl100e9841"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListExports。

list-imports

以下程式碼範例顯示如何使用 list-imports。

AWS CLI

列出匯入

下列list-imports範例列出匯入指定匯出的堆疊。若要取得可用匯出的清單，請使用 list-exports命令。

```
aws cloudformation list-imports \
    --export-name private-vpc-vpcid
```

輸出：

```
{
    "Imports": [
        "my-database-stack"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListImports。

list-stack-instances

以下程式碼範例顯示如何使用 list-stack-instances。

AWS CLI

列出堆疊的執行個體

下列list-stack-instances範例列出從指定堆疊集建立的執行個體。

```
aws cloudformation list-stack-instances \
    --stack-set-name enable-config
```

範例輸出包含因錯誤而無法更新之堆疊的詳細資訊：

```
{
    "Summaries": [
        {
            "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
            "Region": "us-west-2",
            "Account": "123456789012",
```

```
            "StackId": "arn:aws:cloudformation:ap-northeast-1:123456789012:stack/
StackSet-enable-config-35a6ac50-d9f8-4084-86e4-7da34d5de4c4/a1631cd0-e5fb-xmpl-
b474-0aa20f14f06e",
            "Status": "CURRENT"
        },
        {
            "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
            "Region": "us-west-2",
            "Account": "123456789012",
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/
StackSet-enable-config-e6cac20f-xmpl-46e9-8314-53e0d4591532/eab53680-e5fa-xmpl-
ba14-0a522351f81e",
            "Status": "OUTDATED",
            "StatusReason": "ResourceLogicalId:ConfigDeliveryChannel,
 ResourceType:AWS::Config::DeliveryChannel, ResourceStatusReason:Failed to put
 delivery channel 'StackSet-enable-config-e6cac20f-xmpl-46e9-8314-53e0d4591532-
ConfigDeliveryChannel-1OJWJ7XD59WR0' because the maximum number of delivery
 channels: 1 is reached. (Service: AmazonConfig; Status Code: 400; Error Code:
 MaxNumberOfDeliveryChannelsExceededException; Request ID: d14b34a0-ef7c-xmpl-
acf8-8a864370ae56)."
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListStackInstances。

list-stack-resources

以下程式碼範例顯示如何使用 `list-stack-resources`。

AWS CLI

列出堆疊中的資源

下列命令會顯示指定堆疊中的資源清單。

```
aws cloudformation list-stack-resources \
    --stack-name my-stack
```

輸出：

```
{
```

```
    "StackResourceSummaries": [
        {
            "LogicalResourceId": "bucket",
            "PhysicalResourceId": "my-stack-bucket-1vc62xmplgguf",
            "ResourceType": "AWS::S3::Bucket",
            "LastUpdatedTimestamp": "2019-10-02T04:34:11.345Z",
            "ResourceStatus": "CREATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        },
        {
            "LogicalResourceId": "function",
            "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
            "ResourceType": "AWS::Lambda::Function",
            "LastUpdatedTimestamp": "2019-10-02T05:34:27.989Z",
            "ResourceStatus": "UPDATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        },
        {
            "LogicalResourceId": "functionRole",
            "PhysicalResourceId": "my-functionRole-HIZXMPLEOM9E",
            "ResourceType": "AWS::IAM::Role",
            "LastUpdatedTimestamp": "2019-10-02T04:34:06.350Z",
            "ResourceStatus": "CREATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListStackResources。

list-stack-set-operation-results

以下程式碼範例顯示如何使用 list-stack-set-operation-results。

AWS CLI

列出堆疊集操作結果

下列命令會顯示指定堆疊集中執行個體的更新操作結果。

```
aws cloudformation list-stack-set-operation-results \
    --stack-set-name enable-config \
    --operation-id 35d45ebc-ed88-xmpl-ab59-0197a1fc83a0
```

輸出：

```
{
    "Summaries": [
        {
            "Account": "223456789012",
            "Region": "us-west-2",
            "Status": "SUCCEEDED",
            "AccountGateResult": {
                "Status": "SKIPPED",
                "StatusReason": "Function not found: arn:aws:lambda:eu-
west-1:223456789012:function:AWSCloudFormationStackSetAccountGate"
            }
        },
        {
            "Account": "223456789012",
            "Region": "ap-south-1",
            "Status": "CANCELLED",
            "StatusReason": "Cancelled since failure tolerance has exceeded"
        }
    ]
}
```

注意：除非您建立帳戶閘道函數，否則成功操作AccountGateResult時預期 SKIPPED的狀態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListStackSetOperationResults。

list-stack-set-operations

以下程式碼範例顯示如何使用 list-stack-set-operations。

AWS CLI

列出堆疊集操作

下列list-stack-set-operations範例顯示指定堆疊集上的最新操作清單。

```
aws cloudformation list-stack-set-operations \
    --stack-set-name my-stack-set
```

輸出：

```
{
    "Summaries": [
        {
            "OperationId": "35d45ebc-ed88-xmpl-ab59-0197a1fc83a0",
            "Action": "UPDATE",
            "Status": "SUCCEEDED",
            "CreationTimestamp": "2019-10-03T16:28:44.377Z",
            "EndTimestamp": "2019-10-03T16:42:08.607Z"
        },
        {
            "OperationId": "891aa98f-7118-xmpl-00b2-00954d1dd0d6",
            "Action": "UPDATE",
            "Status": "FAILED",
            "CreationTimestamp": "2019-10-03T15:43:53.916Z",
            "EndTimestamp": "2019-10-03T15:45:58.925Z"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListStackSetOperations。

list-stack-sets

以下程式碼範例顯示如何使用 list-stack-sets。

AWS CLI

列出堆疊集

下列list-stack-sets範例顯示目前區域和帳戶中的堆疊集清單。

```
aws cloudformation list-stack-sets
```

輸出：

```
{
```

```
    "Summaries": [
        {
            "StackSetName": "enable-config",
            "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
            "Description": "Enable AWS Config",
            "Status": "ACTIVE"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListStackSets。

`list-stacks`

以下程式碼範例顯示如何使用 `list-stacks`。

AWS CLI

列出 AWS CloudFormation 堆疊

下列`list-stacks`命令顯示狀態為 的所有堆疊摘要`CREATE_COMPLETE`：

```
aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE
```

輸出：

```
[
    {
        "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
        "TemplateDescription": "AWS CloudFormation Sample Template S3_Bucket: Sample
 template showing how to create a publicly accessible S3 bucket. **WARNING** This
 template creates an S3 bucket. You will be billed for the AWS resources used if you
 create a stack from this template.",
        "StackStatusReason": null,
        "CreationTime": "2013-08-26T03:27:10.190Z",
        "StackName": "myteststack",
        "StackStatus": "CREATE_COMPLETE"
    }
]
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListStacks。

list-type-registrations

以下程式碼範例顯示如何使用 list-type-registrations。

AWS CLI

列出已完成的類型註冊

下列list-type-registrations範例顯示指定類型已完成類型註冊的清單。

```
aws cloudformation list-type-registrations \
    --type RESOURCE \
    --type-name My::Logs::LogGroup \
    --registration-status-filter COMPLETE
```

輸出：

```
{
    "RegistrationTokenList": [
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
    ]
}
```

如需詳細資訊，請參閱《CloudFormation 使用者指南》中的使用 CloudFormation 登錄檔。 AWS CloudFormation

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTypeRegistrations。

list-type-versions

以下程式碼範例顯示如何使用 list-type-versions。

AWS CLI

列出延伸模組的版本

下列list-type-versions範例會傳回 擴充功能版本的摘要資訊。

```
aws cloudformation list-type-versions \
  --endpoint https://example.com \
```

```
  --region us-west-2 \
  --type RESOURCE \
  --type-name My::Resource::Example \
  --publisher-id 123456789012
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的使用 CloudFormation 登錄檔。AWS CloudFormation

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTypeVersions。

list-types

以下程式碼範例顯示如何使用 list-types。

AWS CLI

列出帳戶中的私有資源類型

下列list-types範例顯示目前在目前 AWS 帳戶中註冊的私有資源類型清單。

```
aws cloudformation list-types
```

輸出：

```
{
    "TypeSummaries": [
        {
            "Description": "WordPress blog resource for internal use",
            "LastUpdated": "2019-12-04T18:28:15.059Z",
            "TypeName": "My::WordPress::BlogExample",
            "TypeArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/
My-WordPress-BlogExample",
            "DefaultVersionId": "00000005",
            "Type": "RESOURCE"
        },
        {
            "Description": "Customized resource derived from AWS::Logs::LogGroup",
            "LastUpdated": "2019-12-04T18:28:15.059Z",
            "TypeName": "My::Logs::LogGroup",
            "TypeArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/
My-Logs-LogGroup",
```

```
            "DefaultVersionId": "00000003",
            "Type": "RESOURCE"
        }
    ]
}
```

如需詳細資訊，請參閱《CloudFormation 使用者指南》中的使用 CloudFormation 登錄檔。 AWS CloudFormation

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTypes。

package

以下程式碼範例顯示如何使用 package。

AWS CLI

下列命令template.json會將本機成品上傳到 S3 儲存貯體以匯出名為 的範本，bucket-name並將匯出的範本寫入 packaged-template.json：

```
aws cloudformation package --template-file /path_to_template/template.json --s3-
bucket bucket-name --output-template-file packaged-template.json --use-json
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的套件。

publish-type

以下程式碼範例顯示如何使用 publish-type。

AWS CLI

發佈 延伸模組

下列publish-type範例會將指定的延伸項目發佈至 CloudFormation 登錄檔，做為此區域中的公有延伸項目。

```
aws cloudformation publish-type \
  --region us-west-2 \
  --type RESOURCE \
  --type-name Example::Test::1234567890abcdef0
```

輸出：

```
{
    "PublicTypeArn":"arn:aws:cloudformation:us-west-2::type/
resource/000q6TfUovXsEMmgKowxDZLlwqr2QUshd2e75c8c/Example-
Test-1234567890abcdef0/1.0.0"
}
```

如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的使用 CloudFormation 登錄檔。AWS CloudFormation

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PublishType。

register-publisher

以下程式碼範例顯示如何使用 register-publisher。

AWS CLI

註冊發佈者

下列register-publisher範例會註冊發佈者並接受 條款與條件參數。

```
aws cloudformation register-publisher \
  --region us-west-2 \
  --accept-terms-and-conditions
```

輸出：

```
{
    "PublisherId": "000q6TfUovXsEMmgKowxDZLlwqr2QUshd2e75c8c"
}
```

如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的使用 CloudFormation 登錄檔。AWS CloudFormation

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterPublisher。

register-type

以下程式碼範例顯示如何使用 register-type。

AWS CLI

註冊資源類型

下列register-type範例會將指定的資源類型註冊為使用者帳戶中的私有資源類型。

```
aws cloudformation register-type \
    --type-name My::Organization::ResourceName \
    --schema-handler-package s3://bucket_name/my-organization-resource_name.zip \
    --type RESOURCE
```

輸出：

```
{
    "RegistrationToken": "f5525280-104e-4d35-bef5-8f1f1example"
}
```

如需詳細資訊，請參閱 CloudFormation 命令列界面開發使用者指南中的註冊資源提供者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterType。

set-stack-policy

以下程式碼範例顯示如何使用 set-stack-policy。

AWS CLI

套用堆疊政策

下列set-stack-policy範例會停用指定堆疊中指定資源的更新。 stack-policy.json 是定義堆疊中資源上允許之操作的 JSON 文件。

```
aws cloudformation set-stack-policy \
    --stack-name my-stack \
    --stack-policy-body file://stack-policy.json
```

輸出：

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
```

```
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    },
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "LogicalResourceId/bucket"
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetStackPolicy。

set-type-configuration

以下程式碼範例顯示如何使用 set-type-configuration。

AWS CLI

設定資料

下列set-type-configuration範例指定指定帳戶和區域中已註冊 CloudFormation 延伸模組的組態資料。

```
aws cloudformation set-type-configuration \
    --region us-west-2 \
    --type RESOURCE \
    --type-name Example::Test::Type \
    --configuration-alias default \
    --configuration "{\"CredentialKey\": \"testUserCredential\"}"
```

輸出：

```
{
    "ConfigurationArn": "arn:aws:cloudformation:us-west-2:123456789012:type-
configuration/resource/Example-Test-Type/default"
}
```

如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的使用 CloudFormation 登錄檔。
AWS CloudFormation

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetTypeConfiguration。

set-type-default-version

以下程式碼範例顯示如何使用 set-type-default-version。

AWS CLI

設定類型的預設版本

下列set-type-default-version範例會將指定的類型版本設定為此類型的預設值。

```
aws cloudformation set-type-default-version \
    --type RESOURCE \
    --type-name My::Logs::LogGroup \
    --version-id 00000003
```

此命令不會產生輸出。

如需詳細資訊，請參閱《CloudFormation 使用者指南》中的使用 CloudFormation 登錄檔。 AWS CloudFormation

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetTypeDefaultVersion。

signal-resource

以下程式碼範例顯示如何使用 signal-resource。

AWS CLI

向資源發出訊號

下列signal-resource範例會發出訊號success，以滿足名為 之堆疊MyWaitCondition中名為的等待條件my-stack。

```
aws cloudformation signal-resource \
    --stack-name my-stack \
    --logical-resource-id MyWaitCondition \
    --unique-id 1234 \
    --status SUCCESS
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SignalResource。

stop-stack-set-operation

以下程式碼範例顯示如何使用 stop-stack-set-operation。

AWS CLI

停止堆疊集操作

下列stop-stack-set-operation範例會停止指定堆疊集上的進行中更新操作。

```
aws cloudformation stop-stack-set-operation \
    --stack-set-name my-stack-set \
    --operation-id 1261cd27-490b-xmpl-ab42-793a896c69e6
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopStackSetOperation。

test-type

以下程式碼範例顯示如何使用 test-type。

AWS CLI

測試擴充功能

下列test-type範例會測試已註冊的延伸模組，以確保其符合在 CloudFormation 登錄檔中發佈的所有必要要求。

```
aws cloudformation test-type \
    --arn arn:aws:cloudformation:us-west-2:123456789012:type/resource/Sample-Test-Resource123/00000001
```

輸出：

```
{
    "TypeVersionArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/Sample-Test-Resource123/00000001"
```

```
}
```

如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的使用 CloudFormation 登錄檔。AWS CloudFormation

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TestType。

update-stack-instances

以下程式碼範例顯示如何使用 update-stack-instances。

AWS CLI

更新堆疊執行個體

下列update-stack-instances範例會使用最新的設定，在兩個區域中的兩個帳戶中重試堆疊執行個體的更新。指定的容錯能力設定可確保在所有帳戶和區域中嘗試更新，即使某些堆疊無法更新。

```
aws cloudformation update-stack-instances \
    --stack-set-name my-stack-set \
    --accounts 123456789012 567890123456 \
    --regions us-east-1 us-west-2 \
    --operation-preferences FailureToleranceCount=3
```

輸出：

```
{
    "OperationId": "103ebdf2-21ea-xmpl-8892-de5e30733132"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateStackInstances。

update-stack-set

以下程式碼範例顯示如何使用 update-stack-set。

AWS CLI

更新堆疊集

下列update-stack-set範例會將索引鍵名稱為 Owner且值為 的標籤IT新增至指定堆疊集中的堆疊執行個體。

```
aws cloudformation update-stack-set \
    --stack-set-name my-stack-set \
    --use-previous-template \
    --tags Key=Owner,Value=IT
```

輸出：

```
{
    "OperationId": "e2b60321-6cab-xmpl-bde7-530c6f47950e"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateStackSet。

update-stack

以下程式碼範例顯示如何使用 update-stack。

AWS CLI

更新 AWS CloudFormation 堆疊

下列update-stack命令會更新mystack堆疊的範本和輸入參數：

```
aws cloudformation update-stack --stack-name mystack --
template-url https://s3.amazonaws.com/sample/updated.template --
parameters ParameterKey=KeyPairName,ParameterValue=SampleKeyPair
 ParameterKey=SubnetIDs,ParameterValue=SampleSubnetID1\\,SampleSubnetID2
```

下列update-stack命令只會更新mystack堆疊的SubnetIDs參數值。如果您未指定參數值，則會使用範本中指定的預設值：

```
aws cloudformation update-stack --stack-name mystack --
template-url https://s3.amazonaws.com/sample/updated.template
 --parameters ParameterKey=KeyPairName,UsePreviousValue=true
 ParameterKey=SubnetIDs,ParameterValue=SampleSubnetID1\\,UpdatedSampleSubnetID2
```

下列update-stack命令會將兩個堆疊通知主題新增至mystack堆疊：

```
aws cloudformation update-stack --stack-name mystack --use-previous-template --
notification-arns "arn:aws:sns:use-east-1:123456789012:mytopic1" "arn:aws:sns:us-
east-1:123456789012:mytopic2"
```

如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的 CloudFormation 堆疊更新。AWS CloudFormation

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateStack。

update-termination-protection

以下程式碼範例顯示如何使用 update-termination-protection。

AWS CLI

啟用終止保護

下列update-termination-protection範例會在指定的堆疊上啟用終止保護。

```
aws cloudformation update-termination-protection \
    --stack-name my-stack \
    --enable-termination-protection
```

輸出：

```
{
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateTerminationProtection。

validate-template

以下程式碼範例顯示如何使用 validate-template。

AWS CLI

驗證 an AWS CloudFormation 範本

下列validate-template命令會驗證sampletemplate.json範本：

```
aws cloudformation validate-template --template-body file://sampletemplate.json
```

輸出：

```
{
    "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample template
 showing how to create a publicly accessible S3 bucket. **WARNING** This template
 creates an S3 bucket. You will be billed for the AWS resources used if you create a
 stack from this template.",
    "Parameters": [],
    "Capabilities": []
}
```

如需詳細資訊，請參閱《 AWS CloudFormation AWS 使用者指南》中的使用 CloudFormation 範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ValidateTemplate。

使用 的 CloudFront 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 CloudFront 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-distribution-tenant-web-acl`

以下程式碼範例顯示如何使用 `associate-distribution-tenant-web-acl`。

AWS CLI

將 Web ACL 與 CloudFront 分佈租用戶建立關聯

下列associate-distribution-tenant-web-acl範例會將 Web ACL 與 CloudFront 分佈與 ETag 建立關聯E13V1IB3VIYABC。

```
aws cloudfront associate-distribution-tenant-web-acl \
    --id dt_2wjDZi3hD1ivOXf6rpZJO1AB \
    --if-match E13V1IB3VIYABC \
    --web-acl-arn arn:aws:wafv2:us-east-1:123456789012:global/webacl/web-global-example/626900da-5f64-418b-ba9b-743f37123ABC
```

輸出：

```
{
    "ETag": "E1VC38T7YXBABC",
    "Id": "dt_2wjDZi3hD1ivOXf6rpZJO1AB",
    "WebACLArn": "arn:aws:wafv2:us-east-1:123456789012:global/webacl/web-global-example/626900da-5f64-418b-ba9b-743f37123ABC"
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的使用 AWS WAF 保護。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateDistributionTenantWebAcl。

associate-distribution-web-acl

以下程式碼範例顯示如何使用 associate-distribution-web-acl。

AWS CLI

將 Web ACL 與 CloudFront 分佈建立關聯

下列associate-distribution-web-acl範例會將 Web ACL 與 CloudFront 分佈建立關聯。

```
aws cloudfront associate-distribution-web-acl \
    --id E1XNX8R2GOAABC \
    --if-match E2YWS1C2J30ABC \
    --web-acl-arn arn:aws:wafv2:us-east-1:123456789012:global/webacl/web-global-example/626900da-5f64-418b-ba9b-743f3746cABC
```

輸出：

```
{
```

```
    "ETag": "E3QE7ED60U0ABC",
    "Id": "E1XNX8R2GOAABC",
    "WebACLArn": "arn:aws:wafv2:us-east-1:123456789012:global/webacl/web-global-
example/626900da-5f64-418b-ba9b-743f3746cABC"
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的使用 AWS WAF 保護。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateDistributionWebAcl。

create-cloud-front-origin-access-identity

以下程式碼範例顯示如何使用 create-cloud-front-origin-access-identity。

AWS CLI

建立 CloudFront 原始存取身分

下列範例透過提供 OAI 組態做為命令列引數來建立 CloudFront 原始存取身分 (OAI)：

```
aws cloudfront create-cloud-front-origin-access-identity \
    --cloud-front-origin-access-identity-config \
        CallerReference="cli-example",Comment="Example OAI"
```

您可以在 JSON 檔案中提供 OAI 組態來完成相同的作業，如下列範例所示：

```
aws cloudfront create-cloud-front-origin-access-identity \
    --cloud-front-origin-access-identity-config file://OAI-config.json
```

檔案OAI-config.json是目前目錄中的 JSON 文件，其中包含下列項目：

```
{
    "CallerReference": "cli-example",
    "Comment": "Example OAI"
}
```

無論您是使用命令列引數或 JSON 檔案提供 OAI 組態，輸出都是相同的：

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/origin-access-identity/
cloudfront/E74FTE3AEXAMPLE",
```

```
    "ETag": "E2QWRUHEXAMPLE",
    "CloudFrontOriginAccessIdentity": {
        "Id": "E74FTE3AEXAMPLE",
        "S3CanonicalUserId":
 "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
        "CloudFrontOriginAccessIdentityConfig": {
            "CallerReference": "cli-example",
            "Comment": "Example OAI"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCloudFrontOriginAccessIdentity。

`create-connection-group`

以下程式碼範例顯示如何使用 `create-connection-group`。

AWS CLI

在 CloudFront 中建立連線群組

下列`create-connection-group`範例會建立已啟用的連線群組、指定 Anycast 靜態 IP 清單，以及停用 IPv6。

```
aws cloudfront create-connection-group \
    --name cg-with-anycast-ip-list \
    --no-ipv6-enabled \
    --enabled \
    --anycast-ip-list-id aip_CCkW6gKrDiBD4n78123ABC \
    --tags "Items=[{Key=abc,Value=123}]"
```

輸出：

```
{
    "ETag": "E23ZP02F085ABC",
    "ConnectionGroup": {
        "Id": "cg_2yb6uj74B4PCbfhT31WFdiSABC",
        "Name": "cg-with-anycast-ip-list",
        "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2yb6uj74B4PCbfhT31WFdiSABC",
```

```
        "CreatedTime": "2025-06-16T16:25:50.061000+00:00",
        "LastModifiedTime": "2025-06-16T16:25:50.061000+00:00",
        "Tags": {
            "Items": [
                {
                    "Key": "abc",
                    "Value": "123"
                }
            ]
        },
        "Ipv6Enabled": false,
        "RoutingEndpoint": "dj6xusxq65abc.cloudfront.net",
        "AnycastIpListId": "aip_CCkW6gKrDiBD4n78123ABC",
        "Status": "InProgress",
        "Enabled": true,
        "IsDefault": false
    }
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的建立自訂連線群組（選用）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateConnectionGroup。

create-distribution-tenant

以下程式碼範例顯示如何使用 create-distribution-tenant。

AWS CLI

範例 1：建立使用自訂憑證的 CloudFront 分佈租用戶

下列create-distribution-tenant範例會建立 CloudFront 分佈租用戶，指定自訂以停用 WAF、新增地理限制，並使用另一個 TLS 憑證。

```
aws cloudfront create-distribution-tenant \
    --cli-input-json file://tenant.json
```

tenant.json 的內容：

```
{
    "DistributionId": "E1XNX8R2GOAABC",
    "Domains": [
```

```
        {
            "Domain": "example.com"
        }
    ],
    "Parameters": [
        {
            "Name": "testParam",
            "Value": "defaultValue"
        }
    ],
    "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
    "Enabled": false,
    "Tags": {
        "Items": [
            {
                "Key": "tag",
                "Value": "tagValue"
            }
        ]
    },
    "Name": "new-tenant-customizations",
    "Customizations": {
        "GeoRestrictions": {
            "Locations": ["DE"],
            "RestrictionType": "whitelist"
        },
        "WebAcl": {
            "Action": "disable"
        },
        "Certificate": {
            "Arn": "arn:aws:acm:us-east-1:123456789012:certificate/ec53f564-
ea5a-4e4a-a0a2-e3c989449abc"
        }
    }
}
```

輸出：

```
{
    "ETag": "E23ZP02F085ABC",
    "DistributionTenant": {
        "Id": "dt_2yN5tYwVbPKr7m2IB69M1yp1AB",
        "DistributionId": "E1XNX8R2GOAABC",
```

```
        "Name": "new-tenant-customizations",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2yN5tYwVbPKr7m2IB69M1yp1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "active"
            }
        ],
        "Tags": {
            "Items": [
                {
                    "Key": "tag",
                    "Value": "tagValue"
                }
            ]
        },
        "Customizations": {
            "WebAcl": {
                "Action": "disable"
            },
            "Certificate": {
                "Arn": "arn:aws:acm:us-east-1:123456789012:certificate/ec53f564-
ea5a-4e4a-a0a2-e3c989449abc"
            },
            "GeoRestrictions": {
                "RestrictionType": "whitelist",
                "Locations": [
                    "DE"
                ]
            }
        },
        "Parameters": [
            {
                "Name": "testParam",
                "Value": "defaultValue"
            }
        ],
        "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
        "CreatedTime": "2025-06-11T17:20:06.432000+00:00",
        "LastModifiedTime": "2025-06-11T17:20:06.432000+00:00",
        "Enabled": false,
        "Status": "InProgress"
    }
```

```
}
```

範例 2：使用繼承的憑證建立分佈租用戶

下列create-distribution-tenant範例會建立分佈租用戶，並指定來自多租用戶分佈的繼承 TLS 憑證。

```
aws cloudfront create-distribution-tenant \
    --cli-input-json file://tenant.json
```

tenant.json 的內容：

```
{
    "DistributionId": "E1HVIAU7U12ABC",
    "Domains": [
        {
            "Domain": "example.com"
        }
    ],
    "Parameters": [
        {
            "Name": "tenantName",
            "Value": "first-tenant"
        }
    ],
    "Enabled": true,
    "Name": "new-tenant-no-cert"
}
```

輸出：

```
{
    "ETag": "E23ZP02F0ABC",
    "DistributionTenant": {
        "Id": "dt_2zhRB0vBe0B72LZCVy1mgzI1AB",
        "DistributionId": "E1HVIAU7U12ABC",
        "Name": "new-tenant-no-cert",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2zhRB0vBe0B72LZCVy1mgzI1AB",
        "Domains": [
            {
                "Domain": "example.com",
```

```
                "Status": "active"
            }
        ],
        "Parameters": [
            {
                "Name": "tenantName",
                "Value": "first-tenant"
            }
        ],
        "ConnectionGroupId": "cg_2yQEwpipGFN0hhA0ZemPabOABC",
        "CreatedTime": "2025-07-10T20:59:38.414000+00:00",
        "LastModifiedTime": "2025-07-10T20:59:38.414000+00:00",
        "Enabled": true,
        "Status": "InProgress"
    }
}
```

範例 3：使用 CloudFront 託管驗證字符建立 CloudFront 分佈租用戶

下列create-distribution-tenant範例會建立分佈租用戶，並為您的網域名稱使用 CloudFront 託管的驗證字符。

```
aws cloudfront create-distribution-tenant \
    --cli-input-json file://tenant.json
```

tenant.json 的內容：

```
{
    "DistributionId": "E2GJ5J9QN12ABC",
    "Domains": [
        {
            "Domain": "example.com"
        }
    ],
    "Parameters": [
        {
            "Name": "tenantName",
            "Value": "first-tenant"
        }
    ],
    "ConnectionGroupId": "cg_2yQEwpipGFN0hhA0ZemPabOABC",
    "Enabled": true,
    "Name": "new-tenant-cf-hosted",
```

```
    "ManagedCertificateRequest": {
        "ValidationTokenHost": "cloudfront"
    }
}
```

重要：若要成功執行此命令，您必須設定 CNAME DNS 記錄，將新網域 (example.com) 指向與分發租用戶相關聯之連線群組的路由端點。也必須傳播此 CNAME 記錄CloudFront 才能成功完成此請求。

輸出：

```
{
    "ETag": "E23ZP02F0ABC",
    "DistributionTenant": {
        "Id": "dt_2zhStKrA524GvvTWJX92Ozl1AB",
        "DistributionId": "E2GJ5J9QN12ABC",
        "Name": "new-tenant-cf-hosted",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2zhStKrA524GvvTWJX92Ozl1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "inactive"
            }
        ],
        "Parameters": [
            {
                "Name": "tenantName",
                "Value": "first-tenant"
            }
        ],
        "ConnectionGroupId": "cg_2zhSaGatwwXjTjE42nneZzqABC",
        "CreatedTime": "2025-07-10T21:13:46.416000+00:00",
        "LastModifiedTime": "2025-07-10T21:13:46.416000+00:00",
        "Enabled": true,
        "Status": "InProgress"
    }
}
```

範例 4：使用自我託管驗證字符建立 CloudFront 分佈租用戶

下列create-distribution-tenant範例會建立 CloudFront 分佈租用戶，並使用自我託管驗證字符。

```
aws cloudfront create-distribution-tenant \
    --cli-input-json file://tenant.json
```

tenant.json 的內容：

```
{
    "DistributionId": "E2GJ5J9QN12ABC",
    "Domains": [
        {
            "Domain": "example.com"
        }
    ],
    "Parameters": [
        {
            "Name": "tenantName",
            "Value": "first-tenant"
        }
    ],
    "Enabled": true,
    "Name": "new-tenant-self-hosted",
    "ManagedCertificateRequest": {
        "ValidationTokenHost": "self-hosted"
    }
}
```

輸出：

```
{
    "ETag": "E23ZP02F0ABC",
    "DistributionTenant": {
        "Id": "dt_2zhTFBV93OfFJJ3YMdNM5BC1AB",
        "DistributionId": "E2GJ5J9QN12ABC",
        "Name": "new-tenant-self-hosted",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2zhTFBV93OfFJJ3YMdNM5BC1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "inactive"
            }
        ],
        "Parameters": [
```

```
            {
                "Name": "tenantName",
                "Value": "first-tenant"
            }
        ],
        "ConnectionGroupId": "cg_2yQEwpipGFN0hhA0ZemPabOABC",
        "CreatedTime": "2025-07-10T21:16:39.828000+00:00",
        "LastModifiedTime": "2025-07-10T21:16:39.828000+00:00",
        "Enabled": true,
        "Status": "InProgress"
    }
}
```

重要事項：在您執行此命令之後，將會在不驗證的情況下建立分發租用戶。若要在準備好開始接收流量時驗證受管憑證請求並設定 DNS，請參閱《Amazon CloudFront 開發人員指南》中的完成網域設定。

如需建立分佈租用戶的詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的建立分佈。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDistributionTenant。

create-distribution-with-tags

以下程式碼範例顯示如何使用 create-distribution-with-tags。

AWS CLI

使用標籤建立 CloudFront 分佈

下列create-distribution-with-tags範例透過在名為 的 JSON 檔案中提供分佈組態和標籤，建立具有兩個標籤的分佈dist-config-with-tags.json。

```
aws cloudfront create-distribution-with-tags \
    --distribution-config-with-tags file://dist-config-with-tags.json
```

檔案dist-config-with-tags.json是目前資料夾中的 JSON 文件。請注意 檔案頂端的Tags物件，其中包含兩個標籤：

Name = ExampleDistributionProject = ExampleProject

dist-config-with-tags.json 的內容：

```
{
```

```
    "Tags": {
        "Items": [
            {
                "Key": "Name",
                "Value": "ExampleDistribution"
            },
            {
                "Key": "Project",
                "Value": "ExampleProject"
            }
        ]
    },
    "DistributionConfig": {
        "CallerReference": "cli-example",
        "Aliases": {
            "Quantity": 0
        },
        "DefaultRootObject": "index.html",
        "Origins": {
            "Quantity": 1,
            "Items": [
                {
                    "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                    "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                    "OriginPath": "",
                    "CustomHeaders": {
                        "Quantity": 0
                    },
                    "S3OriginConfig": {
                        "OriginAccessIdentity": ""
                    }
                }
            ]
        },
        "OriginGroups": {
            "Quantity": 0
        },
        "DefaultCacheBehavior": {
            "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
            "ForwardedValues": {
                "QueryString": false,
                "Cookies": {
                    "Forward": "none"
                },
```

```
                "Headers": {
                    "Quantity": 0
                },
                "QueryStringCacheKeys": {
                    "Quantity": 0
                }
            },
            "TrustedSigners": {
                "Enabled": false,
                "Quantity": 0
            },
            "ViewerProtocolPolicy": "allow-all",
            "MinTTL": 0,
            "AllowedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ],
                "CachedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ]
                }
            },
            "SmoothStreaming": false,
            "DefaultTTL": 86400,
            "MaxTTL": 31536000,
            "Compress": false,
            "LambdaFunctionAssociations": {
                "Quantity": 0
            },
            "FieldLevelEncryptionId": ""
        },
        "CacheBehaviors": {
            "Quantity": 0
        },
        "CustomErrorResponses": {
            "Quantity": 0
        },
        "Comment": "",
        "Logging": {
```

```
            "Enabled": false,
            "IncludeCookies": false,
            "Bucket": "",
            "Prefix": ""
        },
        "PriceClass": "PriceClass_All",
        "Enabled": true,
        "ViewerCertificate": {
            "CloudFrontDefaultCertificate": true,
            "MinimumProtocolVersion": "TLSv1",
            "CertificateSource": "cloudfront"
        },
        "Restrictions": {
            "GeoRestriction": {
                "RestrictionType": "none",
                "Quantity": 0
            }
        },
        "WebACLId": "",
        "HttpVersion": "http2",
        "IsIPV6Enabled": true
    }
}
```

輸出：

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EDFDVBD6EXAMPLE",
    "ETag": "E2QWRUHEXAMPLE",
    "Distribution": {
        "Id": "EDFDVBD6EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
        "Status": "InProgress",
        "LastModifiedTime": "2019-12-04T23:35:41.433Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-example",
```

```
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                        "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-
example",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
```

```
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": ""
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
```

```
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDistributionWithTags。

`create-distribution`

以下程式碼範例顯示如何使用 `create-distribution`。

AWS CLI

範例 1：建立 CloudFront 分佈

下列`create-distribution`範例會為名為 的 S3 儲存貯體建立分佈，`amzn-s3-demo-bucket`並使用命令列引數將 指定`index.html`為預設根物件。

```
aws cloudfront create-distribution \
    --origin-domain-name amzn-s3-demo-bucket.s3.amazonaws.com \
    --default-root-object index.html
```

輸出：

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EMLARXS9EXAMPLE",
    "ETag": "E9LHASXEXAMPLE",
    "Distribution": {
        "Id": "EMLARXS9EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EMLARXS9EXAMPLE",
        "Status": "InProgress",
        "LastModifiedTime": "2019-11-22T00:55:15.705Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
```

```
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-example",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                        "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-
example",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
```

```
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": ""
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": true,
```

```
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true
        }
    }
}
```

範例 2：使用 JSON 檔案建立 CloudFront 分佈

下列create-distribution範例會為名為 的 S3 儲存貯體建立分佈amzn-s3-demo-bucket，並使用 JSON 檔案將 指定index.html為預設根物件。

```
aws cloudfront create-distribution \
    --distribution-config file://dist-config.json
```

dist-config.json 的內容：

```
{
    "CallerReference": "cli-example",
    "Aliases": {
        "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                "OriginPath": "",
                "CustomHeaders": {
                    "Quantity": 0
```

```
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "OriginGroups": {
        "Quantity": 0
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
        "ForwardedValues": {
            "QueryString": false,
            "Cookies": {
                "Forward": "none"
            },
            "Headers": {
                "Quantity": 0
            },
            "QueryStringCacheKeys": {
                "Quantity": 0
            }
        },
        "TrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ViewerProtocolPolicy": "allow-all",
        "MinTTL": 0,
        "AllowedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ],
            "CachedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ]
            }
        },
```

```
        "SmoothStreaming": false,
        "DefaultTTL": 86400,
        "MaxTTL": 31536000,
        "Compress": false,
        "LambdaFunctionAssociations": {
            "Quantity": 0
        },
        "FieldLevelEncryptionId": ""
    },
    "CacheBehaviors": {
        "Quantity": 0
    },
    "CustomErrorResponses": {
        "Quantity": 0
    },
    "Comment": "",
    "Logging": {
        "Enabled": false,
        "IncludeCookies": false,
        "Bucket": "",
        "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": true,
    "ViewerCertificate": {
        "CloudFrontDefaultCertificate": true,
        "MinimumProtocolVersion": "TLSv1",
        "CertificateSource": "cloudfront"
    },
    "Restrictions": {
        "GeoRestriction": {
            "RestrictionType": "none",
            "Quantity": 0
        }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
}
```

如需範例輸出，請參閱範例 1。

範例 3：使用憑證建立 CloudFront 多租用戶分佈

下列create-distribution範例會建立具有多租用戶支援的 CloudFront 分佈，並指定 TLS 憑證。

```
aws cloudfront create-distribution \
    --distribution-config file://dist-config.json
```

dist-config.json 的內容：

```
{
    "CallerReference": "cli-example-with-cert",
    "Comment": "CLI example distribution",
    "DefaultRootObject": "index.html",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "DomainName": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "OriginPath": "/{{tenantName}}",
                "CustomHeaders": {
                    "Quantity": 0
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
        "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
        "ViewerProtocolPolicy": "allow-all",
        "AllowedMethods": {
            "Quantity": 2,
            "Items": ["HEAD", "GET"],
            "CachedMethods": {
                "Quantity": 2,
                "Items": ["HEAD", "GET"]
            }
        }
    },
    "Enabled": true,
    "ViewerCertificate": {
```

```
        "ACMCertificateArn": "arn:aws:acm:us-
east-1:123456789012:certificate/191306a1-db01-49ca-90ef-fc414ee5dabc",
        "SSLSupportMethod": "sni-only"
    },
    "HttpVersion": "http2",
    "ConnectionMode": "tenant-only",
    "TenantConfig": {
        "ParameterDefinitions": [
            {
                "Name": "tenantName",
                "Definition": {
                    "StringSchema": {
                        "Comment": "tenantName parameter",
                        "DefaultValue": "root",
                        "Required": false
                    }
                }
            }
        ]
    }
}
```

輸出：

```
{
    "Location": "https://cloudfront.amazonaws.com/2020-05-31/distribution/
E1HVIAU7UABC",
    "ETag": "E20LT7R1BABC",
    "Distribution": {
        "Id": "E1HVIAU7U12ABC",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/E1HVIAU7U12ABC",
        "Status": "InProgress",
        "LastModifiedTime": "2025-07-10T20:33:31.117000+00:00",
        "InProgressInvalidationBatches": 0,
        "DomainName": "example.com",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ActiveTrustedKeyGroups": {
            "Enabled": false,
            "Quantity": 0
        },
```

```
        "DistributionConfig": {
            "CallerReference": "cli-example-with-cert",
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                        "DomainName": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                        "OriginPath": "/{{tenantName}}",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        },
                        "ConnectionAttempts": 3,
                        "ConnectionTimeout": 10,
                        "OriginShield": {
                            "Enabled": false
                        },
                        "OriginAccessControlId": ""
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "TrustedKeyGroups": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": ["HEAD", "GET"],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": ["HEAD", "GET"]
                    }
                },
```

```
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": "",
                "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
                "GrpcConfig": {
                    "Enabled": false
                }
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "CLI example distribution",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": false,
                "ACMCertificateArn": "arn:aws:acm:us-
east-1:123456789012:certificate/1954f095-11b6-4daf-9952-0c308a00abc",
                "SSLSupportMethod": "sni-only",
                "MinimumProtocolVersion": "TLSv1.2_2021",
                "Certificate": "arn:aws:acm:us-
east-1:123456789012:certificate/1954f095-11b6-4daf-9952-0c308a00abc",
                "CertificateSource": "acm"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
```

```
            "HttpVersion": "http2",
            "TenantConfig": {
                "ParameterDefinitions": [
                    {
                        "Name": "tenantName",
                        "Definition": {
                            "StringSchema": {
                                "Comment": "tenantName parameter",
                                "DefaultValue": "root",
                                "Required": false
                            }
                        }
                    }
                ]
            },
            "ConnectionMode": "tenant-only"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的使用分佈。

範例 4：建立不含憑證的 CloudFront 多租戶分佈

下列create-distribution範例會建立具有多租用戶支援但沒有 TLS 憑證的 CloudFront 分佈。

```
aws cloudfront create-distribution \
    --distribution-config file://dist-config.json
```

dist-config.json 的內容：

```
{
    "CallerReference": "cli-example",
    "Comment": "CLI example distribution",
    "DefaultRootObject": "index.html",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "DomainName": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "OriginPath": "/{{tenantName}}",
                "CustomHeaders": {
```

```
                    "Quantity": 0
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
        "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
        "ViewerProtocolPolicy": "allow-all",
        "AllowedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ],
            "CachedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ]
            }
        }
    },
    "Enabled": true,
    "HttpVersion": "http2",
    "ConnectionMode": "tenant-only",
    "TenantConfig": {
        "ParameterDefinitions": [
            {
                "Name": "tenantName",
                "Definition": {
                    "StringSchema": {
                        "Comment": "tenantName parameter",
                        "DefaultValue": "root",
                        "Required": false
                    }
                }
            }
        ]
    }
```

```
}
```

輸出：

```
{
    "Location": "https://cloudfront.amazonaws.com/2020-05-31/distribution/
E2GJ5J9QN12ABC",
    "ETag": "E37YLVVQIABC",
    "Distribution": {
        "Id": "E2GJ5J9QNABC",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/E2GJ5J9QN12ABC",
        "Status": "InProgress",
        "LastModifiedTime": "2025-07-10T20:35:20.565000+00:00",
        "InProgressInvalidationBatches": 0,
        "DomainName": "example.com",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ActiveTrustedKeyGroups": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-example-no-cert",
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                        "DomainName": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                        "OriginPath": "/{{tenantName}}",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        },
                        "ConnectionAttempts": 3,
                        "ConnectionTimeout": 10,
                        "OriginShield": {
```

```
                    "Enabled": false
                },
                "OriginAccessControlId": ""
            }
        ]
    },
    "OriginGroups": {
        "Quantity": 0
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
        "TrustedKeyGroups": {
            "Enabled": false,
            "Quantity": 0
        },
        "ViewerProtocolPolicy": "allow-all",
        "AllowedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ],
            "CachedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ]
            }
        },
        "Compress": false,
        "LambdaFunctionAssociations": {
            "Quantity": 0
        },
        "FunctionAssociations": {
            "Quantity": 0
        },
        "FieldLevelEncryptionId": "",
        "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
        "GrpcConfig": {
            "Enabled": false
        }
    },
    "CacheBehaviors": {
```

```
            "Quantity": 0
        },
        "CustomErrorResponses": {
            "Quantity": 0
        },
        "Comment": "CLI example distribution",
        "Logging": {
            "Enabled": false,
            "IncludeCookies": false,
            "Bucket": "",
            "Prefix": ""
        },
        "Enabled": true,
        "ViewerCertificate": {
            "CloudFrontDefaultCertificate": true,
            "SSLSupportMethod": "sni-only",
            "MinimumProtocolVersion": "TLSv1",
            "CertificateSource": "cloudfront"
        },
        "Restrictions": {
            "GeoRestriction": {
                "RestrictionType": "none",
                "Quantity": 0
            }
        },
        "WebACLId": "",
        "HttpVersion": "http2",
        "TenantConfig": {
            "ParameterDefinitions": [
                {
                    "Name": "tenantName",
                    "Definition": {
                        "StringSchema": {
                            "Comment": "tenantName parameter",
                            "DefaultValue": "root",
                            "Required": false
                        }
                    }
                }
            ]
        },
        "ConnectionMode": "tenant-only"
    }
}
```

```
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的設定分佈。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDistribution。

`create-field-level-encryption-config`

以下程式碼範例顯示如何使用 `create-field-level-encryption-config`。

AWS CLI

建立 CloudFront 欄位層級加密組態

下列範例透過在名為 的 JSON 檔案中提供組態參數來建立欄位層級加密組態`fle-config.json`。您必須先擁有欄位層級加密設定檔，才能建立欄位層級加密組態。若要建立設定檔，請參閱 create-field-level-encryption-profile 命令。

如需 CloudFront 欄位層級加密的詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的使用欄位層級加密協助保護敏感資料。

```
aws cloudfront create-field-level-encryption-config \
    --field-level-encryption-config file://fle-config.json
```

檔案`fle-config.json`是目前資料夾中的 JSON 文件，其中包含下列項目：

```
{
    "CallerReference": "cli-example",
    "Comment": "Example FLE configuration",
    "QueryArgProfileConfig": {
        "ForwardWhenQueryArgProfileIsUnknown": true,
        "QueryArgProfiles": {
            "Quantity": 0
        }
    },
    "ContentTypeProfileConfig": {
        "ForwardWhenContentTypeIsUnknown": true,
        "ContentTypeProfiles": {
            "Quantity": 1,
            "Items": [
                {
                    "Format": "URLEncoded",
```

```
                    "ProfileId": "P280MFCLSYOCVU",
                    "ContentType": "application/x-www-form-urlencoded"
                }
            ]
        }
    }
}
```

輸出：

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/field-level-encryption/
C3KM2WVD605UAY",
    "ETag": "E2P4Z4VU7TY5SG",
    "FieldLevelEncryption": {
        "Id": "C3KM2WVD605UAY",
        "LastModifiedTime": "2019-12-10T21:30:18.974Z",
        "FieldLevelEncryptionConfig": {
            "CallerReference": "cli-example",
            "Comment": "Example FLE configuration",
            "QueryArgProfileConfig": {
                "ForwardWhenQueryArgProfileIsUnknown": true,
                "QueryArgProfiles": {
                    "Quantity": 0,
                    "Items": []
                }
            },
            "ContentTypeProfileConfig": {
                "ForwardWhenContentTypeIsUnknown": true,
                "ContentTypeProfiles": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "Format": "URLEncoded",
                            "ProfileId": "P280MFCLSYOCVU",
                            "ContentType": "application/x-www-form-urlencoded"
                        }
                    ]
                }
            }
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFieldLevelEncryptionConfig。

create-field-level-encryption-profile

以下程式碼範例顯示如何使用 create-field-level-encryption-profile。

AWS CLI

建立 CloudFront 欄位層級加密設定檔

下列範例會透過在名為 的 JSON 檔案中提供參數來建立欄位層級加密設定檔fle-profile-config.json。您必須先擁有 CloudFront 公有金鑰，才能建立欄位層級加密設定檔。若要建立 CloudFront 公有金鑰，請參閱 create-public-key 命令。

如需 CloudFront 欄位層級加密的詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的使用欄位層級加密協助保護敏感資料。

```
aws cloudfront create-field-level-encryption-profile \
    --field-level-encryption-profile-config file://fle-profile-config.json
```

檔案fle-profile-config.json是目前資料夾中的 JSON 文件，其中包含下列項目：

```
{
    "Name": "ExampleFLEProfile",
    "CallerReference": "cli-example",
    "Comment": "FLE profile for AWS CLI example",
    "EncryptionEntities": {
        "Quantity": 1,
        "Items": [
            {
                "PublicKeyId": "K2K8NC4HVFE3M0",
                "ProviderId": "ExampleFLEProvider",
                "FieldPatterns": {
                    "Quantity": 1,
                    "Items": [
                        "ExampleSensitiveField"
                    ]
                }
            }
        ]
    }
```

```
}
```

輸出：

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/field-level-encryption-
profile/PPK0UOSIF5WSV",
    "ETag": "E2QWRUHEXAMPLE",
    "FieldLevelEncryptionProfile": {
        "Id": "PPK0UOSIF5WSV",
        "LastModifiedTime": "2019-12-10T01:03:16.537Z",
        "FieldLevelEncryptionProfileConfig": {
            "Name": "ExampleFLEProfile",
            "CallerReference": "cli-example",
            "Comment": "FLE profile for AWS CLI example",
            "EncryptionEntities": {
                "Quantity": 1,
                "Items": [
                    {
                        "PublicKeyId": "K2K8NC4HVFE3M0",
                        "ProviderId": "ExampleFLEProvider",
                        "FieldPatterns": {
                            "Quantity": 1,
                            "Items": [
                                "ExampleSensitiveField"
                            ]
                        }
                    }
                ]
            }
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFieldLevelEncryptionProfile。

create-invalidation-for-distribution-tenant

以下程式碼範例顯示如何使用 create-invalidation-for-distribution-tenant。

AWS CLI

為分佈租用戶建立 CloudFront 無效

下列create-invalidation-for-distribution-tenant範例會為 CloudFront 分佈租用戶中的所有檔案建立失效。

```
aws cloudfront create-invalidation-for-distribution-tenant \
    --id dt_2wjDZi3hD1ivOXf6rpZJO1AB \
    --invalidation-batch '{"Paths": {"Quantity": 1, "Items": ["/*"]},
 "CallerReference": "invalidation-$(date +%s)"}'
```

輸出：

```
{
    "Location": "https://cloudfront.amazonaws.com/2020-05-31/distribution-tenant/
dt_2wjDZi3hD1ivOXf6rpZJO1AB/invalidation/I2JGL2F1ZAA426PGG0YLLKABC",
    "Invalidation": {
        "Id": "I2JGL2F1ZAA426PGG0YLLKABC",
        "Status": "InProgress",
        "CreateTime": "2025-05-07T16:59:25.947000+00:00",
        "InvalidationBatch": {
            "Paths": {
                "Quantity": 1,
                "Items": [
                    "/*"
                ]
            },
            "CallerReference": "invalidation-$(date +%s)"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的使檔案失效以移除內容。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateInvalidationForDistributionTenant。

create-invalidation

以下程式碼範例顯示如何使用 create-invalidation。

AWS CLI

為 CloudFront 分佈建立失效

下列create-invalidation範例會為指定 CloudFront 分佈中的指定檔案建立失效：

```
aws cloudfront create-invalidation \
    --distribution-id EDFDVBD6EXAMPLE \
    --paths "/example-path/example-file.jpg" "/example-path/example-file2.png"
```

輸出：

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EDFDVBD6EXAMPLE/invalidation/I1JLWSDAP8FU89",
    "Invalidation": {
        "Id": "I1JLWSDAP8FU89",
        "Status": "InProgress",
        "CreateTime": "2019-12-05T18:24:51.407Z",
        "InvalidationBatch": {
            "Paths": {
                "Quantity": 2,
                "Items": [
                    "/example-path/example-file2.png",
                    "/example-path/example-file.jpg"
                ]
            },
            "CallerReference": "cli-1575570291-670203"
        }
    }
}
```

在先前的範例中，CLI AWS 會自動產生隨機的 CallerReference。若要指定您自己的 CallerReference，或避免傳遞失效參數做為命令列引數，您可以使用 JSON 檔案。以下範例透過在名為 的 JSON 檔案中提供失效參數，為兩個檔案建立失效inv-batch.json：

```
aws cloudfront create-invalidation \
    --distribution-id EDFDVBD6EXAMPLE \
    --invalidation-batch file://inv-batch.json
```

inv-batch.json 的內容：

```
{
    "Paths": {
        "Quantity": 2,
        "Items": [
```

```
            "/example-path/example-file.jpg",
            "/example-path/example-file2.png"
        ]
    },
    "CallerReference": "cli-example"
}
```

輸出：

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EDFDVBD6EXAMPLE/invalidation/I2J0I21PCUYOIK",
    "Invalidation": {
        "Id": "I2J0I21PCUYOIK",
        "Status": "InProgress",
        "CreateTime": "2019-12-05T18:40:49.413Z",
        "InvalidationBatch": {
            "Paths": {
                "Quantity": 2,
                "Items": [
                    "/example-path/example-file.jpg",
                    "/example-path/example-file2.png"
                ]
            },
            "CallerReference": "cli-example"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateInvalidation。

`create-public-key`

以下程式碼範例顯示如何使用 `create-public-key`。

AWS CLI

建立 CloudFront 公有金鑰

下列範例透過在名為 的 JSON 檔案中提供參數來建立 CloudFront 公有金鑰`pub-key-config.json`。您必須先擁有 PEM 編碼的公有金鑰，才能使用此命令。如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的建立 RSA 金鑰對。

```
aws cloudfront create-public-key \
    --public-key-config file://pub-key-config.json
```

檔案pub-key-config.json是目前資料夾中的 JSON 文件，其中包含下列項目。請注意，公有金鑰是以 PEM 格式編碼。

```
{
    "CallerReference": "cli-example",
    "Name": "ExampleKey",
    "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----END
 PUBLIC KEY-----\n",
    "Comment": "example public key"
}
```

輸出：

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/public-key/
KDFB19YGCR002",
    "ETag": "E2QWRUHEXAMPLE",
    "PublicKey": {
        "Id": "KDFB19YGCR002",
        "CreatedTime": "2019-12-05T18:51:43.781Z",
        "PublicKeyConfig": {
            "CallerReference": "cli-example",
            "Name": "ExampleKey",
            "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----END
 PUBLIC KEY-----\n",
            "Comment": "example public key"
        }
```

```
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePublicKey。

delete-cloud-front-origin-access-identity

以下程式碼範例顯示如何使用 delete-cloud-front-origin-access-identity。

AWS CLI

刪除 CloudFront 原始存取身分

下列範例會刪除 ID 為 的原始存取身分 (OAI)E74FTE3AEXAMPLE。若要刪除 OAI，您必須具有 OAI 的 ID 和 ETag。OAI ID 會在 create-cloud-front-origin-access-identity 和 list-cloud-front-origin-access-identities 命令的輸出中傳回。若要取得 ETag，請使用 get-cloud-front-origin-access-identity 或 get-cloud-front-origin-access-identity-config 命令。使用 --if-match選項提供 OAI 的 ETag。

```
aws cloudfront delete-cloud-front-origin-access-identity \
    --id E74FTE3AEXAMPLE \
    --if-match E2QWRUHEXAMPLE
```

成功時，此命令沒有輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCloudFrontOriginAccessIdentity。

delete-connection-group

以下程式碼範例顯示如何使用 delete-connection-group。

AWS CLI

刪除連線群組

下列delete-connection-group範例會刪除連線群組。必須停用連線群組，且無法與任何 CloudFront 資源建立關聯。

```
aws cloudfront delete-connection-group \
    --id cg_2wjLpjbHkLUdhWAjHllcOeABC \
```

```
    --if-match ETVPDKIKX0DABC
```

成功時，此命令沒有輸出。

如需管理連線群組的詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的建立自訂連線群組（選用）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteConnectionGroup。

delete-distribution-tenant

以下程式碼範例顯示如何使用 delete-distribution-tenant。

AWS CLI

刪除分佈租用戶

下列delete-distribution-tenant範例會刪除具有 ETag 的分佈租用戶ETVPDKIKX0DABC。分發租用戶必須停用，且無法與任何 CloudFront 資源建立關聯。

```
aws cloudfront delete-distribution-tenant \
    --id dt_2wjMUbg3NHZEQ7OfoalP5zi1AB \
    --if-match ETVPDKIKX0DABC
```

成功時，此命令沒有輸出。

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的刪除分佈。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDistributionTenant。

delete-distribution

以下程式碼範例顯示如何使用 delete-distribution。

AWS CLI

刪除 CloudFront 分佈

下列範例會刪除 ID 為 的 CloudFront 分佈EDFDVBD6EXAMPLE。您必須先停用分發，才能刪除分發。若要停用分佈，請使用 update-distribution 命令。如需詳細資訊，請參閱 update-distribution 範例。

停用分佈時，您可以將其刪除。若要刪除分佈，您必須使用 `--if-match`選項來提供分佈的 `ETag`。若要取得 `ETag`，請使用 get-distribution 或 get-distribution-config 命令。

```
aws cloudfront delete-distribution \
    --id EDFDVBD6EXAMPLE \
    --if-match E2QWRUHEXAMPLE
```

成功時，此命令沒有輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDistribution。

`delete-field-level-encryption-config`

以下程式碼範例顯示如何使用 `delete-field-level-encryption-config`。

AWS CLI

刪除 CloudFront 欄位層級加密組態

下列範例會刪除 ID 為 的 CloudFront 欄位層級加密組態`C3KM2WVD605UAY`。若要刪除欄位層級加密組態，您必須擁有其 ID 和 `ETag`。ID 會在 create-field-level-encryption-config 和 list-field-level-encryption-configs 命令的輸出中傳回。若要取得 `ETag`，請使用 get-field-level-encryption 或 get-field-level-encryption-config 命令。使用 `--if-match`選項來提供組態的 `ETag`。

```
aws cloudfront delete-field-level-encryption-config \
    --id C3KM2WVD605UAY \
    --if-match E26M4BIAV81ZF6
```

成功時，此命令沒有輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFieldLevelEncryptionConfig。

`delete-field-level-encryption-profile`

以下程式碼範例顯示如何使用 `delete-field-level-encryption-profile`。

AWS CLI

刪除 CloudFront 欄位層級加密設定檔

下列範例會刪除 ID 為 的 CloudFront 欄位層級加密設定檔`PPK0UOSIF5WSV`。若要刪除欄位層級加密設定檔，您必須擁有其 ID 和 `ETag`。ID 會在 create-field-level-encryption-profile 和 list-field-

level-encryption-profiles 命令的輸出中傳回。若要取得 ETag，請使用 get-field-level-encryption-profile 或 get-field-level-encryption-profile-config 命令。使用 --if-match選項來提供設定檔的 ETag。

```
aws cloudfront delete-field-level-encryption-profile \
    --id PPK0UOSIF5WSV \
    --if-match EJETYFJ9CL66D
```

成功時，此命令沒有輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFieldLevelEncryptionProfile。

delete-public-key

以下程式碼範例顯示如何使用 delete-public-key。

AWS CLI

刪除 CloudFront 公有金鑰

下列範例會刪除 ID 為 的 CloudFront 公有金鑰KDFB19YGCR002。若要刪除公有金鑰，您必須擁有其 ID 和 ETag。ID 會在 create-public-key 和 list-public-keys 命令的輸出中傳回。若要取得 ETag，請使用 get-public-key 或 get-public-key-config 命令。使用 --if-match選項提供公有金鑰的 ETag。

```
aws cloudfront delete-public-key \
    --id KDFB19YGCR002 \
    --if-match E2QWRUHEXAMPLE
```

成功時，此命令沒有輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePublicKey。

disassociate-distribution-tenant-web-acl

以下程式碼範例顯示如何使用 disassociate-distribution-tenant-web-acl。

AWS CLI

取消 Web ACL 與分佈租用戶的關聯

下列disassociate-distribution-tenant-web-acl範例會取消 Web ACL 與分佈租用戶與 ETag 的關聯E1PA6795UKMABC。

```
aws cloudfront disassociate-distribution-tenant-web-acl \
    --id dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB \
    --if-match E1PA6795UKMABC
```

輸出：

```
{
    "ETag": "E13V1IB3VIYABC",
    "Id": "dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB"
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的停用 AWS WAF 安全保護。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateDistributionTenantWebAcl。

disassociate-distribution-web-acl

以下程式碼範例顯示如何使用 disassociate-distribution-web-acl。

AWS CLI

取消 Web ACL 與 CloudFront 分佈的關聯

下列disassociate-distribution-web-acl範例會移除 Web ACL 與具有 ETag 的 CloudFront 分佈之間的關聯E13V1IB3VIYABC。

```
aws cloudfront disassociate-distribution-web-acl \
    --id E1XNX8R2GOAABC \
    --if-match EEZQ9Z24VM1ABC
```

輸出：

```
{
    "ETag": "E2YWS1C2J3OABC",
    "Id": "E1XNX8R2GOAABC"
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的停用 AWS WAF 安全保護。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateDistributionWebAcl。

get-cloud-front-origin-access-identity-config

以下程式碼範例顯示如何使用 get-cloud-front-origin-access-identity-config。

AWS CLI

取得 CloudFront 原始存取身分組態

下列範例會使用 ID 取得有關 CloudFront 原始存取身分 (OAI) 的中繼資料E74FTE3AEXAMPLE，包括其 ETag。OAI ID 會在 create-cloud-front-origin-access-identity 和 list-cloud-front-origin-access-identities 命令的輸出中傳回。

```
aws cloudfront get-cloud-front-origin-access-identity-config --id E74FTE3AEXAMPLE
```

輸出：

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "CloudFrontOriginAccessIdentityConfig": {
        "CallerReference": "cli-example",
        "Comment": "Example OAI"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCloudFrontOriginAccessIdentityConfig。

get-cloud-front-origin-access-identity

以下程式碼範例顯示如何使用 get-cloud-front-origin-access-identity。

AWS CLI

取得 CloudFront 原始存取身分

下列範例會取得 ID 為 的 CloudFront 原始存取身分 (OAI)E74FTE3AEXAMPLE，包括其 ETag和相關聯的 S3 正式 ID。OAI ID 會在 create-cloud-front-origin-access-identity 和 list-cloud-front-origin-access-identities 命令的輸出中傳回。

```
aws cloudfront get-cloud-front-origin-access-identity --id E74FTE3AEXAMPLE
```

輸出：

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "CloudFrontOriginAccessIdentity": {
        "Id": "E74FTE3AEXAMPLE",
        "S3CanonicalUserId":
 "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
        "CloudFrontOriginAccessIdentityConfig": {
            "CallerReference": "cli-example",
            "Comment": "Example OAI"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCloudFrontOriginAccessIdentity。

get-connection-group-by-routing-endpoint

以下程式碼範例顯示如何使用 get-connection-group-by-routing-endpoint。

AWS CLI

依路由端點取得連線群組

下列get-connection-group-by-routing-endpoint範例會使用其路由端點擷取連線群組的相關資訊。

```
aws cloudfront get-connection-group-by-routing-endpoint \
    --routing-endpoint dvdg9gprgabc.cloudfront.net
```

輸出：

```
{
    "ETag": "E23ZP02F085ABC",
    "ConnectionGroup": {
        "Id": "cg_2wjDWTBKTlRB87cAaUQFaakABC",
        "Name": "connection-group-2",
```

```
        "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2wjDWTBKTlRB87cAaUQFaakABC",
        "CreatedTime": "2025-05-06T15:42:00.790000+00:00",
        "LastModifiedTime": "2025-05-06T15:42:00.790000+00:00",
        "Ipv6Enabled": true,
        "RoutingEndpoint": "dvdg9gprgabc.cloudfront.net",
        "Status": "Deployed",
        "Enabled": true,
        "IsDefault": false
    }
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的建立自訂連線群組（選用）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetConnectionGroupByRoutingEndpoint。

get-connection-group

以下程式碼範例顯示如何使用 get-connection-group。

AWS CLI

取得 CloudFront 連線群組

下列get-connection-group範例會擷取 CloudFront 連線群組的相關資訊。

```
aws cloudfront get-connection-group \
    --identifier cg_2wjDWTBKTlRB87cAaUQFaakABC
```

輸出：

```
{
    "ETag": "E23ZP02F085ABC",
    "ConnectionGroup": {
        "Id": "cg_2wjDWTBKTlRB87cAaUQFaakABC",
        "Name": "connection-group-2",
        "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2wjDWTBKTlRB87cAaUQFaakABC",
        "CreatedTime": "2025-05-06T15:42:00.790000+00:00",
        "LastModifiedTime": "2025-05-06T15:42:00.790000+00:00",
        "Ipv6Enabled": true,
        "RoutingEndpoint": "dvdg9gprgabc.cloudfront.net",
```

```
        "Status": "Deployed",
        "Enabled": true,
        "IsDefault": false
    }
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的建立自訂連線群組（選用）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetConnectionGroup。

get-distribution-config

以下程式碼範例顯示如何使用 get-distribution-config。

AWS CLI

取得 CloudFront 分佈組態

下列範例會使用 ID 取得有關 CloudFront 分佈的中繼資料EDFDVBD6EXAMPLE，包括其 ETag。分佈 ID 會在 create-distribution 和 list-distributions 命令中傳回。

```
aws cloudfront get-distribution-config \
    --id EDFDVBD6EXAMPLE
```

輸出：

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "DistributionConfig": {
        "CallerReference": "cli-example",
        "Aliases": {
            "Quantity": 0
        },
        "DefaultRootObject": "index.html",
        "Origins": {
            "Quantity": 1,
            "Items": [
                {
                    "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                    "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                    "OriginPath": "",
                    "CustomHeaders": {
                        "Quantity": 0
```

```
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "OriginGroups": {
        "Quantity": 0
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
        "ForwardedValues": {
            "QueryString": false,
            "Cookies": {
                "Forward": "none"
            },
            "Headers": {
                "Quantity": 0
            },
            "QueryStringCacheKeys": {
                "Quantity": 0
            }
        },
        "TrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ViewerProtocolPolicy": "allow-all",
        "MinTTL": 0,
        "AllowedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ],
            "CachedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ]
            }
        },
```

```
            "SmoothStreaming": false,
            "DefaultTTL": 86400,
            "MaxTTL": 31536000,
            "Compress": false,
            "LambdaFunctionAssociations": {
                "Quantity": 0
            },
            "FieldLevelEncryptionId": ""
        },
        "CacheBehaviors": {
            "Quantity": 0
        },
        "CustomErrorResponses": {
            "Quantity": 0
        },
        "Comment": "",
        "Logging": {
            "Enabled": false,
            "IncludeCookies": false,
            "Bucket": "",
            "Prefix": ""
        },
        "PriceClass": "PriceClass_All",
        "Enabled": true,
        "ViewerCertificate": {
            "CloudFrontDefaultCertificate": true,
            "MinimumProtocolVersion": "TLSv1",
            "CertificateSource": "cloudfront"
        },
        "Restrictions": {
            "GeoRestriction": {
                "RestrictionType": "none",
                "Quantity": 0
            }
        },
        "WebACLId": "",
        "HttpVersion": "http2",
        "IsIPV6Enabled": true
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDistributionConfig。

get-distribution-tenant-by-domain

以下程式碼範例顯示如何使用 get-distribution-tenant-by-domain。

AWS CLI

依網域取得分佈租用戶資訊

下列get-distribution-tenant-by-domain範例使用指定的網域擷取分佈租用戶的相關資訊。

```
aws cloudfront get-distribution-tenant-by-domain \
    --domain example.com
```

輸出：

```
{
    "ETag": "E23ZP02F085ABC",
    "DistributionTenant": {
        "Id": "dt_2xVInRKCfUzQHgxosDs9hiLk1AB",
        "DistributionId": "E1XNX8R2GOAABC",
        "Name": "example-tenant-4",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2xVInRKCfUzQHgxosDs9hiLk1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "active"
            }
        ],
        "Parameters": [
            {
                "Name": "testParam",
                "Value": "defaultValue"
            }
        ],
        "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
        "CreatedTime": "2025-05-23T16:16:20.871000+00:00",
        "LastModifiedTime": "2025-05-23T16:16:20.871000+00:00",
        "Enabled": false,
        "Status": "Deployed"
    }
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的了解多租戶分佈的運作方式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDistributionTenantByDomain。

get-distribution-tenant

以下程式碼範例顯示如何使用 get-distribution-tenant。

AWS CLI

取得 CloudFront 分佈租用戶的詳細資訊

下列get-distribution-tenant範例會擷取 CloudFront 分佈租用戶的相關資訊。

```
aws cloudfront get-distribution-tenant \
    --id dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB
```

輸出：

```
{
    "ETag": "E23ZP02F085ABC",
    "DistributionTenant": {
        "Id": "dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB",
        "DistributionId": "E1XNX8R2GOAABC",
        "Name": "example-tenant-2",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "inactive"
            }
        ],
        "ConnectionGroupId": "cg_2wjDWTBKTlRB87cAaUQFaakABC",
        "CreatedTime": "2025-05-06T15:42:28.542000+00:00",
        "LastModifiedTime": "2025-05-06T15:42:37.724000+00:00",
        "Enabled": true,
        "Status": "InProgress"
    }
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的了解多租戶分佈的運作方式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDistributionTenant。

get-distribution

以下程式碼範例顯示如何使用 get-distribution。

AWS CLI

取得 CloudFront 分佈

下列get-distribution範例會取得 ID 為 的 CloudFront 分佈EDFDVBD6EXAMPLE，包括其 ETag。分佈 ID 會在 create-distribution 和 list-distributions 命令中傳回。

```
aws cloudfront get-distribution \
    --id EDFDVBD6EXAMPLE
```

輸出：

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "Distribution": {
        "Id": "EDFDVBD6EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
        "Status": "Deployed",
        "LastModifiedTime": "2019-12-04T23:35:41.433Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-example",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                        "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
```

```
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-
example",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
```

```
                        "GET"
                    ]
                }
            },
            "SmoothStreaming": false,
            "DefaultTTL": 86400,
            "MaxTTL": 31536000,
            "Compress": false,
            "LambdaFunctionAssociations": {
                "Quantity": 0
            },
            "FieldLevelEncryptionId": ""
        },
        "CacheBehaviors": {
            "Quantity": 0
        },
        "CustomErrorResponses": {
            "Quantity": 0
        },
        "Comment": "",
        "Logging": {
            "Enabled": false,
            "IncludeCookies": false,
            "Bucket": "",
            "Prefix": ""
        },
        "PriceClass": "PriceClass_All",
        "Enabled": true,
        "ViewerCertificate": {
            "CloudFrontDefaultCertificate": true,
            "MinimumProtocolVersion": "TLSv1",
            "CertificateSource": "cloudfront"
        },
        "Restrictions": {
            "GeoRestriction": {
                "RestrictionType": "none",
                "Quantity": 0
            }
        },
        "WebACLId": "",
        "HttpVersion": "http2",
        "IsIPV6Enabled": true
    }
}
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDistribution。

get-field-level-encryption-config

以下程式碼範例顯示如何使用 get-field-level-encryption-config。

AWS CLI

若要取得 CloudFront 欄位層級加密組態的中繼資料

下列範例會使用 ID 取得有關 CloudFront 欄位層級加密組態的中繼資料C3KM2WVD605UAY，包括其 ETag：

```
aws cloudfront get-field-level-encryption-config --id C3KM2WVD605UAY
```

輸出：

```
{
    "ETag": "E2P4Z4VU7TY5SG",
    "FieldLevelEncryptionConfig": {
        "CallerReference": "cli-example",
        "Comment": "Example FLE configuration",
        "QueryArgProfileConfig": {
            "ForwardWhenQueryArgProfileIsUnknown": true,
            "QueryArgProfiles": {
                "Quantity": 0,
                "Items": []
            }
        },
        "ContentTypeProfileConfig": {
            "ForwardWhenContentTypeIsUnknown": true,
            "ContentTypeProfiles": {
                "Quantity": 1,
                "Items": [
                    {
                        "Format": "URLEncoded",
                        "ProfileId": "P280MFCLSYOCVU",
                        "ContentType": "application/x-www-form-urlencoded"
                    }
                ]
```

```
            }
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFieldLevelEncryptionConfig。

get-field-level-encryption-profile-config

以下程式碼範例顯示如何使用 get-field-level-encryption-profile-config。

AWS CLI

取得 CloudFront 欄位層級加密設定檔組態

下列範例取得 ID PPK0UOSIF5WSV 為 之 CloudFront 欄位層級加密設定檔的中繼資料，包括其 ETag：

```
aws cloudfront get-field-level-encryption-profile-config --id PPK0UOSIF5WSV
```

輸出：

```
{
    "ETag": "E1QQG65FS2L2GC",
    "FieldLevelEncryptionProfileConfig": {
        "Name": "ExampleFLEProfile",
        "CallerReference": "cli-example",
        "Comment": "FLE profile for AWS CLI example",
        "EncryptionEntities": {
            "Quantity": 1,
            "Items": [
                {
                    "PublicKeyId": "K2K8NC4HVFE3M0",
                    "ProviderId": "ExampleFLEProvider",
                    "FieldPatterns": {
                        "Quantity": 1,
                        "Items": [
                            "ExampleSensitiveField"
                        ]
                    }
                }
            ]
```

```
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFieldLevelEncryptionProfileConfig。

get-field-level-encryption-profile

以下程式碼範例顯示如何使用 get-field-level-encryption-profile。

AWS CLI

取得 CloudFront 欄位層級加密設定檔

下列範例取得 ID 為 PPK0UOSIF5WSV 的 CloudFront 欄位層級加密設定檔，包括其 ETag：

```
aws cloudfront get-field-level-encryption-profile --id PPK0UOSIF5WSV
```

輸出：

```
{
    "ETag": "E1QQG65FS2L2GC",
    "FieldLevelEncryptionProfile": {
        "Id": "PPK0UOSIF5WSV",
        "LastModifiedTime": "2019-12-10T01:03:16.537Z",
        "FieldLevelEncryptionProfileConfig": {
            "Name": "ExampleFLEProfile",
            "CallerReference": "cli-example",
            "Comment": "FLE profile for AWS CLI example",
            "EncryptionEntities": {
                "Quantity": 1,
                "Items": [
                    {
                        "PublicKeyId": "K2K8NC4HVFE3M0",
                        "ProviderId": "ExampleFLEProvider",
                        "FieldPatterns": {
                            "Quantity": 1,
                            "Items": [
                                "ExampleSensitiveField"
                            ]
                        }
```

```
                    }
                ]
            }
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFieldLevelEncryptionProfile。

get-field-level-encryption

以下程式碼範例顯示如何使用 get-field-level-encryption。

AWS CLI

取得 CloudFront 欄位層級加密組態

下列範例取得 ID 為 的 CloudFront 欄位層級加密組態C3KM2WVD605UAY，包括其 ETag：

```
aws cloudfront get-field-level-encryption --id C3KM2WVD605UAY
```

輸出：

```
{
    "ETag": "E2P4Z4VU7TY5SG",
    "FieldLevelEncryption": {
        "Id": "C3KM2WVD605UAY",
        "LastModifiedTime": "2019-12-10T21:30:18.974Z",
        "FieldLevelEncryptionConfig": {
            "CallerReference": "cli-example",
            "Comment": "Example FLE configuration",
            "QueryArgProfileConfig": {
                "ForwardWhenQueryArgProfileIsUnknown": true,
                "QueryArgProfiles": {
                    "Quantity": 0,
                    "Items": []
                }
            },
            "ContentTypeProfileConfig": {
                "ForwardWhenContentTypeIsUnknown": true,
                "ContentTypeProfiles": {
                    "Quantity": 1,
                    "Items": [
```

```
                        {
                            "Format": "URLEncoded",
                            "ProfileId": "P280MFCLSYOCVU",
                            "ContentType": "application/x-www-form-urlencoded"
                        }
                    ]
                }
            }
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFieldLevelEncryption。

get-invalidation-for-distribution-tenant

以下程式碼範例顯示如何使用 get-invalidation-for-distribution-tenant。

AWS CLI

取得分佈租用戶的失效

下列get-invalidation-for-distribution-tenant範例取得分佈租用戶失效的相關資訊。

```
aws cloudfront get-invalidation-for-distribution-tenant \
    --distribution-tenant-id dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB \
    --id I4CU23QAPKMUDUU06F9OFGFABC
```

輸出：

```
{
    "Invalidation": {
        "Id": "I4CU23QAPKMUDUU06F9OFGFABC",
        "Status": "Completed",
        "CreateTime": "2025-05-06T15:46:12.824000+00:00",
        "InvalidationBatch": {
            "Paths": {
                "Quantity": 2,
                "Items": [
                    "/example/invalidation",
                    "/more/invalidations"
                ]
            },
```

```
            "CallerReference": "007ee5a6-d0a0-42be-bb61-e7b915969b48"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的使檔案失效以移除內容。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInvalidationForDistributionTenant。

`get-invalidation`

以下程式碼範例顯示如何使用 `get-invalidation`。

AWS CLI

取得 CloudFront 無效

下列範例使用 ID `I2J0I21PCUYOIK`為 的 CloudFront 分佈 ID 取得失效`EDFDVBD6EXAMPLE`：

```
aws cloudfront get-invalidation --id I2J0I21PCUYOIK --distribution-
id EDFDVBD6EXAMPLE
```

輸出：

```
{
    "Invalidation": {
        "Status": "Completed",
        "InvalidationBatch": {
            "Paths": {
                "Items": [
                    "/example-path/example-file.jpg",
                    "/example-path/example-file-2.jpg"
                ],
                "Quantity": 2
            },
            "CallerReference": "cli-example"
        },
        "Id": "I2J0I21PCUYOIK",
        "CreateTime": "2019-12-05T18:40:49.413Z"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInvalidation。

get-managed-certificate-details

以下程式碼範例顯示如何使用 get-managed-certificate-details。

AWS CLI

若要取得受管憑證詳細資訊

下列get-managed-certificate-details範例會擷取 CloudFront 受管 ACM 憑證的詳細資訊。

```
aws cloudfront get-managed-certificate-details \
    --identifier dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB
```

輸出：

```
{
    "ManagedCertificateDetails": {
        "CertificateArn": "arn:aws:acm:us-
east-1:123456789012:certificate/655dc1fe-6d37-451d-a013-c2db3a034abc",
        "CertificateStatus": "pending-validation",
        "ValidationTokenHost": "self-hosted",
        "ValidationTokenDetails": [
            {
                "Domain": "example.com",
                "RedirectTo": "validation.us-east-1.acm-
validations.aws/123456789012/.well-known/pki-validation/
b315c9ae21284e7918bb9f3f422ab1c7.txt",
                "RedirectFrom": "example.com/.well-known/pki-validation/
b315c9ae21284e7918bb9f3f422ac3c7.txt"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的為您的 CloudFront 分發租用戶請求憑證。 Amazon CloudFront

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetManagedCertificateDetails。

get-public-key-config

以下程式碼範例顯示如何使用 get-public-key-config。

AWS CLI

取得 CloudFront 公有金鑰組態

下列範例會使用 ID 取得有關 CloudFront 公有金鑰的中繼資料KDFB19YGCR002，包括其 ETag。公有金鑰 ID 會在 create-public-key 和 list-public-keys 命令中傳回。

```
aws cloudfront get-public-key-config --id KDFB19YGCR002
```

輸出：

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "PublicKeyConfig": {
        "CallerReference": "cli-example",
        "Name": "ExampleKey",
        "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----END
 PUBLIC KEY-----\n",
        "Comment": "example public key"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPublicKeyConfig。

get-public-key

以下程式碼範例顯示如何使用 get-public-key。

AWS CLI

取得 CloudFront 公有金鑰

下列範例會取得 ID 為 的 CloudFront 公有金鑰KDFB19YGCR002，包括其 ETag。公有金鑰 ID 會在 create-public-key 和 list-public-keys 命令中傳回。

```
aws cloudfront get-public-key --id KDFB19YGCR002
```

輸出：

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "PublicKey": {
        "Id": "KDFB19YGCR002",
        "CreatedTime": "2019-12-05T18:51:43.781Z",
        "PublicKeyConfig": {
            "CallerReference": "cli-example",
            "Name": "ExampleKey",
            "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----END
 PUBLIC KEY-----\n",
            "Comment": "example public key"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPublicKey。

list-cloud-front-origin-access-identities

以下程式碼範例顯示如何使用 `list-cloud-front-origin-access-identities`。

AWS CLI

列出 CloudFront 原始存取身分

下列範例會取得您 AWS 帳戶中 CloudFront 原始存取身分 (OAIs的清單：

```
aws cloudfront list-cloud-front-origin-access-identities
```

輸出：

```
{
    "CloudFrontOriginAccessIdentityList": {
        "Items": [
```

```
            {
                "Id": "E74FTE3AEXAMPLE",
                "S3CanonicalUserId":
 "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
                "Comment": "Example OAI"
            },
            {
                "Id": "EH1HDMBEXAMPLE",
                "S3CanonicalUserId":
 "1489f6f2e6faacaae7ff64c4c3e6956c24f78788abfc1718c3527c263bf7a17EXAMPLE",
                "Comment": "Test OAI"
            },
            {
                "Id": "E2X2C9TEXAMPLE",
                "S3CanonicalUserId":
 "cbfeebb915a64749f9be546a45b3fcfd3a31c779673c13c4dd460911ae402c2EXAMPLE",
                "Comment": "Example OAI #2"
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCloudFrontOriginAccessIdentities。

`list-connection-groups`

以下程式碼範例顯示如何使用 `list-connection-groups`。

AWS CLI

列出連線群組

下列`list-connection-groups`範例列出您 AWS 帳戶中可用的連線群組。

```
aws cloudfront list-connection-groups
```

輸出：

```
{
    "ConnectionGroups": [
        {
            "Id": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
```

```
            "Name": "CreatedByCloudFront-cg_2whCJoXMYCjHcxaLGrkllvyABC",
            "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2whCJoXMYCjHcxaLGrkllvyABC",
            "RoutingEndpoint": "d3sx0pso7m5abc.cloudfront.net",
            "CreatedTime": "2025-05-05T22:32:29.630000+00:00",
            "LastModifiedTime": "2025-05-05T22:32:29.630000+00:00",
            "ETag": "E23ZP02F085ABC",
            "Enabled": true,
            "Status": "Deployed",
            "IsDefault": true
        },
        {
            "Id": "cg_2wjDWTBKTlRB87cAaUQFaakABC",
            "Name": "connection-group-2",
            "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2wjDWTBKTlRB87cAaUQFaakABC",
            "RoutingEndpoint": "dvdg9gprgabc.cloudfront.net",
            "CreatedTime": "2025-05-06T15:42:00.790000+00:00",
            "LastModifiedTime": "2025-05-06T15:42:00.790000+00:00",
            "ETag": "E23ZP02F085ABC",
            "Enabled": true,
            "Status": "Deployed",
            "IsDefault": false
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的建立自訂連線群組（選用）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListConnectionGroups。

list-distribution-tenants-by-customization

以下程式碼範例顯示如何使用 `list-distribution-tenants-by-customization`。

AWS CLI

依自訂列出分佈租用戶

下列`list-distribution-tenants-by-customization`範例列出使用指定 Web ACL 的分佈租用戶。

```
aws cloudfront list-distribution-tenants-by-customization \
```

```
    --web-acl-arn arn:aws:wafv2:us-east-1:123456789012:global/webacl/
CreatedByCloudFront-0273cd2f/a3c19bce-42b5-48a1-a8d4-b2bb2f28eabc
```

輸出：

```
{
    "DistributionTenantList": [
        {
            "Id": "dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB",
            "DistributionId": "E1XNX8R2GOAABC",
            "Name": "example-tenant-2",
            "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB",
            "Domains": [
                {
                    "Domain": "example.com",
                    "Status": "inactive"
                }
            ],
            "ConnectionGroupId": "cg_2wjDWTBKTlRB87cAaUQFaakABC",
            "Customizations": {
                "WebAcl": {
                    "Action": "override",
                    "Arn": "arn:aws:wafv2:us-east-1:123456789012:global/webacl/
CreatedByCloudFront-0273cd2f/a3c19bce-42b5-48a1-a8d4-b2bb2f28eabc"
                },
                "GeoRestrictions": {
                    "RestrictionType": "whitelist",
                    "Locations": [
                        "AL"
                    ]
                }
            },
            "CreatedTime": "2025-05-06T15:42:28.542000+00:00",
            "LastModifiedTime": "2025-05-06T16:14:08.710000+00:00",
            "ETag": "E1F83G8C2ARABC",
            "Enabled": true,
            "Status": "Deployed"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的分發租用戶自訂。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDistributionTenantsByCustomization。

`list-distribution-tenants`

以下程式碼範例顯示如何使用 `list-distribution-tenants`。

AWS CLI

列出 CloudFront 分佈租用戶

下列`list-distribution-tenants`範例會依關聯的連線群組列出您 AWS 帳戶中的 3 個 CloudFront 分佈租用戶。

```
aws cloudfront list-distribution-tenants \
    --association-filter ConnectionGroupId=cg_2whCJoXMYCjHcxaLGrkllvyABC \
    --max-items 3
```

輸出：

```
{
    "DistributionTenantList": [
        {
            "Id": "dt_2yMvQgam3QkJo2z54FDl91dk1AB",
            "DistributionId": "E1XNX8R2GOAABC",
            "Name": "new-tenant-customizations",
            "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2yMvQgam3QkJo2z54FDl91dk1AB",
            "Domains": [
                {
                    "Domain": "example.com",
                    "Status": "active"
                }
            ],
            "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
            "Customizations": {
                "WebAcl": {
                    "Action": "disable"
                },
                "GeoRestrictions": {
                    "RestrictionType": "whitelist",
                    "Locations": [
```

```
                        "DE"
                    ]
                }
            },
            "CreatedTime": "2025-06-11T15:54:02.142000+00:00",
            "LastModifiedTime": "2025-06-11T15:54:02.142000+00:00",
            "ETag": "E23ZP02F085ABC",
            "Enabled": false,
            "Status": "Deployed"
        },
        {
            "Id": "dt_2yMuV7NJuBcAB0cwwxMCBZQ1AB",
            "DistributionId": "E1XNX8R2GOAABC",
            "Name": "new-tenant",
            "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2yMuV7NJuBcAB0cwwxMCBZQ1AB",
            "Domains": [
                {
                    "Domain": "1.example.com",
                    "Status": "active"
                }
            ],
            "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
            "Customizations": {
                "GeoRestrictions": {
                    "RestrictionType": "whitelist",
                    "Locations": [
                        "DE"
                    ]
                }
            },
            "CreatedTime": "2025-06-11T15:46:23.466000+00:00",
            "LastModifiedTime": "2025-06-11T15:46:23.466000+00:00",
            "ETag": "E23ZP02F085ABC",
            "Enabled": false,
            "Status": "Deployed"
        },
        {
            "Id": "dt_2xVInRKCfUzQHgxosDs9hiLk1AB",
            "DistributionId": "E1XNX8R2GOAABC",
            "Name": "new-tenant-2",
            "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2xVInRKCfUzQHgxosDs9hiLk1AB",
            "Domains": [
```

```
                {
                    "Domain": "2.example.com",
                    "Status": "active"
                }
            ],
            "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
            "CreatedTime": "2025-05-23T16:16:20.871000+00:00",
            "LastModifiedTime": "2025-05-23T16:16:20.871000+00:00",
            "ETag": "E23ZP02F085ABC",
            "Enabled": false,
            "Status": "Deployed"
        }
    ],
    "NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAzfQ=="
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的了解多租戶分佈的運作方式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDistributionTenants。

list-distributions-by-connection-mode

以下程式碼範例顯示如何使用 `list-distributions-by-connection-mode`。

AWS CLI

依連線模式列出 CloudFront 分佈

下列`list-distributions-by-connection-mode`範例列出具有指定連線模式的 CloudFront 分佈。

```
aws cloudfront list-distributions-by-connection-mode \
    --connection-mode tenant-only
```

輸出：

```
{
    "DistributionList": {
        "Items": [
            {
                "Id": "E1XNX8R2GOAABC",
                "ARN": "arn:aws:cloudfront::123456789012:distribution/
E1XNX8R2GOAABC",
```

```
                "ETag": "EPT4JPJQDY1ABC",
                "Status": "Deployed",
                "LastModifiedTime": "2025-05-23T16:16:15.691000+00:00",
                "DomainName": "-",
                "Aliases": {
                    "Quantity": 0
                },
                "Origins": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "Id": "example-cfn-simple-distribution123",
                            "DomainName": "example.com",
                            "OriginPath": "",
                            "CustomHeaders": {
                                "Quantity": 0
                            },
                            "S3OriginConfig": {
                                "OriginAccessIdentity": ""
                            },
                            "ConnectionAttempts": 3,
                            "ConnectionTimeout": 10,
                            "OriginShield": {
                                "Enabled": false
                            },
                            "OriginAccessControlId": "E2CJRMB5LKEABC"
                        }
                    ]
                },
                "OriginGroups": {
                    "Quantity": 0
                },
                "DefaultCacheBehavior": {
                    "TargetOriginId": "example-cfn-simple-distribution123",
                    "TrustedKeyGroups": {
                        "Enabled": false,
                        "Quantity": 0
                    },
                    "ViewerProtocolPolicy": "allow-all",
                    "AllowedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
```

```
                        ],
                        "CachedMethods": {
                            "Quantity": 2,
                            "Items": [
                                "HEAD",
                                "GET"
                            ]
                        }
                    },
                    "Compress": true,
                    "LambdaFunctionAssociations": {
                        "Quantity": 0
                    },
                    "FunctionAssociations": {
                        "Quantity": 0
                    },
                    "FieldLevelEncryptionId": "",
                    "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5abc",
                    "GrpcConfig": {
                        "Enabled": false
                    }
                },
                "CacheBehaviors": {
                    "Quantity": 0
                },
                "CustomErrorResponses": {
                    "Quantity": 0
                },
                "Comment": "",
                "PriceClass": "PriceClass_All",
                "Enabled": true,
                "ViewerCertificate": {
                    "CloudFrontDefaultCertificate": false,
                    "ACMCertificateArn": "arn:aws:acm:us-
east-1:123456789012:certificate/ec53f564-ea5a-4e4a-a0a2-e3c989449abc",
                    "SSLSupportMethod": "sni-only",
                    "MinimumProtocolVersion": "TLSv1.2_2021",
                    "Certificate": "arn:aws:acm:us-east-1:123456789012:certificate/
ec53f564-ea5a-4e4a-a0a2-e3c989449abc",
                    "CertificateSource": "acm"
                },
                "Restrictions": {
                    "GeoRestriction": {
                        "RestrictionType": "none",
```

```
                        "Quantity": 0
                    }
                },
                "WebACLId": "arn:aws:wafv2:us-east-1:123456789012:global/webacl/web-
global-example/626900da-5f64-418b-ba9b-743f3746cabc",
                "HttpVersion": "http2",
                "IsIPV6Enabled": false,
                "Staging": false,
                "ConnectionMode": "tenant-only"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的建立自訂連線群組（選用）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDistributionsByConnectionMode。

`list-distributions`

以下程式碼範例顯示如何使用 `list-distributions`。

AWS CLI

列出 CloudFront 分佈

下列範例會取得您 AWS 帳戶中 CloudFront 分佈的清單。

```
aws cloudfront list-distributions
```

輸出：

```
{
    "DistributionList": {
        "Items": [
            {
                "Id": "E23YS8OEXAMPLE",
                "ARN": "arn:aws:cloudfront::123456789012:distribution/
E23YS8OEXAMPLE",
                "Status": "Deployed",
                "LastModifiedTime": "2024-08-05T18:23:40.375000+00:00",
                "DomainName": "abcdefgh12ijk.cloudfront.net",
                "Aliases": {
```

```
                    "Quantity": 0
                },
                "Origins": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                            "DomainName": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                            "OriginPath": "",
                            "CustomHeaders": {
                                "Quantity": 0
                            },
                            "S3OriginConfig": {
                                "OriginAccessIdentity": ""
                            },
                            "ConnectionAttempts": 3,
                            "ConnectionTimeout": 10,
                            "OriginShield": {
                                "Enabled": false
                            },
                            "OriginAccessControlId": "EIAP8PEXAMPLE"
                        }
                    ]
                },
                "OriginGroups": {
                    "Quantity": 0
                },
                "DefaultCacheBehavior": {
                    "TargetOriginId": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                    "TrustedSigners": {
                        "Enabled": false,
                        "Quantity": 0
                    },
                    "TrustedKeyGroups": {
                        "Enabled": false,
                        "Quantity": 0
                    },
                    "ViewerProtocolPolicy": "allow-all",
                    "AllowedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
```

```
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "Compress": true,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": "",
                "CachePolicyId": "658327ea-f89d-4fab-a63d-7e886EXAMPLE"
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "PriceClass": "PriceClass_All",
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "SSLSupportMethod": "vip",
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "HTTP2",
```

```
                "IsIPV6Enabled": true,
                "Staging": false
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDistributions。

`list-domain-conflicts`

以下程式碼範例顯示如何使用 `list-domain-conflicts`。

AWS CLI

列出 CloudFront 分佈的網域衝突

下列`list-domain-conflicts`範例列出 CloudFront 分佈的網域衝突。

```
aws cloudfront list-domain-conflicts \
    --domain example.com \
    --domain-control-validation-
resource "DistributionTenantId=dt_2x9GhoK0TZRsohWzv1b9It8J1AB"
```

輸出：

```
{
    "DomainConflicts": [
        {
            "Domain": "example.com",
            "ResourceType": "distribution-tenant",
            "ResourceId": "***************ohWzv1b9It8J1AB",
            "AccountId": "123456789012"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的將替代網域名稱移至不同的分佈。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDomainConflicts。

list-field-level-encryption-configs

以下程式碼範例顯示如何使用 list-field-level-encryption-configs。

AWS CLI

列出 CloudFront 欄位層級加密組態

下列範例會取得您 AWS 帳戶中 CloudFront 欄位層級加密組態的清單：

```
aws cloudfront list-field-level-encryption-configs
```

輸出：

```
{
    "FieldLevelEncryptionList": {
        "MaxItems": 100,
        "Quantity": 1,
        "Items": [
            {
                "Id": "C3KM2WVD605UAY",
                "LastModifiedTime": "2019-12-10T21:30:18.974Z",
                "Comment": "Example FLE configuration",
                "QueryArgProfileConfig": {
                    "ForwardWhenQueryArgProfileIsUnknown": true,
                    "QueryArgProfiles": {
                        "Quantity": 0,
                        "Items": []
                    }
                },
                "ContentTypeProfileConfig": {
                    "ForwardWhenContentTypeIsUnknown": true,
                    "ContentTypeProfiles": {
                        "Quantity": 1,
                        "Items": [
                            {
                                "Format": "URLEncoded",
                                "ProfileId": "P280MFCLSYOCVU",
                                "ContentType": "application/x-www-form-urlencoded"
                            }
                        ]
                    }
                }
```

```
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFieldLevelEncryptionConfigs。

list-field-level-encryption-profiles

以下程式碼範例顯示如何使用 `list-field-level-encryption-profiles`。

AWS CLI

列出 CloudFront 欄位層級加密設定檔

下列範例會取得您 AWS 帳戶中 CloudFront 欄位層級加密設定檔的清單：

```
aws cloudfront list-field-level-encryption-profiles
```

輸出：

```
{
    "FieldLevelEncryptionProfileList": {
        "MaxItems": 100,
        "Quantity": 2,
        "Items": [
            {
                "Id": "P280MFCLSYOCVU",
                "LastModifiedTime": "2019-12-05T01:05:39.896Z",
                "Name": "ExampleFLEProfile",
                "EncryptionEntities": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "PublicKeyId": "K2K8NC4HVFE3M0",
                            "ProviderId": "ExampleFLEProvider",
                            "FieldPatterns": {
                                "Quantity": 1,
                                "Items": [
                                    "ExampleSensitiveField"
                                ]
                            }
                        }
```

```
                    ]
                },
                "Comment": "FLE profile for AWS CLI example"
            },
            {
                "Id": "PPK0UOSIF5WSV",
                "LastModifiedTime": "2019-12-10T01:03:16.537Z",
                "Name": "ExampleFLEProfile2",
                "EncryptionEntities": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "PublicKeyId": "K2ABC10EXAMPLE",
                            "ProviderId": "ExampleFLEProvider2",
                            "FieldPatterns": {
                                "Quantity": 1,
                                "Items": [
                                    "ExampleSensitiveField2"
                                ]
                            }
                        }
                    ]
                },
                "Comment": "FLE profile #2 for AWS CLI example"
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFieldLevelEncryptionProfiles。

`list-invalidations-for-distribution-tenant`

以下程式碼範例顯示如何使用 `list-invalidations-for-distribution-tenant`。

AWS CLI

列出 CloudFront 分佈租用戶的失效

下列`list-invalidations-for-distribution-tenant`範例列出 CloudFront 分佈租用戶的失效。

```
aws cloudfront list-invalidations-for-distribution-tenant \
```

```
    --id dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB
```

輸出：

```
{
    "InvalidationList": {
        "Items": [
            {
                "Id": "I4CU23QAPKMUDUU06F9OFGFABC",
                "CreateTime": "2025-05-06T15:46:12.824000+00:00",
                "Status": "Completed"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的使檔案失效以移除內容。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListInvalidationsForDistributionTenant。

`list-invalidations`

以下程式碼範例顯示如何使用 `list-invalidations`。

AWS CLI

列出 CloudFront 無效

下列範例會取得 ID 為 的 CloudFront 分佈失效清單EDFDVBD6EXAMPLE：

```
aws cloudfront list-invalidations --distribution-id EDFDVBD6EXAMPLE
```

輸出：

```
{
    "InvalidationList": {
        "Marker": "",
        "Items": [
            {
                "Status": "Completed",
                "Id": "YNY2LI2BVJ4NJU",
```

```
                "CreateTime": "2019-08-31T21:15:52.042Z"
            }
        ],
        "IsTruncated": false,
        "MaxItems": 100,
        "Quantity": 1
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListInvalidations。

list-public-keys

以下程式碼範例顯示如何使用 `list-public-keys`。

AWS CLI

列出 CloudFront 公有金鑰

下列範例會取得您 AWS 帳戶中 CloudFront 公有金鑰的清單：

```
aws cloudfront list-public-keys
```

輸出：

```
{
    "PublicKeyList": {
        "MaxItems": 100,
        "Quantity": 2,
        "Items": [
            {
                "Id": "K2K8NC4HVFE3M0",
                "Name": "ExampleKey",
                "CreatedTime": "2019-12-05T01:04:28.818Z",
                "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----END
 PUBLIC KEY-----\n",
```

```
                "Comment": "example public key"
            },
            {
                "Id": "K1S0LWQ2L5HTBU",
                "Name": "ExampleKey2",
                "CreatedTime": "2019-12-09T23:28:11.110Z",
                "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEApOCAg88A8+f4dujn9Izt
\n26LxtgAkn2opGgo/NKpMiaisyw5qlg3f1gol7FV6pYNl78iJg3EO8JBbwtlH
+cR9\nLGSf60NDeVhm76Oc39Np/vWgOdsGQcRbi9WmKZeSODqjQGzVZWqPmito3FzWVk6b
\nfVY5N36U/RdbVAJm95Km+qaMYlbIdF40t72bi3IkKYV5hlB2XoDjlQ9F6ajQKyTB
\nMHa3SN8q+3ZjQ4sJJ7D1V6r4wR8jDcFVD5NckWJmmgIVnkOQM37NYeoDnkaOuTpu\nha/
+3b8tOb2z3LBVHPkp85zJRAOXacSwf5rZtPYKBNFsixTa2n55k2r218mOkMC4\nUwIDAQAB\n-----END
 PUBLIC KEY-----",
                "Comment": "example public key #2"
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPublicKeys。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出 CloudFront 分佈的標籤

下列範例會取得 CloudFront 分佈的標籤清單：

```
aws cloudfront list-tags-for-resource \
    --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE
```

輸出：

```
{
    "Tags": {
        "Items": [
            {
                "Key": "DateCreated",
```

```
                "Value": "2019-12-04"
            },
            {
                "Key": "Name",
                "Value": "Example name"
            },
            {
                "Key": "Project",
                "Value": "Example project"
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`sign`

以下程式碼範例顯示如何使用 `sign`。

AWS CLI

簽署 CloudFront URL

下列範例會簽署 CloudFront URL。若要簽署 URL，您需要金鑰對 ID（在 AWS 管理主控台中稱為存取金鑰 ID) 和信任簽署者的 CloudFront 金鑰對的私有金鑰。如需已簽署 URLs的詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的使用已簽署 URLs和已簽署 Cookie 提供私有內容。

```
aws cloudfront sign \
    --url https://d111111abcdef8.cloudfront.net/private-content/private-file.html \
    --key-pair-id APKAEIBAERJR2EXAMPLE \
    --private-key file://cf-signer-priv-key.pem \
    --date-less-than 2020-01-01
```

輸出：

```
https://d111111abcdef8.cloudfront.net/private-content/private-
file.html?Expires=1577836800&Signature=nEXK7Kby47XKeZQKVc6pwkif6oZc-
JWSpDkH0UH7EBGGqvgurkecCbgL5VfUAXyLQuJxFwRQWscz-
owcq9KpmewCXrXQbPaJZNi9XSNwf4YKurPDQYaRQawKoeenH0GFteRf9ELK-
```

```
Bs3nljTLjtbgzIUt7QJNKXcWr8AuUYikzGdJ4-qzx6WnxXfH~fxg4-
GGl6l2kgCpXUB6Jx6K~Y3kpVOdzUPOIqFLHAnJojbhxqrVejomZZ2XrquDvNUCCIbePGnR3d24UPaLXG4FKOqNEaWDIB
GNvjRJxqWf93uMobeMOiVYahb-e0KItiQewGcm0eLZQ__&Key-Pair-Id=APKAEIBAERJR2EXAMPLE
```

- 如需 API 詳細資訊，請參閱登入AWS CLI 命令參考。

`tag-resource`

以下程式碼範例顯示如何使用 `tag-resource`。

AWS CLI

標記 CloudFront 分佈

下列`tag-resource`範例會將兩個標籤新增至指定的 CloudFront 分佈。

```
aws cloudfront tag-resource \
    --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \
    --tags 'Items=[{Key=Name,Value="Example name"},{Key=Project,Value="Example
 project"}]'
```

您可以提供 JSON 檔案中的標籤，而不是使用命令列引數，如下列範例所示：

```
aws cloudfront tag-resource \
    --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \
    --tags file://tags.json
```

`tags.json` 的內容：

```
{
    "Items": [
        {
            "Key": "Name",
            "Value": "Example name"
        },
        {
            "Key": "Project",
            "Value": "Example project"
        }
    ]
}
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從 CloudFront 分佈中移除標籤

下列範例使用命令列引數從 CloudFront 分佈移除兩個標籤：

```
aws cloudfront untag-resource \
    --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \
    --tag-keys Items=Name,Project
```

您可以提供 JSON 檔案中的標籤索引鍵，而不是使用命令列引數，如下列範例所示：

```
aws cloudfront untag-resource \
    --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \
    --tag-keys file://tag-keys.json
```

檔案tag-keys.json是目前資料夾中的 JSON 文件，其中包含下列項目：

```
{
    "Items": [
        "Name",
        "Project"
    ]
}
```

成功時，此命令沒有輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-cloud-front-origin-access-identity

以下程式碼範例顯示如何使用 update-cloud-front-origin-access-identity。

AWS CLI

更新 CloudFront 原始存取身分

下列範例會使用 ID 更新原始存取身分 (OAI)E74FTE3AEXAMPLE。您可以更新的唯一欄位是 OAI 的 Comment。

若要更新 OAI，您必須擁有 OAI 的 ID 和 ETag。OAI ID 會在 create-cloud-front-origin-access-identity 和 list-cloud-front-origin-access-identities 命令的輸出中傳回。若要取得 ETag，請使用 get-cloud-front-origin-access-identity 或 get-cloud-front-origin-access-identity-config 命令。使用 --if-match選項來提供 OAI 的 ETag。

```
aws cloudfront update-cloud-front-origin-access-identity \
    --id E74FTE3AEXAMPLE \
    --if-match E2QWRUHEXAMPLE \
    --cloud-front-origin-access-identity-config \
        CallerReference=cli-example,Comment="Example OAI Updated"
```

您可以在 JSON 檔案中提供 OAI 組態來完成相同的作業，如下列範例所示：

```
aws cloudfront update-cloud-front-origin-access-identity \
    --id E74FTE3AEXAMPLE \
    --if-match E2QWRUHEXAMPLE \
    --cloud-front-origin-access-identity-config file://OAI-config.json
```

檔案OAI-config.json是目前目錄中的 JSON 文件，其中包含下列項目：

```
{
    "CallerReference": "cli-example",
    "Comment": "Example OAI Updated"
}
```

無論您是使用命令列引數或 JSON 檔案提供 OAI 組態，輸出都是相同的：

```
{
    "ETag": "E9LHASXEXAMPLE",
    "CloudFrontOriginAccessIdentity": {
        "Id": "E74FTE3AEXAMPLE",
        "S3CanonicalUserId":
 "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
        "CloudFrontOriginAccessIdentityConfig": {
```

```
            "CallerReference": "cli-example",
            "Comment": "Example OAI Updated"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateCloudFrontOriginAccessIdentity。

update-connection-group

以下程式碼範例顯示如何使用 update-connection-group。

AWS CLI

更新 CloudFront 連線群組

下列update-connection-group範例會停用 CloudFront 連線群組並停用 IPv6。

```
aws cloudfront update-connection-group \
    --id cg_2yHsDkcPKeUlVkk3aEgLKcjABC \
    --no-ipv6-enabled \
    --no-enabled \
    --if-match E3UN6WX5RRO2ABC
```

輸出：

```
{
    "ETag": "E1F83G8C2ARABC",
    "ConnectionGroup": {
        "Id": "cg_2yHsDkcPKeUlVkk3aEgLKcjABC",
        "Name": "cg-example",
        "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2yHsDkcPKeUlVkk3aEgLKcjABC",
        "CreatedTime": "2025-06-09T20:58:35.481000+00:00",
        "LastModifiedTime": "2025-06-11T16:25:54.280000+00:00",
        "Ipv6Enabled": false,
        "RoutingEndpoint": "du9xp1elo1abc.cloudfront.net",
        "Status": "InProgress",
        "Enabled": false,
        "IsDefault": false
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的建立自訂連線群組（選用）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateConnectionGroup。

update-distribution-tenant

以下程式碼範例顯示如何使用 update-distribution-tenant。

AWS CLI

更新 CloudFront 分佈租用戶

下列update-distribution-tenant範例會使用新的參數值更新 CloudFront 分佈租用戶，並將國家/地區新增至地理限制。

```
aws cloudfront update-distribution-tenant \
    --cli-input-json file://update-tenant.json
```

update-tenant.json 的內容：

```
{
    "Id": "dt_2yMvQgam3QkJo2z54FDl91dk1AB",
    "IfMatch": "E1F83G8C2ARABC",
    "Parameters": [
        {
            "Name": "testParam",
            "Value": "newParameterValue"
        }
    ],
    "Customizations": {
        "WebAcl": {
            "Action": "disable"
        },
        "GeoRestrictions": {
            "RestrictionType": "whitelist",
            "Locations": [
                "DE",
                "GB",
                "ES"
            ]
```

```
        }
    }
}
```

輸出：

```
{
    "ETag": "E1PA6795UKMABC",
    "DistributionTenant": {
        "Id": "dt_2yMvQgam3QkJo2z54FDl91dk1AB",
        "DistributionId": "E1XNX8R2GOAABC",
        "Name": "new-tenant-customizations",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2yMvQgam3QkJo2z54FDl91dk1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "active"
            }
        ],
        "Customizations": {
            "WebAcl": {
                "Action": "disable"
            },
            "GeoRestrictions": {
                "RestrictionType": "whitelist",
                "Locations": [
                    "DE",
                    "ES",
                    "GB"
                ]
            }
        },
        "Parameters": [
            {
                "Name": "testParam",
                "Value": "newParameterValue"
            }
        ],
        "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
        "CreatedTime": "2025-06-11T15:54:02.142000+00:00",
        "LastModifiedTime": "2025-06-11T16:42:45.531000+00:00",
        "Enabled": false,
```

```
        "Status": "InProgress"
    }
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的分發租用戶自訂。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDistributionTenant。

update-distribution

以下程式碼範例顯示如何使用 update-distribution。

AWS CLI

範例 1：更新 CloudFront 分發的預設根物件

下列範例會使用 ID 將 CloudFront 分佈的預設根物件更新index.html為 EDFDVBD6EXAMPLE。

```
aws cloudfront update-distribution \
    --id EDFDVBD6EXAMPLE \
    --default-root-object index.html
```

輸出：

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "Distribution": {
        "Id": "EDFDVBD6EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
        "Status": "InProgress",
        "LastModifiedTime": "2019-12-06T18:55:39.870Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "6b10378d-49be-4c4b-a642-419ccaf8f3b5",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
```

```
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "example-website",
                        "DomainName": "www.example.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "CustomOriginConfig": {
                            "HTTPPort": 80,
                            "HTTPSPort": 443,
                            "OriginProtocolPolicy": "match-viewer",
                            "OriginSslProtocols": {
                                "Quantity": 2,
                                "Items": [
                                    "SSLv3",
                                    "TLSv1"
                                ]
                            },
                            "OriginReadTimeout": 30,
                            "OriginKeepaliveTimeout": 5
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "example-website",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 1,
                        "Items": [
                            "*"
                        ]
                    },
                    "QueryStringCacheKeys": {
```

```
                    "Quantity": 0
                }
            },
            "TrustedSigners": {
                "Enabled": false,
                "Quantity": 0
            },
            "ViewerProtocolPolicy": "allow-all",
            "MinTTL": 0,
            "AllowedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ],
                "CachedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ]
                }
            },
            "SmoothStreaming": false,
            "DefaultTTL": 86400,
            "MaxTTL": 31536000,
            "Compress": false,
            "LambdaFunctionAssociations": {
                "Quantity": 0
            },
            "FieldLevelEncryptionId": ""
        },
        "CacheBehaviors": {
            "Quantity": 0
        },
        "CustomErrorResponses": {
            "Quantity": 0
        },
        "Comment": "",
        "Logging": {
            "Enabled": false,
            "IncludeCookies": false,
            "Bucket": "",
            "Prefix": ""
```

```
            },
            "PriceClass": "PriceClass_All",
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http1.1",
            "IsIPV6Enabled": true
        }
    }
}
```

範例 2：更新 CloudFront 分佈

下列範例會在名為 的 JSON 檔案中提供分佈組態EMLARXS9EXAMPLE，以停用具有 ID 的 CloudFront 分佈dist-config-disable.json。若要更新分佈，您必須使用 --if-match選項來提供分佈的 ETag。若要取得 ETag，請使用 get-distribution 或 get-distribution-config 命令。請注意， Enabled 欄位在 JSON 檔案中設定為 false 。

使用以下範例停用分佈之後，您可以使用 delete-distribution 命令將其刪除。

```
aws cloudfront update-distribution \
    --id EMLARXS9EXAMPLE \
    --if-match E2QWRUHEXAMPLE \
    --distribution-config file://dist-config-disable.json
```

dist-config-disable.json 的內容：

```
{
    "CallerReference": "cli-1574382155-496510",
    "Aliases": {
        "Quantity": 0
    },
    "DefaultRootObject": "index.html",
```

```
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-1574382155-273939",
                "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                "OriginPath": "",
                "CustomHeaders": {
                    "Quantity": 0
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "OriginGroups": {
        "Quantity": 0
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-1574382155-273939",
        "ForwardedValues": {
            "QueryString": false,
            "Cookies": {
                "Forward": "none"
            },
            "Headers": {
                "Quantity": 0
            },
            "QueryStringCacheKeys": {
                "Quantity": 0
            }
        },
        "TrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ViewerProtocolPolicy": "allow-all",
        "MinTTL": 0,
        "AllowedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
```

```
            ],
            "CachedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ]
            }
        },
        "SmoothStreaming": false,
        "DefaultTTL": 86400,
        "MaxTTL": 31536000,
        "Compress": false,
        "LambdaFunctionAssociations": {
            "Quantity": 0
        },
        "FieldLevelEncryptionId": ""
    },
    "CacheBehaviors": {
        "Quantity": 0
    },
    "CustomErrorResponses": {
        "Quantity": 0
    },
    "Comment": "",
    "Logging": {
        "Enabled": false,
        "IncludeCookies": false,
        "Bucket": "",
        "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": false,
    "ViewerCertificate": {
        "CloudFrontDefaultCertificate": true,
        "MinimumProtocolVersion": "TLSv1",
        "CertificateSource": "cloudfront"
    },
    "Restrictions": {
        "GeoRestriction": {
            "RestrictionType": "none",
            "Quantity": 0
        }
    },
```

```
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
}
```

輸出：

```
{
    "ETag": "E9LHASXEXAMPLE",
    "Distribution": {
        "Id": "EMLARXS9EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EMLARXS9EXAMPLE",
        "Status": "InProgress",
        "LastModifiedTime": "2019-12-06T18:32:35.553Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-1574382155-496510",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-
bucket.s3.amazonaws.com-1574382155-273939",
                        "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
                    }
                ]
            },
```

```
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-
bucket.s3.amazonaws.com-1574382155-273939",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
```

```
                },
                "FieldLevelEncryptionId": ""
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": false,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDistribution。

update-domain-association

以下程式碼範例顯示如何使用 update-domain-association。

AWS CLI

更新網域關聯

下列update-domain-association範例會使用 ETag 更新分佈租用戶的網域關聯E23ZP02F085ABC。

```
aws cloudfront update-domain-association \
    --domain example.com \
    --target-resource DistributionTenantId=dt_2x9GhoK0TZRsohWzv1b9It8J1AB \
    --if-match E23ZP02F085ABC
```

輸出：

```
{
    "ETag": "ETVPDKIKX0ABC",
    "Domain": "example.com",
    "ResourceId": "dt_2x9GhoK0TZRsohWzv1b9It8J1AB"
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的將替代網域名稱移至不同的分佈。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDomainAssociation。

update-field-level-encryption-config

以下程式碼範例顯示如何使用 update-field-level-encryption-config。

AWS CLI

更新 CloudFront 欄位層級加密組態

下列範例C3KM2WVD605UAY透過在 JSON 檔案中提供參數，使用 ID 更新Comment欄位層級加密組態的欄位。

若要更新欄位層級加密組態，您必須具有組態的 ID 和 ETag。ID 會在 create-field-level-encryption-config 和 list-field-level-encryption-configs 命令的輸出中傳回。若要取得 ETag，請使用 get-field-level-encryption 或 get-field-level-encryption-config 命令。使用 --if-match選項來提供組態的 ETag。

```
aws cloudfront update-field-level-encryption-config \
    --id C3KM2WVD605UAY \
    --if-match E2P4Z4VU7TY5SG \
    --field-level-encryption-config file://fle-config.json
```

檔案fle-config.json是目前目錄中的 JSON 文件，其中包含下列項目：

```
{
    "CallerReference": "cli-example",
    "Comment": "Updated example FLE configuration",
    "QueryArgProfileConfig": {
        "ForwardWhenQueryArgProfileIsUnknown": true,
        "QueryArgProfiles": {
            "Quantity": 0
        }
    },
    "ContentTypeProfileConfig": {
        "ForwardWhenContentTypeIsUnknown": true,
        "ContentTypeProfiles": {
            "Quantity": 1,
            "Items": [
                {
                    "Format": "URLEncoded",
                    "ProfileId": "P280MFCLSYOCVU",
                    "ContentType": "application/x-www-form-urlencoded"
                }
            ]
        }
    }
}
```

輸出：

```
{
    "ETag": "E26M4BIAV81ZF6",
    "FieldLevelEncryption": {
        "Id": "C3KM2WVD605UAY",
        "LastModifiedTime": "2019-12-10T22:26:26.170Z",
        "FieldLevelEncryptionConfig": {
            "CallerReference": "cli-example",
            "Comment": "Updated example FLE configuration",
            "QueryArgProfileConfig": {
```

```
                "ForwardWhenQueryArgProfileIsUnknown": true,
                "QueryArgProfiles": {
                    "Quantity": 0,
                    "Items": []
                }
            },
            "ContentTypeProfileConfig": {
                "ForwardWhenContentTypeIsUnknown": true,
                "ContentTypeProfiles": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "Format": "URLEncoded",
                            "ProfileId": "P280MFCLSYOCVU",
                            "ContentType": "application/x-www-form-urlencoded"
                        }
                    ]
                }
            }
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateFieldLevelEncryptionConfig。

update-field-level-encryption-profile

以下程式碼範例顯示如何使用 update-field-level-encryption-profile。

AWS CLI

更新 CloudFront 欄位層級加密設定檔

下列範例會使用 ID 更新欄位層級加密設定檔PPK0UOSIF5WSV。此範例會更新設定檔的Name和 ，Comment並透過在 JSON 檔案中提供參數來新增第二個FieldPatterns項目。

若要更新欄位層級加密設定檔，您必須具有設定檔的 ID 和 ETag。ID 會在 create-field-level-encryption-profile 和 list-field-level-encryption-profiles 命令的輸出中傳回。若要取得 ETag，請使用 get-field-level-encryption-profile 或 get-field-level-encryption-profile-config 命令。使用 --if-match選項來提供設定檔的 ETag。

```
aws cloudfront update-field-level-encryption-profile \
```

```
    --id PPK0UOSIF5WSV \
    --if-match E1QQG65FS2L2GC \
    --field-level-encryption-profile-config file://fle-profile-config.json
```

檔案fle-profile-config.json是目前目錄中的 JSON 文件，其中包含下列項目：

```
{
    "Name": "ExampleFLEProfileUpdated",
    "CallerReference": "cli-example",
    "Comment": "Updated FLE profile for AWS CLI example",
    "EncryptionEntities": {
        "Quantity": 1,
        "Items": [
            {
                "PublicKeyId": "K2K8NC4HVFE3M0",
                "ProviderId": "ExampleFLEProvider",
                "FieldPatterns": {
                    "Quantity": 2,
                    "Items": [
                        "ExampleSensitiveField",
                        "SecondExampleSensitiveField"
                    ]
                }
            }
        ]
    }
}
```

輸出：

```
{
    "ETag": "EJETYFJ9CL66D",
    "FieldLevelEncryptionProfile": {
        "Id": "PPK0UOSIF5WSV",
        "LastModifiedTime": "2019-12-10T19:05:58.296Z",
        "FieldLevelEncryptionProfileConfig": {
            "Name": "ExampleFLEProfileUpdated",
            "CallerReference": "cli-example",
            "Comment": "Updated FLE profile for AWS CLI example",
            "EncryptionEntities": {
                "Quantity": 1,
                "Items": [
                    {
```

```
                        "PublicKeyId": "K2K8NC4HVFE3M0",
                        "ProviderId": "ExampleFLEProvider",
                        "FieldPatterns": {
                            "Quantity": 2,
                            "Items": [
                                "ExampleSensitiveField",
                                "SecondExampleSensitiveField"
                            ]
                        }
                    }
                ]
            }
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateFieldLevelEncryptionProfile。

verify-dns-configuration

以下程式碼範例顯示如何使用 verify-dns-configuration。

AWS CLI

驗證網域的 DNS 組態

下列verify-dns-configuration範例會驗證網域的 DNS 組態。

```
aws cloudfront verify-dns-configuration \
    --domain example.com \
    --identifier dt_2x9GhoK0TZRsohWzv1b9It8J1AB
```

輸出：

```
{
    "DnsConfigurationList": [
        {
            "Domain": "example.com",
            "Status": "valid-configuration"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的將替代網域名稱移至不同的分佈。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 VerifyDnsConfiguration。

使用 的 Amazon CloudSearch 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon CloudSearch 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`upload-documents`

以下程式碼範例顯示如何使用 `upload-documents`。

AWS CLI

下列`upload-documents`命令會將一批 JSON 文件上傳至 Amazon CloudSearch 網域：

```
aws cloudsearchdomain upload-documents --endpoint-url https://doc-my-domain.us-west-1.cloudsearch.amazonaws.com --content-type application/json --documents document-batch.json
```

輸出：

```
{
  "status": "success",
  "adds": 5000,
  "deletes": 0
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UploadDocuments。

使用 的 CloudTrail 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 CloudTrail 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`add-tags`

以下程式碼範例顯示如何使用 `add-tags`。

AWS CLI

將標籤新增至線索

下列`add-tags`命令會新增 的標籤`Trail1`：

```
aws cloudtrail add-tags --resource-id arn:aws:cloudtrail:us-
east-1:123456789012:trail/Trail1 --tags-
list Key=name,Value=Alice Key=location,Value=us
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddTags。

`create-subscription`

以下程式碼範例顯示如何使用 `create-subscription`。

AWS CLI

建立和設定線索 AWS 的資源

下列`create-subscription`命令會為 建立新的 S3 儲存貯體和 SNS 主題`Trail1`。

```
aws cloudtrail create-subscription \
```

```
    --name Trail1 \
    --s3-new-bucket amzn-s3-demo-bucket \
    --sns-new-topic my-topic
```

輸出：

```
Setting up new S3 bucket amzn-s3-demo-bucket...
Setting up new SNS topic my-topic...
Creating/updating CloudTrail configuration...
CloudTrail configuration:
    {
        "trailList": [
            {
                "IncludeGlobalServiceEvents": true,
                "Name": "Trail1",
                "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/
Trail1",
                "LogFileValidationEnabled": false,
                "IsMultiRegionTrail": false,
                "S3BucketName": "amzn-s3-demo-bucket",
                "SnsTopicName": "my-topic",
                "HomeRegion": "us-east-1"
            }
        ],
        "ResponseMetadata": {
        "HTTPStatusCode": 200,
        "RequestId": "f39e51f6-c615-11e5-85bd-d35ca21ee3e2"
        }
    }
Starting CloudTrail service...
Logs will be delivered to my-bucket
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSubscription。

create-trail

以下程式碼範例顯示如何使用 create-trail。

AWS CLI

建立線索

下列create-trail範例會建立名為 的多區域線索，Trail1並指定 S3 儲存貯體。

```
aws cloudtrail create-trail \
    --name Trail1 \
    --s3-bucket-name amzn-s3-demo-bucket \
    --is-multi-region-trail
```

輸出：

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "Trail1",
    "TrailARN": "arn:aws:cloudtrail:us-west-2:123456789012:trail/Trail1",
    "LogFileValidationEnabled": false,
    "IsMultiRegionTrail": true,
    "S3BucketName": "amzn-s3-demo-bucket"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTrail。

delete-trail

以下程式碼範例顯示如何使用 delete-trail。

AWS CLI

刪除線索

下列delete-trail命令會刪除名為 的線索Trail1：

```
aws cloudtrail delete-trail --name Trail1
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTrail。

describe-trails

以下程式碼範例顯示如何使用 describe-trails。

AWS CLI

描述線索

下列describe-trails範例會傳回 Trail1和 的設定Trail2。

```
aws cloudtrail describe-trails \
    --trail-name-list Trail1 Trail2
```

輸出：

```
{
    "trailList": [
        {
            "IncludeGlobalServiceEvents": true,
            "Name": "Trail1",
            "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1",
            "LogFileValidationEnabled": false,
            "IsMultiRegionTrail": false,
            "S3BucketName": "amzn-s3-demo-bucket",
            "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/
CloudTrail_CloudWatchLogs_Role",
            "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:123456789012:log-
group:CloudTrail:*",
            "SnsTopicName": "my-topic",
            "HomeRegion": "us-east-1"
        },
        {
            "IncludeGlobalServiceEvents": true,
            "Name": "Trail2",
            "S3KeyPrefix": "my-prefix",
            "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail2",
            "LogFileValidationEnabled": false,
            "IsMultiRegionTrail": false,
            "S3BucketName": "amzn-s3-demo-bucket2",
            "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/4c5ae5ac-3c13-421e-8335-c7868ef6a769",
            "HomeRegion": "us-east-1"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTrails。

get-event-selectors

以下程式碼範例顯示如何使用 get-event-selectors。

AWS CLI

檢視線索的事件選取器設定

下列get-event-selectors命令會傳回 的設定Trail1：

```
aws cloudtrail get-event-selectors --trail-name Trail1
```

輸出：

```
{
  "EventSelectors": [
      {
          "IncludeManagementEvents": true,
          "DataResources": [],
          "ReadWriteType": "All"
      }
  ],
  "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetEventSelectors。

get-trail-status

以下程式碼範例顯示如何使用 get-trail-status。

AWS CLI

取得追蹤的狀態

下列get-trail-status命令會傳回 的交付和記錄詳細資訊Trail1：

```
aws cloudtrail get-trail-status --name Trail1
```

輸出：

```
{
  "LatestNotificationTime": 1454022144.869,
  "LatestNotificationAttemptSucceeded": "2016-01-28T23:02:24Z",
  "LatestDeliveryAttemptTime": "2016-01-28T23:02:24Z",
```

```
  "LatestDeliveryTime": 1454022144.869,
  "TimeLoggingStarted": "2015-11-06T18:36:38Z",
  "LatestDeliveryAttemptSucceeded": "2016-01-28T23:02:24Z",
  "IsLogging": true,
  "LatestCloudWatchLogsDeliveryTime": 1454022144.918,
  "StartLoggingTime": 1446834998.695,
  "StopLoggingTime": 1446834996.933,
  "LatestNotificationAttemptTime": "2016-01-28T23:02:24Z",
  "TimeLoggingStopped": "2015-11-06T18:36:36Z"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTrailStatus。

list-public-keys

以下程式碼範例顯示如何使用 list-public-keys。

AWS CLI

列出線索的所有公有金鑰

下列list-public-keys命令會傳回其私有金鑰用於在指定時間範圍內簽署摘要檔案的所有公有金鑰：

```
aws cloudtrail list-public-keys --start-time 2016-01-01T20:30:00.000Z
```

輸出：

```
{
  "PublicKeyList": [
      {
         "ValidityStartTime": 1453076702.0,
         "ValidityEndTime": 1455668702.0,
         "Value": "MIIBCgKCAQEAlSS3cl92HDycr/MTj0moOhas8habjrraXw+KzlWF0axSI2tcF
+3iJ9BKQAVSKxGwxwu3m0wG3J
+kUl1xboEcEPHYoIYMbgfSw7KGnuDKwkLzsQWhUJ0cIbOHASox1vv/5fNXkrHhGbDCHeVXm804c83nvHUEFYThr1PfyP
+4WGDk+BGH5m9iuiAKkipEHWmUl8/P7XpfpWQuk4h8g3pXZOrNXr08lbh4d39svj7UqdhvOXoBISp9t/
EXYuePGEtBdrKD9Dz+VHwyUPtBQvYr9BnkF88qBnaPNhS44rzwIDAQAB",
         "Fingerprint": "7f3f401420072e50a65a141430817ab3"
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPublicKeys。

list-tags

以下程式碼範例顯示如何使用 list-tags。

AWS CLI

列出線索的標籤

下列list-tags命令會列出 Trail1和 的標籤Trail2：

```
aws cloudtrail list-tags --resource-id-list arn:aws:cloudtrail:us-
east-1:123456789012:trail/Trail1 arn:aws:cloudtrail:us-east-1:123456789012:trail/
Trail2
```

輸出：

```
{
 "ResourceTagList": [
     {
         "ResourceId": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1",
         "TagsList": [
             {
                 "Value": "Alice",
                 "Key": "name"
             },
             {
                 "Value": "us",
                 "Key": "location"
             }
         ]
     },
     {
         "ResourceId": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail2",
         "TagsList": [
             {
                 "Value": "Bob",
                 "Key": "name"
             }
         ]
     }
  ]
```

```
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTags。

lookup-events

以下程式碼範例顯示如何使用 lookup-events。

AWS CLI

查詢追蹤的事件

下列lookup-events命令會依屬性 查詢 API 活動事件EventName：

```
aws cloudtrail lookup-events --lookup-
attributes AttributeKey=EventName,AttributeValue=ConsoleLogin
```

輸出：

```
{
  "Events": [
     {
         "EventId": "654ccbc0-ba0d-486a-9076-dbf7274677a7",
         "Username": "my-session-name",
         "EventTime": "2021-11-18T09:41:02-08:00",
         "CloudTrailEvent": "{\"eventVersion\":\"1.02\",\"userIdentity\":{\"type\":
\"AssumedRole\",\"principalId\":\"AROAJIKPFTA72SWU4L7T4:my-session-name\",\"arn\":
\"arn:aws:sts::123456789012:assumed-role/my-role/my-session-name\",\"accountId\":
\"123456789012\",\"sessionContext\":{\"attributes\":{\"mfaAuthenticated\":\"false
\",\"creationDate\":\"2016-01-26T21:42:12Z\"},\"sessionIssuer\":{\"type\":\"Role\",
\"principalId\":\"AROAJIKPFTA72SWU4L7T4\",\"arn\":\"arn:aws:iam::123456789012:role/
my-role\",\"accountId\":\"123456789012\",\"userName\":\"my-role\"}}},\"eventTime
\":\"2016-01-26T21:42:12Z\",\"eventSource\":\"signin.amazonaws.com\",\"eventName\":
\"ConsoleLogin\",\"awsRegion\":\"us-east-1\",\"sourceIPAddress\":\"72.21.198.70\",
\"userAgent\":\"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/47.0.2526.111 Safari/537.36\",\"requestParameters
\":null,\"responseElements\":{\"ConsoleLogin\":\"Success\"},\"additionalEventData\":
{\"MobileVersion\":\"No\",\"MFAUsed\":\"No\"},\"eventID\":\"654ccbc0-ba0d-486a-9076-
dbf7274677a7\",\"eventType\":\"AwsConsoleSignIn\",\"recipientAccountId\":
\"123456789012\"}",
         "EventName": "ConsoleLogin",
         "Resources": []
     }
```

```
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 LookupEvents。

put-event-selectors

以下程式碼範例顯示如何使用 put-event-selectors。

AWS CLI

範例 1：使用進階事件選取器設定追蹤記錄管理事件和資料事件

您可以為進階事件選取器新增進階事件選取器和條件，追蹤上所有條件和選取器的值上限為 500 個。您可以使用進階事件選取器來記錄所有可用的資料事件類型。您可以使用進階事件選取器或基本事件選取器，但不能同時使用兩者。如果您將進階事件選取器套用至追蹤，則會覆寫任何現有的基本事件選取器。

下列put-event-selectors範例會為名為 的線索建立進階事件選取器，myTrail以記錄所有管理事件、記錄除一個 S3 儲存貯體以外的所有 S3 PutObject 和 DeleteObject API 呼叫、記錄名為 之 Lambda 函數的資料 API 呼叫myFunction，以及在名為 的 SNS 主題上記錄發佈 API 呼叫myTopic。

```
aws cloudtrail put-event-selectors \
    --trail-name myTrail \
    --advanced-event-selectors '[{"Name": "Log all management events",
 "FieldSelectors": [{ "Field": "eventCategory", "Equals": ["Management"] }] },
{"Name": "Log PutObject and DeleteObject events for all but one
 bucket","FieldSelectors": [{ "Field": "eventCategory", "Equals": ["Data"] },
{ "Field": "resources.type", "Equals": ["AWS::S3::Object"] },{ "Field":
 "eventName", "Equals": ["PutObject","DeleteObject"] },{ "Field": "resources.ARN",
 "NotStartsWith": ["arn:aws:s3:::amzn-s3-demo-bucket/"] }]},{"Name": "Log
 data events for a specific Lambda function","FieldSelectors": [{ "Field":
 "eventCategory", "Equals": ["Data"] },{ "Field": "resources.type",
 "Equals": ["AWS::Lambda::Function"] },{ "Field": "resources.ARN", "Equals":
 ["arn:aws:lambda:us-east-1:123456789012:function:myFunction"] }]},{"Name":
 "Log all Publish API calls on a specific SNS topic","FieldSelectors":
 [{ "Field": "eventCategory", "Equals": ["Data"] },{ "Field": "resources.type",
 "Equals": ["AWS::SNS::Topic"] },{ "Field": "eventName", "Equals":
 ["Publish"] },{ "Field": "resources.ARN", "Equals": ["arn:aws:sns:us-
east-1:123456789012:myTopic.fifo"] }]}]'
```

輸出：

```
{
    "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/myTrail",
    "AdvancedEventSelectors": [
        {
            "Name": "Log all management events",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Management"
                    ]
                }
            ]
        },
        {
            "Name": "Log PutObject and DeleteObject events for all but one bucket",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::S3::Object"
                    ]
                },
                {
                    "Field": "eventName",
                    "Equals": [
                        "PutObject",
                        "DeleteObject"
                    ]
                },
                {
                    "Field": "resources.ARN",
                    "NotStartsWith": [
                        "arn:aws:s3:::amzn-s3-demo-bucket/"
                    ]
                }
```

```
            ]
        },
        {
            "Name": "Log data events for a specific Lambda function",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::Lambda::Function"
                    ]
                },
                {
                    "Field": "resources.ARN",
                    "Equals": [
                        "arn:aws:lambda:us-east-1:123456789012:function:myFunction"
                    ]
                }
            ]
        },
        {
            "Name": "Log all Publish API calls on a specific SNS topic",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::SNS::Topic"
                    ]
                },
                {
                    "Field": "eventName",
                    "Equals": [
                        "Publish"
```

```
                    ]
                },
                {
                    "Field": "resources.ARN",
                    "Equals": [
                        "arn:aws:sns:us-east-1:123456789012:myTopic.fifo"
                    ]
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《AWS CloudTrail 使用者指南》中的使用進階事件選取器記錄事件。

範例 2：設定線索的事件選取器，以記錄所有管理事件和資料事件

一個追蹤最多可以設定 5 個事件選取器和 250 個資料資源。事件選取器也稱為基本事件選取器。您可以使用事件選取器來記錄 S3 物件、Lambda 函數和 DynnamoDB 資料表的管理事件和資料事件。若要記錄其他資源類型的資料事件，您必須使用進階事件選取器。

下列put-event-selectors範例會為名為 的線索建立事件選取器TrailName，以包含所有管理事件、兩個 Amazon S3 儲存貯體/字首組合的資料事件，以及名為 之單一 AWS Lambda 函數的資料事件hello-world-python-function。

```
aws cloudtrail put-event-selectors \
    --trail-name TrailName \
    --event-selectors '[{"ReadWriteType": "All","IncludeManagementEvents":
 true,"DataResources": [{"Type":"AWS::S3::Object", "Values": ["arn:aws:s3:::amzn-
s3-demo-bucket/prefix","arn:aws:s3:::amzn-s3-demo-bucket2/prefix2"]},
{"Type": "AWS::Lambda::Function","Values": ["arn:aws:lambda:us-
west-2:999999999999:function:hello-world-python-function"]}]}]'
```

輸出：

```
{
    "EventSelectors": [
        {
            "IncludeManagementEvents": true,
            "DataResources": [
                {
                    "Values": [
```

```
                        "arn:aws:s3:::amzn-s3-demo-bucket/prefix",
                        "arn:aws:s3:::amzn-s3-demo-bucket2/prefix2"
                    ],
                    "Type": "AWS::S3::Object"
                },
                {
                    "Values": [
                        "arn:aws:lambda:us-west-2:123456789012:function:hello-world-
python-function"
                    ],
                    "Type": "AWS::Lambda::Function"
                },
            ],
            "ReadWriteType": "All"
        }
    ],
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName"
}
```

如需詳細資訊，請參閱《AWS CloudTrail 使用者指南》中的使用基本事件選取器記錄事件。

範例 3：設定線索的事件選取器，以記錄管理事件、S3 物件上的所有 S3 資料事件，以及您帳戶中函數上的所有 Lambda 資料事件

下列put-event-selectors範例會為名為 的線索建立事件選取器TrailName2，其中包含所有管理事件，以及 AWS 帳戶中所有 Amazon S3 儲存貯體和 AWS Lambda 函數的所有資料事件。

```
aws cloudtrail put-event-selectors \
    --trail-name TrailName2 \
    --event-selectors '[{"ReadWriteType": "All","IncludeManagementEvents":
 true,"DataResources": [{"Type":"AWS::S3::Object", "Values": ["arn:aws:s3"]},
{"Type": "AWS::Lambda::Function","Values": ["arn:aws:lambda"]}]}]'
```

輸出：

```
{
    "EventSelectors": [
        {
            "IncludeManagementEvents": true,
            "DataResources": [
                {
                    "Values": [
                        "arn:aws:s3"
```

```
                    ],
                    "Type": "AWS::S3::Object"
                },
                {
                    "Values": [
                        "arn:aws:lambda"
                    ],
                    "Type": "AWS::Lambda::Function"
                },
            ],
            "ReadWriteType": "All"
        }
    ],
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName2"
}
```

如需詳細資訊，請參閱《AWS CloudTrail 使用者指南》中的使用基本事件選取器記錄事件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutEventSelectors。

remove-tags

以下程式碼範例顯示如何使用 remove-tags。

AWS CLI

移除線索的標籤

下列remove-tags命令會移除 的指定標籤Trail1：

```
aws cloudtrail remove-tags --resource-id arn:aws:cloudtrail:us-
east-1:123456789012:trail/Trail1 --tags-list Key=name Key=location
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveTags。

start-logging

以下程式碼範例顯示如何使用 start-logging。

AWS CLI

開始記錄線索

下列start-logging命令會開啟 的記錄Trail1：

```
aws cloudtrail start-logging --name Trail1
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartLogging。

stop-logging

以下程式碼範例顯示如何使用 stop-logging。

AWS CLI

停止記錄線索

下列stop-logging命令會關閉 的記錄Trail1：

```
aws cloudtrail stop-logging --name Trail1
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopLogging。

update-subscription

以下程式碼範例顯示如何使用 update-subscription。

AWS CLI

更新線索的組態設定

下列update-subscription範例會更新線索以指定新的 S3 儲存貯體和 SNS 主題。

```
aws cloudtrail update-subscription \
    --name Trail1 \
    --s3-new-bucket amzn-s3-demo-bucket \
    --sns-new-topic my-topic-new
```

輸出：

```
Setting up new S3 bucket amzn-s3-demo-bucket...
Setting up new SNS topic my-topic-new...
Creating/updating CloudTrail configuration...
CloudTrail configuration:
```

```
{
    "trailList": [
        {
            "IncludeGlobalServiceEvents": true,
            "Name": "Trail1",
            "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1",
            "LogFileValidationEnabled": false,
            "IsMultiRegionTrail": false,
            "S3BucketName": "amzn-s3-demo-bucket",
            "SnsTopicName": "my-topic-new",
            "HomeRegion": "us-east-1"
        }
    ],
    "ResponseMetadata": {
        "HTTPStatusCode": 200,
        "RequestId": "31126f8a-c616-11e5-9cc6-2fd637936879"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSubscription。

update-trail

以下程式碼範例顯示如何使用 update-trail。

AWS CLI

更新線索

下列update-trail範例會更新追蹤，以使用現有的 儲存貯體進行日誌交付。

```
aws cloudtrail update-trail \
    --name Trail1 \
    --s3-bucket-name amzn-s3-demo-bucket
```

輸出：

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "Trail1",
    "TrailARN": "arn:aws:cloudtrail:us-west-2:123456789012:trail/Trail1",
    "LogFileValidationEnabled": false,
```

```
    "IsMultiRegionTrail": true,
    "S3BucketName": "amzn-s3-demo-bucket"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateTrail。

validate-logs

以下程式碼範例顯示如何使用 validate-logs。

AWS CLI

驗證日誌檔案

下列validate-logs命令會驗證 的日誌Trail1：

```
aws cloudtrail validate-logs --trail-arn arn:aws:cloudtrail:us-
east-1:123456789012:trail/Trail1 --start-time 20160129T19:00:00Z
```

輸出：

```
Validating log files for trail arn:aws:cloudtrail:us-east-1:123456789012:trail/
Trail1 between 2016-01-29T19:00:00Z and 2016-01-29T22:15:43Z
Results requested for 2016-01-29T19:00:00Z to 2016-01-29T22:15:43Z
Results found for 2016-01-29T19:24:57Z to 2016-01-29T21:24:57Z:
3/3 digest files valid
15/15 log files valid
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ValidateLogs。

使用 的 CloudWatch 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 CloudWatch 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

delete-alarms

以下程式碼範例顯示如何使用 delete-alarms。

AWS CLI

刪除警示

下列範例會使用 delete-alarms 命令刪除名為 "myalarm" 的 Amazon CloudWatch 警示：

```
aws cloudwatch delete-alarms --alarm-names myalarm
```

輸出：

```
This command returns to the prompt if successful.
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteAlarms。

delete-anomaly-detector

以下程式碼範例顯示如何使用 delete-anomaly-detector。

AWS CLI

刪除指定的異常偵測模型

下列delete-anomaly-detector範例會刪除指定帳戶中的異常偵測器模型。

```
aws cloudwatch delete-anomaly-detector \
    --namespace AWS/Logs \
    --metric-name IncomingBytes \
    --stat SampleCount
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的刪除異常偵測模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAnomalyDetector。

delete-dashboards

以下程式碼範例顯示如何使用 delete-dashboards。

AWS CLI

刪除指定的儀表板

下列delete-dashboards範例會刪除指定帳戶中名為 Dashboard-A和 Dashboard-B 的兩個儀表板。

```
aws cloudwatch delete-dashboards \
    --dashboard-names Dashboard-A Dashboard-B
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch 儀表板。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDashboards。

delete-insight-rules

以下程式碼範例顯示如何使用 delete-insight-rules。

AWS CLI

刪除指定的參與者洞見規則

下列delete-insight-rules範例會刪除指定帳戶中名為 Rule-A和 Rule-B的兩個參與者洞察規則。

```
aws cloudwatch delete-insight-rules \
    --rule-names Rule-A Rule-B
```

輸出：

```
{
    "Failures": []
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用 Contributor Insights 來分析高基數資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteInsightRules。

delete-metric-stream

以下程式碼範例顯示如何使用 delete-metric-stream。

AWS CLI

刪除指定的指標串流

下列delete-metric-stream範例會刪除指定帳戶中名為 QuickPartial-gSCKvO的指標串流。

```
aws cloudwatch delete-metric-stream \
    --name QuickPartial-gSCKvO
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用指標串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteMetricStream。

describe-alarm-history

以下程式碼範例顯示如何使用 describe-alarm-history。

AWS CLI

若要擷取警示的歷史記錄

下列範例會使用 describe-alarm-history 命令擷取名為 "myalarm" 的 Amazon CloudWatch 警示的歷史記錄：

```
aws cloudwatch describe-alarm-history --alarm-name "myalarm" --history-item-
type StateUpdate
```

輸出：

```
{
```

```
    "AlarmHistoryItems": [
        {
            "Timestamp": "2014-04-09T18:59:06.442Z",
            "HistoryItemType": "StateUpdate",
            "AlarmName": "myalarm",
            "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue\":
\"ALARM\",\"stateReason\":\"testing purposes\"},\"newState\":{\"stateValue\":\"OK
\",\"stateReason\":\"Threshold Crossed: 2 datapoints were not greater than the
 threshold (70.0). The most recent datapoints: [38.958, 40.292].\",\"stateReasonData
\":{\"version\":\"1.0\",\"queryDate\":\"2014-04-09T18:59:06.419+0000\",\"startDate
\":\"2014-04-09T18:44:00.000+0000\",\"statistic\":\"Average\",\"period\":300,
\"recentDatapoints\":[38.958,40.292],\"threshold\":70.0}}",
            "HistorySummary": "Alarm updated from ALARM to OK"
        },
        {
            "Timestamp": "2014-04-09T18:59:05.805Z",
            "HistoryItemType": "StateUpdate",
            "AlarmName": "myalarm",
            "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue
\":\"OK\",\"stateReason\":\"Threshold Crossed: 2 datapoints were
 not greater than the threshold (70.0). The most recent datapoints:
 [38.839999999999996, 39.714].\",\"stateReasonData\":{\"version\":
\"1.0\",\"queryDate\":\"2014-03-11T22:45:41.569+0000\",\"startDate\":
\"2014-03-11T22:30:00.000+0000\",\"statistic\":\"Average\",\"period\":300,
\"recentDatapoints\":[38.839999999999996,39.714],\"threshold\":70.0}},\"newState\":
{\"stateValue\":\"ALARM\",\"stateReason\":\"testing purposes\"}}",
            "HistorySummary": "Alarm updated from OK to ALARM"
        }
    ]
}
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeAlarmHistory。

describe-alarms-for-metric

以下程式碼範例顯示如何使用 describe-alarms-for-metric。

AWS CLI

顯示與指標相關聯的警示資訊

下列範例使用 describe-alarms-for-metric 命令，顯示與 Amazon EC2 CPUUtilization 指標和 ID 為 i-0c986c72 的執行個體相關聯之任何警示的資訊：

```
aws cloudwatch describe-alarms-for-metric --metric-name CPUUtilization --
namespace AWS/EC2 --dimensions Name=InstanceId,Value=i-0c986c72
```

輸出：

```
{
    "MetricAlarms": [
        {
            "EvaluationPeriods": 10,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:111122223333:alarm:myHighCpuAlarm2",
            "StateUpdatedTimestamp": "2013-10-30T03:03:51.479Z",
            "AlarmConfigurationUpdatedTimestamp": "2013-10-30T03:03:50.865Z",
            "ComparisonOperator": "GreaterThanOrEqualToThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:111122223333:NotifyMe"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2013-10-30T03:03:51.479+0000\",\"startDate\":\"2013-10-30T02:08:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":
[40.698,39.612,42.432,39.796,38.816,42.28,42.854,40.088,40.760000000000005,41.316],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myHighCpuAlarm2",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 10 datapoints were not greater than
 or equal to the threshold (70.0). The most recent datapoints: [40.760000000000005,
 41.316].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": true,
            "MetricName": "CPUUtilization"
        },
```

```
        {
            "EvaluationPeriods": 2,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:111122223333:alarm:myHighCpuAlarm",
            "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
            "AlarmConfigurationUpdatedTimestamp": "2014-04-09T22:26:05.958Z",
            "ComparisonOperator": "GreaterThanThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:111122223333:HighCPUAlarm"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myHighCpuAlarm",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 2 datapoints were not greater than
 the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": false,
            "MetricName": "CPUUtilization"
        }
    ]
}
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeAlarmsForMetric。

describe-alarms

以下程式碼範例顯示如何使用 describe-alarms。

AWS CLI

列出有關警示的資訊

下列範例使用 describe-alarms 命令提供名為 "myalarm" 的警示相關資訊：

```
aws cloudwatch describe-alarms --alarm-names "myalarm"
```

輸出：

```
{
    "MetricAlarms": [
        {
            "EvaluationPeriods": 2,
            "AlarmArn": "arn:aws:cloudwatch:us-east-1:123456789012:alarm:myalarm",
            "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
            "AlarmConfigurationUpdatedTimestamp": "2012-12-27T00:49:54.032Z",
            "ComparisonOperator": "GreaterThanThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:123456789012:myHighCpuAlarm"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myalarm",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 2 datapoints were not greater than
 the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": true,
```

```
            "MetricName": "CPUUtilization"
        }
    ]
}
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeAlarms。

describe-anomaly-detectors

以下程式碼範例顯示如何使用 describe-anomaly-detectors。

AWS CLI

擷取異常偵測模型的清單

下列describe-anomaly-detectors範例顯示與指定帳戶中AWS/Logs的命名空間相關聯的異常偵測器模型相關資訊。

```
aws cloudwatch describe-anomaly-detectors \
    --namespace AWS/Logs
```

輸出：

```
{
    "AnomalyDetectors": [
        {
            "Namespace": "AWS/Logs",
            "MetricName": "IncomingBytes",
            "Dimensions": [],
            "Stat": "SampleCount",
            "Configuration": {
                "ExcludedTimeRanges": []
            },
            "StateValue": "TRAINED",
            "SingleMetricAnomalyDetector": {
                "AccountId": "123456789012",
                "Namespace": "AWS/Logs",
                "MetricName": "IncomingBytes",
                "Dimensions": [],
                "Stat": "SampleCount"
            }
```

```
        },
        {
            "Namespace": "AWS/Logs",
            "MetricName": "IncomingBytes",
            "Dimensions": [
                {
                    "Name": "LogGroupName",
                    "Value": "demo"
                }
            ],
            "Stat": "Average",
            "Configuration": {
                "ExcludedTimeRanges": []
            },
            "StateValue": "PENDING_TRAINING",
            "SingleMetricAnomalyDetector": {
                "AccountId": "123456789012",
                "Namespace": "AWS/Logs",
                "MetricName": "IncomingBytes",
                "Dimensions": [
                    {
                        "Name": "LogGroupName",
                        "Value": "demo"
                    }
                ],
                "Stat": "Average"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用 CloudWatch 異常偵測。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAnomalyDetectors。

`describe-insight-rules`

以下程式碼範例顯示如何使用 `describe-insight-rules`。

AWS CLI

擷取 Contributor Insights 規則的清單

下列describe-insight-rules範例顯示指定帳戶中的所有 Contributor Insight 規則。

```
aws cloudwatch describe-insight-rules
```

輸出：

```
{
    "InsightRules": [
        {
            "Name": "Rule-A",
            "State": "ENABLED",
            "Schema": "CloudWatchLogRule/1",
            "Definition": "{\n\t\"AggregateOn\": \"Count\",\n\t\"Contribution\":
 {\n\t\t\"Filters\": [],\n\t\t\"Keys\": [\n\t\t\t\"$.requestId\"\n\t\t]\n\t},\n
\t\"LogFormat\": \"JSON\",\n\t\"Schema\": {\n\t\t\"Name\": \"CloudWatchLogRule
\",\n\t\t\"Version\": 1\n\t},\n\t\"LogGroupARNs\": [\n\t\t\"arn:aws:logs:us-
east-1:123456789012:log-group:demo\"\n\t]\n}",
            "ManagedRule": false
        },
        {
            "Name": "Rule-B",
            "State": "ENABLED",
            "Schema": "CloudWatchLogRule/1",
            "Definition": "{\n\t\"AggregateOn\": \"Count\",\n\t\"Contribution\":
 {\n\t\t\"Filters\": [],\n\t\t\"Keys\": [\n\t\t\t\"$.requestId\"\n\t\t]\n\t},\n
\t\"LogFormat\": \"JSON\",\n\t\"Schema\": {\n\t\t\"Name\": \"CloudWatchLogRule
\",\n\t\t\"Version\": 1\n\t},\n\t\"LogGroupARNs\": [\n\t\t\"arn:aws:logs:us-
east-1:123456789012:log-group:demo-1\"\n\t]\n}",
            "ManagedRule": false
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用 Contributor Insights 來分析高基數資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInsightRules。

disable-alarm-actions

以下程式碼範例顯示如何使用 disable-alarm-actions。

AWS CLI

停用警示的動作

下列範例使用 disable-alarm-actions 命令來停用名為 myalarm 之警示的所有動作：

```
aws cloudwatch disable-alarm-actions --alarm-names myalarm
```

如果成功，此命令會回到提示字元。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DisableAlarmActions。

disable-insight-rules

以下程式碼範例顯示如何使用 disable-insight-rules。

AWS CLI

停用指定的參與者洞見規則

下列disable-insight-rules範例會在指定的帳戶中停用兩個名為 Rule-A和 Rule-B的貢獻者洞察規則。

```
aws cloudwatch disable-insight-rules \
    --rule-names Rule-A Rule-B
```

輸出：

```
{
    "Failures": []
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用 Contributor Insights 來分析高基數資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableInsightRules。

enable-alarm-actions

以下程式碼範例顯示如何使用 enable-alarm-actions。

AWS CLI

啟用警示的所有動作

下列範例使用 enable-alarm-actions 命令來啟用名為 myalarm 之警示的所有動作：

```
aws cloudwatch enable-alarm-actions --alarm-names myalarm
```

如果成功，此命令會回到提示字元。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 EnableAlarmActions。

enable-insight-rules

以下程式碼範例顯示如何使用 enable-insight-rules。

AWS CLI

啟用指定的參與者洞見規則

下列enable-insight-rules範例會在指定的帳戶中啟用兩個名為 Rule-A和 Rule-B的貢獻者洞察規則。

```
aws cloudwatch enable-insight-rules \
    --rule-names Rule-A Rule-B
```

輸出：

```
{
    "Failures": []
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用 Contributor Insights 來分析高基數資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableInsightRules。

get-dashboard

以下程式碼範例顯示如何使用 get-dashboard。

AWS CLI

擷取儀表板的相關資訊

下列get-dashboard範例顯示Dashboard-A指定帳戶中名為 的儀表板相關資訊。

```
aws cloudwatch get-dashboard \
    --dashboard-name Dashboard-A
```

輸出：

```
{
    "DashboardArn": "arn:aws:cloudwatch::123456789012:dashboard/Dashboard-A",
    "DashboardBody": "{\"widgets\":[{\"type\":\"metric\",\"x\":0,\"y\":0,\"width
\":6,\"height\":6,\"properties\":{\"view\":\"timeSeries\",\"stacked\":false,
\"metrics\":[[\"AWS/EC2\",\"NetworkIn\",\"InstanceId\",\"i-0131f062232ade043\"],[\".
\",\"NetworkOut\",\".\",\".\"]],\"region\":\"us-east-1\"}}]}",
    "DashboardName": "Dashboard-A"
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch 儀表板。

Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDashboard。

get-insight-rule-report

以下程式碼範例顯示如何使用 get-insight-rule-report。

AWS CLI

擷取 Contributor Insights 規則所收集的時間序列資料

下列get-insight-rule-report範例會傳回 Contributor Insights 規則所收集的時間序列資料。

```
aws cloudwatch get-insight-rule-report \
    --rule-name Rule-A \
    --start-time 2024-10-13T20:15:00Z \
    --end-time 2024-10-13T20:30:00Z \
    --period 300
```

輸出：

```
{
    "KeyLabels": [
        "PartitionKey"
    ],
    "AggregationStatistic": "Sum",
    "AggregateValue": 0.5,
    "ApproximateUniqueCount": 1,
    "Contributors": [
        {
            "Keys": [
                "RequestID"
            ],
            "ApproximateAggregateValue": 0.5,
            "Datapoints": [
                {
                    "Timestamp": "2024-10-13T21:00:00+00:00",
                    "ApproximateValue": 0.5
                }
            ]
        }
    ],
    "RuleAttributes": []
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用 Contributor Insights 來分析高基數資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInsightRuleReport。

get-metric-data

以下程式碼範例顯示如何使用 get-metric-data。

AWS CLI

範例 1：使用數學表達式取得指定 EC2 的平均總 IOPS

下列get-metric-data範例i-abcdef使用結合 EBSReadOps和 指標的指標數學偏差，擷取具有 InstanceID 之 EC2 執行個體的 CloudWatch EBSWriteOps指標值。

```
aws cloudwatch get-metric-data \
    --metric-data-queries file://file.json \
```

```
    --start-time 2024-09-29T22:10:00Z \
    --end-time 2024-09-29T22:15:00Z
```

file.json 的內容：

```
[
    {
        "Id": "m3",
        "Expression": "(m1+m2)/300",
        "Label": "Avg Total IOPS"
    },
    {
        "Id": "m1",
        "MetricStat": {
            "Metric": {
                "Namespace": "AWS/EC2",
                "MetricName": "EBSReadOps",
                "Dimensions": [
                    {
                        "Name": "InstanceId",
                        "Value": "i-abcdef"
                    }
                ]
            },
            "Period": 300,
            "Stat": "Sum",
            "Unit": "Count"
        },
        "ReturnData": false
    },
    {
        "Id": "m2",
        "MetricStat": {
            "Metric": {
                "Namespace": "AWS/EC2",
                "MetricName": "EBSWriteOps",
                "Dimensions": [
                    {
                        "Name": "InstanceId",
                        "Value": "i-abcdef"
                    }
                ]
            },
```

```
            "Period": 300,
            "Stat": "Sum",
            "Unit": "Count"
        },
        "ReturnData": false
    }
]
```

輸出：

```
{
    "MetricDataResults": [
        {
            "Id": "m3",
            "Label": "Avg Total IOPS",
            "Timestamps": [
                "2024-09-29T22:10:00+00:00"
            ],
            "Values": [
                96.85
            ],
            "StatusCode": "Complete"
        }
    ],
    "Messages": []
}
```

範例 2：使用 CloudWatch 帳單指標監控預估 AWS 費用

下列get-metric-data範例會從 AWS/計費命名空間擷取 EstimatedCharges CloudWatch 指標。

```
aws cloudwatch get-metric-data \
    --metric-data-queries '[{"Id":"m1","MetricStat":{"Metric":
{"Namespace":"AWS/Billing","MetricName":"EstimatedCharges","Dimensions":
[{"Name":"Currency","Value":"USD"}]},"Period":21600,"Stat":"Maximum"}}]' \
    --start-time 2024-09-26T12:00:00Z \
    --end-time 2024-09-26T18:00:00Z \
    --region us-east-1
```

輸出：

```
{
    "MetricDataResults": [
        {
            "Id": "m1",
            "Label": "EstimatedCharges",
            "Timestamps": [
                "2024-09-26T12:00:00+00:00"
            ],
            "Values": [
                542.38
            ],
            "StatusCode": "Complete"
        }
    ],
    "Messages": []
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的搭配 CloudWatch 指標使用數學表達式。 Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMetricData。

get-metric-statistics

以下程式碼範例顯示如何使用 get-metric-statistics。

AWS CLI

取得每個 EC2 執行個體的 CPU 使用率

下列範例使用 get-metric-statistics 命令來取得 ID 為 i-abcdef 之 EC2 執行個體的 CPU 使用率。

```
aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-
time 2014-04-08T23:18:00Z --end-time 2014-04-09T23:18:00Z --period 3600 --
namespace AWS/EC2 --statistics Maximum --dimensions Name=InstanceId,Value=i-abcdef
```

輸出：

```
{
    "Datapoints": [
        {
```

```
            "Timestamp": "2014-04-09T11:18:00Z",
            "Maximum": 44.79,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T20:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T19:18:00Z",
            "Maximum": 50.85,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T09:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T03:18:00Z",
            "Maximum": 76.84,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T21:18:00Z",
            "Maximum": 48.96,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T14:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T08:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T16:18:00Z",
            "Maximum": 45.55,
            "Unit": "Percent"
        },
```

```
        {
            "Timestamp": "2014-04-09T06:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T13:18:00Z",
            "Maximum": 45.08,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T05:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T18:18:00Z",
            "Maximum": 46.88,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T17:18:00Z",
            "Maximum": 52.08,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T07:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T02:18:00Z",
            "Maximum": 51.23,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T12:18:00Z",
            "Maximum": 47.67,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-08T23:18:00Z",
            "Maximum": 46.88,
            "Unit": "Percent"
```

```
        },
        {
            "Timestamp": "2014-04-09T10:18:00Z",
            "Maximum": 51.91,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T04:18:00Z",
            "Maximum": 47.13,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T15:18:00Z",
            "Maximum": 48.96,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T00:18:00Z",
            "Maximum": 48.16,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T01:18:00Z",
            "Maximum": 49.18,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```

指定多個維度

下列範例說明如何指定多個維度。每個維度都指定為「名稱/值」對，名稱和值之間用逗號分隔。使用空格分隔多個維度。如果單一指標包含多個維度，則必須為每個定義的維度指定值。

如需有關使用 `get-metric-statistics` 命令的更多範例，請參閱《Amazon CloudWatch 開發人員指南》中的「取得指標的統計資料」。

```
aws cloudwatch get-metric-statistics --metric-name Buffers --namespace MyNameSpace
 --dimensions Name=InstanceID,Value=i-abcdef Name=InstanceType,Value=m1.small --
start-time 2016-10-15T04:00:00Z --end-time 2016-10-19T07:00:00Z --statistics Average
 --period 60
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetMetricStatistics。

get-metric-stream

以下程式碼範例顯示如何使用 get-metric-stream。

AWS CLI

擷取指標串流的相關資訊

下列get-metric-stream範例顯示QuickFull-GuaFbs指定帳戶中名為 的指標串流的相關資訊。

```
aws cloudwatch get-metric-stream \
    --name QuickFull-GuaFbs
```

輸出：

```
{
    "Arn": "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/QuickFull-
GuaFbs",
    "Name": "QuickFull-GuaFbs",
    "FirehoseArn": "arn:aws:firehose:us-east-1:123456789012:deliverystream/
MetricStreams-QuickFull-GuaFbs-WnySbECG",
    "RoleArn": "arn:aws:iam::123456789012:role/service-role/MetricStreams-
FirehosePutRecords-JN10W9B3",
    "State": "running",
    "CreationDate": "2024-10-11T18:48:59.187000+00:00",
    "LastUpdateDate": "2024-10-11T18:48:59.187000+00:00",
    "OutputFormat": "json",
    "IncludeLinkedAccountsMetrics": false
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用指標串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMetricStream。

get-metric-widget-image

以下程式碼範例顯示如何使用 get-metric-widget-image。

AWS CLI

擷取 CPUUtilization 的快照圖形

下列get-metric-widget-image範例會使用 ID 擷取 EC2 執行個體指標CPUUtilization的快照圖形，i-abcde並將擷取的影像儲存為本機電腦上名為 "image.png" 的檔案。

```
aws cloudwatch get-metric-widget-image \
    --metric-widget '{"metrics":[["AWS/EC2","CPUUtilization","InstanceId","i-
abcde"]]}' \
    --output-format png \
    --output text | base64 --decode > image.png
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMetricWidgetImage。

list-dashboards

以下程式碼範例顯示如何使用 list-dashboards。

AWS CLI

擷取儀表板清單

下列list-dashboards範例列出指定帳戶中的所有儀表板。

```
aws cloudwatch list-dashboards
```

輸出：

```
{
    "DashboardEntries": [
        {
            "DashboardName": "Dashboard-A",
            "DashboardArn": "arn:aws:cloudwatch::123456789012:dashboard/Dashboard-
A",
            "LastModified": "2024-10-11T18:40:11+00:00",
            "Size": 271
        },
        {
            "DashboardName": "Dashboard-B",
```

```
            "DashboardArn": "arn:aws:cloudwatch::123456789012:dashboard/Dashboard-
B",
            "LastModified": "2024-10-11T18:44:41+00:00",
            "Size": 522
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch 儀表板。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDashboards。

`list-metric-streams`

以下程式碼範例顯示如何使用 `list-metric-streams`。

AWS CLI

擷取指標串流清單

下列`list-metric-streams`範例列出指定帳戶中的所有指標串流。

```
aws cloudwatch list-metric-streams
```

輸出：

```
{
    "Entries": [
        {
            "Arn": "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/
QuickFull-GuaFbs",
            "CreationDate": "2024-10-11T18:48:59.187000+00:00",
            "LastUpdateDate": "2024-10-11T18:48:59.187000+00:00",
            "Name": "QuickFull-GuaFbs",
            "FirehoseArn": "arn:aws:firehose:us-east-1:123456789012:deliverystream/
MetricStreams-QuickFull-GuaFbs-WnySbECG",
            "State": "running",
            "OutputFormat": "json"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用指標串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMetricStreams。

`list-metrics`

以下程式碼範例顯示如何使用 `list-metrics`。

AWS CLI

列出 Amazon SNS 的指標

下列 `list-metrics` 範例顯示 Amazon SNS 的指標。

```
aws cloudwatch list-metrics \
    --namespace "AWS/SNS"
```

輸出：

```
{
    "Metrics": [
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "PublishSize"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "PublishSize"
        },
        {
            "Namespace": "AWS/SNS",
```

```
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfNotificationsFailed"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfNotificationsDelivered"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfMessagesPublished"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfMessagesPublished"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
```

```
                }
            ],
            "MetricName": "NumberOfNotificationsDelivered"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfNotificationsFailed"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListMetrics。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出與現有警示相關聯的標籤*

下列`list-tags-for-resource`範例列出與demo指定帳戶中名為 的警示相關聯的所有標籤。

```
aws cloudwatch list-tags-for-resource \
    --resource-arn arn:aws:cloudwatch:us-east-1:123456789012:alarm:demo
```

輸出：

```
{
    "Tags": [
        {
            "Key": "stack",
            "Value": "Production"
        },
        {
            "Key": "team",
```

```
            "Value": "Devops"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的警示和標記。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

put-anomaly-detector

以下程式碼範例顯示如何使用 put-anomaly-detector。

AWS CLI

建立異常偵測模型

下列put-anomaly-detector範例會為 CloudWatch 指標建立異常偵測模型。

```
aws cloudwatch put-anomaly-detector \
    --namespace AWS/Logs \
    --metric-name IncomingBytes \
    --stat SampleCount
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用 CloudWatch 異常偵測。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutAnomalyDetector。

put-composite-alarm

以下程式碼範例顯示如何使用 put-composite-alarm。

AWS CLI

建立複合 cloudwatch 警示

下列put-composite-alarm範例ProdAlarm會在指定的帳戶中建立名為 的複合警示。

```
aws cloudwatch put-composite-alarm \
    --alarm-name ProdAlarm \
```

```
    --alarm-rule "ALARM(CPUUtilizationTooHigh) AND ALARM(MemUsageTooHigh)" \
    --alarm-actions arn:aws:sns:us-east-1:123456789012:demo \
    --actions-enabled
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的建立複合警示。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutCompositeAlarm。

put-dashboard

以下程式碼範例顯示如何使用 put-dashboard。

AWS CLI

建立儀表板

下列put-dashboard範例Dashboard-A會在指定的帳戶中建立名為 的儀表板。

```
aws cloudwatch put-dashboard \
    --dashboard-name Dashboard-A \
    --dashboard-body '{"widgets":
[{"height":6,"width":6,"y":0,"x":0,"type":"metric","properties":
{"view":"timeSeries","stacked":false,"metrics":
[["Namespace","CPUUtilization","Environment","Prod","Type","App"]],"region":"us-
east-1"}}]}'
```

輸出：

```
{
    "DashboardValidationMessages": []
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的建立 CloudWatch 儀表板。
Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutDashboard。

put-insight-rule

以下程式碼範例顯示如何使用 put-insight-rule。

AWS CLI

建立參與者洞見規則

下列put-insight-rule範例VPCFlowLogsContributorInsights會在指定的帳戶中建立名為 的 Contributor Insights 規則。

```
aws cloudwatch put-insight-rule \
    --rule-name VPCFlowLogsContributorInsights \
    --rule-definition file://insight-rule.json \
    --rule-state ENABLED
```

insight-rule.json 的內容：

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Count",
    "Contribution": {
        "Filters": [],
        "Keys": [
            "tcp-flag"
        ]
    },
    "LogFormat": "CLF",
    "LogGroupNames": [
        "/vpc/flowlogs/*"
    ],
    "Fields": {
        "23": "tcp-flag"
    }
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的在 CloudWatch 中建立 Contributor Insights 規則。 Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutInsightRule。

put-metric-alarm

以下程式碼範例顯示如何使用 put-metric-alarm。

AWS CLI

在 CPU 使用率超過 70% 時傳送 Amazon Simple Notification Service 電子郵件訊息

以下範例使用 put-metric-alarm 命令，在 CPU 使用率超過 70% 時傳送 Amazon Simple Notification Service 電子郵件訊息：

```
aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-description "Alarm
 when CPU exceeds 70 percent" --metric-name CPUUtilization --namespace AWS/
EC2 --statistic Average --period 300 --threshold 70 --comparison-
operator GreaterThanThreshold  --dimensions "Name=InstanceId,Value=i-12345678" --
evaluation-periods 2 --alarm-actions arn:aws:sns:us-east-1:111122223333:MyTopic --
unit Percent
```

如果成功，此命令會回到提示字元。如果已存在具有相同名稱的警示，將會被新警示所覆寫。

指定多個維度

下列範例說明如何指定多個維度。每個維度都指定為「名稱/值」對，名稱和值之間用逗號分隔。多個維度由空格分隔：

```
aws cloudwatch put-metric-alarm --alarm-name "Default_Test_Alarm3" --alarm-
description "The default example alarm" --namespace "CW EXAMPLE METRICS"
 --metric-name Default_Test --statistic Average --period 60 --evaluation-
periods 3 --threshold 50 --comparison-operator GreaterThanOrEqualToThreshold --
dimensions Name=key1,Value=value1 Name=key2,Value=value2
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutMetricAlarm。

put-metric-data

以下程式碼範例顯示如何使用 put-metric-data。

AWS CLI

將自訂指標發佈至 Amazon CloudWatch

下列範例使用 put-metric-data 命令將自訂指標發佈至 Amazon CloudWatch：

```
aws cloudwatch put-metric-data --namespace "Usage Metrics" --metric-data file://
metric.json
```

指標本身的值會儲存在 JSON 檔案 metric.json 中。

以下是該檔案的內容：

```
[
  {
    "MetricName": "New Posts",
    "Timestamp": "Wednesday, June 12, 2013 8:28:20 PM",
    "Value": 0.50,
    "Unit": "Count"
  }
]
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的「發佈自訂指標」。

指定多個維度

下列範例說明如何指定多個維度。每個維度均指定為 Name=Value 對。使用逗號分隔多個維度：

```
aws cloudwatch put-metric-data --metric-name Buffers --
namespace MyNameSpace --unit Bytes --value 231434333 --
dimensions InstanceID=1-23456789,InstanceType=m1.small
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutMetricData。

put-metric-stream

以下程式碼範例顯示如何使用 put-metric-stream。

AWS CLI

建立指標串流

下列put-metric-stream範例會在指定的帳戶中建立名為 QuickFull-GuaFb的指標串流。

```
aws cloudwatch put-metric-stream \
```

```
    --name QuickFull-GuaFbs \
    --firehose-arn arn:aws:firehose:us-east-1:123456789012:deliverystream/
MetricStreams-QuickFull-GuaFbs-WnySbECG \
    --role-arn arn:aws:iam::123456789012:role/service-role/MetricStreams-
FirehosePutRecords-JN10W9B3 \
    --output-format json \
    --no-include-linked-accounts-metrics
```

輸出：

```
{
    "Arn": "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/QuickFull-
GuaFbs"
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的設定指標串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutMetricStream。

set-alarm-state

以下程式碼範例顯示如何使用 set-alarm-state。

AWS CLI

暫時變更警示的狀態

下列範例使用 set-alarm-state命令暫時變更名為 "myalarm" 的 Amazon CloudWatch 警示狀態，並將其設定為 ALARM 狀態以進行測試：

```
aws cloudwatch set-alarm-state --alarm-name "myalarm" --state-value ALARM --state-
reason "testing purposes"
```

如果成功，此命令會回到提示字元。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetAlarmState。

start-metric-streams

以下程式碼範例顯示如何使用 start-metric-streams。

AWS CLI

啟動指定的指標串流

下列start-metric-streams範例會啟動指定帳戶中名為 QuickFull-GuaFbs的指標串流。

```
aws cloudwatch start-metric-streams \
    --names QuickFull-GuaFbs
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用指標串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartMetricStreams。

stop-metric-streams

以下程式碼範例顯示如何使用 stop-metric-streams。

AWS CLI

停止指定的指標串流

下列stop-metric-streams範例會停止指定帳戶中名為 QuickFull-GuaFbs的指標串流。

```
aws cloudwatch stop-metric-streams \
    --names QuickFull-GuaFbs
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用指標串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopMetricStreams。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將一或多個標籤新增至指定的資源

下列tag-resource範例會將 2 個標籤新增至指定帳戶中名為 demo的 cloudwatch 警示。

```
aws cloudwatch tag-resource \
    --resource-arn arn:aws:cloudwatch:us-east-1:123456789012:alarm:demo \
    --tags Key=stack,Value=Production Key=team,Value=Devops
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的標記 Amazon CloudWatch 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從指定的資源移除一或多個標籤

下列untag-resource範例會從指定帳戶中名為 的 cloudwatch 警示中移除 demo 2 個標籤。

```
aws cloudwatch untag-resource \
    --resource-arn arn:aws:cloudwatch:us-east-1:123456789012:alarm:demo \
    --tag-keys stack team
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的標記 Amazon CloudWatch 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

使用 的 CloudWatch Logs 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 CloudWatch Logs 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-log-group`

以下程式碼範例顯示如何使用 `create-log-group`。

AWS CLI

下列命令會建立名為 的日誌群組`my-logs`：

```
aws logs create-log-group --log-group-name my-logs
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLogGroup。

`create-log-stream`

以下程式碼範例顯示如何使用 `create-log-stream`。

AWS CLI

下列命令會在日誌群組 `20150601`中建立名為 的日誌串流`my-logs`：

```
aws logs create-log-stream --log-group-name my-logs --log-stream-name 20150601
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLogStream。

`delete-log-group`

以下程式碼範例顯示如何使用 `delete-log-group`。

AWS CLI

下列命令會刪除名為 的日誌群組`my-logs`：

```
aws logs delete-log-group --log-group-name my-logs
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLogGroup。

delete-log-stream

以下程式碼範例顯示如何使用 delete-log-stream。

AWS CLI

下列命令20150531會從名為 的日誌群組中刪除名為 的日誌串流my-logs：

```
aws logs delete-log-stream --log-group-name my-logs --log-stream-name 20150531
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLogStream。

delete-retention-policy

以下程式碼範例顯示如何使用 delete-retention-policy。

AWS CLI

下列命令會移除先前已套用至名為 之日誌群組的保留政策my-logs：

```
aws logs delete-retention-policy --log-group-name my-logs
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRetentionPolicy。

describe-log-groups

以下程式碼範例顯示如何使用 describe-log-groups。

AWS CLI

下列命令描述名為 的日誌群組my-logs：

```
aws logs describe-log-groups --log-group-name-prefix my-logs
```

輸出：

```
{
    "logGroups": [
        {
            "storedBytes": 0,
            "metricFilterCount": 0,
            "creationTime": 1433189500783,
            "logGroupName": "my-logs",
            "retentionInDays": 5,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:*"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLogGroups。

describe-log-streams

以下程式碼範例顯示如何使用 describe-log-streams。

AWS CLI

下列命令顯示從日誌群組 2015中字首 開始的所有日誌串流my-logs：

```
aws logs describe-log-streams --log-group-name my-logs --log-stream-name-prefix 2015
```

輸出：

```
{
    "logStreams": [
        {
            "creationTime": 1433189871774,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:log-
stream:20150531",
            "logStreamName": "20150531",
            "storedBytes": 0
        },
        {
            "creationTime": 1433189873898,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:log-
stream:20150601",
            "logStreamName": "20150601",
```

```
            "storedBytes": 0
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLogStreams。

get-log-events

以下程式碼範例顯示如何使用 get-log-events。

AWS CLI

下列命令會從日誌群組 20150601中名為 的日誌串流擷取日誌事件my-logs：

```
aws logs get-log-events --log-group-name my-logs --log-stream-name 20150601
```

輸出：

```
{
    "nextForwardToken":
 "f/31961209122447488583055879464742346735121166569214640130",
    "events": [
        {
            "ingestionTime": 1433190494190,
            "timestamp": 1433190184356,
            "message": "Example Event 1"
        },
        {
            "ingestionTime": 1433190516679,
            "timestamp": 1433190184356,
            "message": "Example Event 1"
        },
        {
            "ingestionTime": 1433190494190,
            "timestamp": 1433190184358,
            "message": "Example Event 2"
        }
    ],
    "nextBackwardToken":
 "b/31961209122358285602261756944988674324553373268216709120"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLogEvents。

put-log-events

以下程式碼範例顯示如何使用 put-log-events。

AWS CLI

下列命令會將日誌事件放入日誌群組 20150601中名為 的日誌串流my-logs：

```
aws logs put-log-events --log-group-name my-logs --log-stream-name 20150601 --log-events file://events
```

輸出：

```
{
    "nextSequenceToken": "495426724868310740095796045676567882148068632824696073 46"
}
```

上述範例會從目前目錄中名為 的檔案讀取事件的 JSON events 陣列：

```
[
  {
    "timestamp": 1433190184356,
    "message": "Example Event 1"
  },
  {
    "timestamp": 1433190184358,
    "message": "Example Event 2"
  },
  {
    "timestamp": 1433190184360,
    "message": "Example Event 3"
  }
]
```

每個後續呼叫都需要使用序列字符選項指定上一個呼叫提供的下一個序列字符：

```
aws logs put-log-events --log-group-name my-logs --log-stream-name 20150601 --log-events file://events2 --sequence-token "495426724868310740095796045676567882148068632824696073 46"
```

輸出：

```
{
    "nextSequenceToken": "495426724868310740095796045679009912303690199563082198 26"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutLogEvents。

put-retention-policy

以下程式碼範例顯示如何使用 put-retention-policy。

AWS CLI

下列命令會將 5 天保留政策新增至名為 的日誌群組my-logs：

```
aws logs put-retention-policy --log-group-name my-logs --retention-in-days 5
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutRetentionPolicy。

使用 的 CloudWatch Network Monitoring 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 CloudWatch Network Monitoring 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

create-monitor

以下程式碼範例顯示如何使用 create-monitor。

AWS CLI

範例 1：使用彙總期間建立網路監視器

下列create-monitor範例會建立名為 的監視器，Example_NetworkMonitor並將aggregationPeriod 設定為 30 秒。監視器state的初始 將是 ，INACTIVE因為沒有與其相關聯的探查。ACTIVE 只有在新增探查時，狀態才會變更為 。您可以使用 update-monitor 或 create-probe 命令，將探查新增至此監視器。

```
aws networkmonitor create-monitor \
     --monitor-name Example_NetworkMonitor \
     --aggregation-period 30
```

輸出：

```
{
    "monitorArn": "arn:aws:networkmonitor:region:111122223333:monitor/
Example_NetworkMonitor",
    "monitorName": "Example_NetworkMonitor",
    "state": "INACTIVE",
    "aggregationPeriod": 30,
    "tags": {}
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch Network Monitor 的運作方式。 Amazon CloudWatch

範例 2：使用 TCP 建立具有探查的網路監視器，並包含標籤

下列create-monitor範例會建立名為 的監視器Example_NetworkMonitor。命令也會建立一個使用ICMP通訊協定並包含標籤的探查。由於請求中aggregationPeriod不會傳遞任何 ，因此60秒會設定為預設值。具有探查的監視器state的 將是 ，PENDING直到監視器是 為止ACTIVE。這可能需要幾分鐘的時間，此時 state會變更為 ACTIVE，而且您可以開始檢視 CloudWatch 指標。

```
aws networkmonitor create-monitor \
    --monitor-name Example_NetworkMonitor \
    --probes sourceArn=arn:aws:ec2:region:111122223333:subnet/subnet-
id,destination=10.0.0.100,destinationPort=80,protocol=TCP,packetSize=56,probeTags={Name=Prob
 \
```

```
    --tags Monitor=Monitor1
```

輸出：

```
{
    "monitorArn": "arn:aws:networkmonitor:region111122223333:monitor/
Example_NetworkMonitor",
    "monitorName": "Example_NetworkMonitor",
    "state": "PENDING",
    "aggregationPeriod": 60,
    "tags": {
        "Monitor": "Monitor1"
    }
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch Network Monitor 的運作方式。 Amazon CloudWatch

範例 3：使用 ICMP 建立具有探查的網路監視器，並包含標籤

下列create-monitor範例會建立名為 aggregationPeriod的監視器Example_NetworkMonitor，其 為 30 秒。命令也會建立一個使用ICMP通訊協定並包含標籤的探查。由於請求中aggregationPeriod不會傳遞任何 ，因此系統會將60秒設定為預設值。具有探查的監視器state的 將是 ，PENDING直到監視器是 為止ACTIVE。這可能需要幾分鐘的時間，此時 state會變更為 ACTIVE，而且您可以開始檢視 CloudWatch 指標。

```
aws networkmonitor create-monitor \
     --monitor-name Example_NetworkMonitor \
     --aggregation-period 30 \
     --probes sourceArn=arn:aws:ec2:region111122223333:subnet/subnet-
id,destination=10.0.0.100,protocol=ICMP,packetSize=56,probeTags={Name=Probe1} \
     --tags Monitor=Monitor1
```

輸出：

```
{
    "monitorArn": "arn:aws:networkmonitor:region:111122223333:monitor/
Example_NetworkMonitor",
    "monitorName": "Example_NetworkMonitor",
    "state": "PENDING",
    "aggregationPeriod": 30,
```

```
    "tags": {
        "Monitor": "Monitor1"
    }
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch Network Monitor 的運作方式。 Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateMonitor。

create-probe

以下程式碼範例顯示如何使用 create-probe。

AWS CLI

範例 1：建立使用 TCP 的探查，並將其新增至網路監視器

下列create-probe範例會建立使用 的探查，TCPprotocol並將探查新增至名為 的監視器Example_NetworkMonitor。建立之後，具有探查的監視器state的 將會是 ，PENDING直到監視器是 為止ACTIVE。這可能需要幾分鐘的時間，此時狀態會變更為 ACTIVE，而且您可以開始檢視 CloudWatch 指標。

```
aws networkmonitor create-probe \
    --monitor-name Example_NetworkMonitor \
    --probe sourceArn=arn:aws:ec2:region:111122223333:subnet/subnet-
id,destination=10.0.0.100,destinationPort=80,protocol=TCP,packetSize=56,tags={Name=Probe1}
```

輸出：

```
{
    "probeId": "probe-12345",
    "probeArn": "arn:aws:networkmonitor:region:111122223333:probe/probe-12345",
    "destination": "10.0.0.100",
    "destinationPort": 80,
    "packetSize": 56,
    "addressFamily": "IPV4",
    "vpcId": "vpc-12345",
    "state": "PENDING",
    "createdAt": "2024-03-29T12:41:57.314000-04:00",
    "modifiedAt": "2024-03-29T12:41:57.314000-04:00",
    "tags": {
```

```
        "Name": "Probe1"
    }
}
```

範例 2：使用 ICMP 建立使用探查的探查，並將其新增至網路監視器

下列create-probe範例會建立使用 的探查，ICMPprotocol並將探查新增至名為 的監視器Example_NetworkMonitor。建立之後，具有探查的監視器state的 將會是 ，PENDING直到監視器是 為止ACTIVE。這可能需要幾分鐘的時間，此時狀態會變更為 ACTIVE，而且您可以開始檢視 CloudWatch 指標。

```
aws networkmonitor create-probe \
    --monitor-name Example_NetworkMonitor \
    --probe sourceArn=arn:aws:ec2:region:012345678910:subnet/subnet-
id,destination=10.0.0.100,protocol=ICMP,packetSize=56,tags={Name=Probe1}
```

輸出：

```
{
    "probeId": "probe-12345",
    "probeArn": "arn:aws:networkmonitor:region:111122223333:probe/probe-12345",
    "destination": "10.0.0.100",
    "packetSize": 56,
    "addressFamily": "IPV4",
    "vpcId": "vpc-12345",
    "state": "PENDING",
    "createdAt": "2024-03-29T12:44:02.452000-04:00",
    "modifiedAt": "2024-03-29T12:44:02.452000-04:00",
    "tags": {
        "Name": "Probe1"
    }
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch Network Monitor 的運作方式。 Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateProbe。

delete-monitor

以下程式碼範例顯示如何使用 delete-monitor。

AWS CLI

刪除監視器

下列delete-monitor範例會刪除名為 的監視器Example_NetworkMonitor。

```
aws networkmonitor delete-monitor \
    --monitor-name Example_NetworkMonitor
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch Network Monitor 的運作方式。 Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteMonitor。

delete-probe

以下程式碼範例顯示如何使用 delete-probe。

AWS CLI

刪除探查

下列delete-probe範例probe-12345會從名為 的網路監視器中刪除 ID 為 的探查Example_NetworkMonitor。

```
aws networkmonitor delete-probe \
    --monitor-name Example_NetworkMonitor \
    --probe-id probe-12345
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch Network Monitor 的運作方式。 Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteProbe。

get-monitor

以下程式碼範例顯示如何使用 get-monitor。

AWS CLI

取得監視器資訊

下列get-monitor範例取得名為 的監視器相關資訊Example_NetworkMonitor。

```
aws networkmonitor get-monitor \
    --monitor-name Example_NetworkMonitor
```

輸出：

```
{
    "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor",
    "monitorName": "Example_NetworkMonitor",
    "state": "ACTIVE",
    "aggregationPeriod": 60,
    "tags": {},
    "probes": [],
    "createdAt": "2024-04-01T17:58:07.211000-04:00",
    "modifiedAt": "2024-04-01T17:58:07.211000-04:00"
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch Network Monitor 的運作方式。 Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMonitor。

get-probe

以下程式碼範例顯示如何使用 get-probe。

AWS CLI

檢視探查詳細資訊

下列get-probe範例會傳回與名為 之監視器probeIDprobe-12345相關聯的 探查詳細資訊Example_NetworkMonitor。

```
aws networkmonitor get-probe \
    --monitor-name Example_NetworkMonitor \
```

```
    --probe-id probe-12345
```

輸出：

```
{
    "probeId": "probe-12345",
    "probeArn": "arn:aws:networkmonitor:region:012345678910:probe/probe-12345",
    "sourceArn": "arn:aws:ec2:region:012345678910:subnet/subnet-12345",
    "destination": "10.0.0.100",
    "destinationPort": 80,
    "protocol": "TCP",
    "packetSize": 56,
    "addressFamily": "IPV4",
    "vpcId": "vpc-12345",
    "state": "ACTIVE",
    "createdAt": "2024-03-29T12:41:57.314000-04:00",
    "modifiedAt": "2024-03-29T12:42:28.610000-04:00",
    "tags": {
        "Name": "Probe1"
    }
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch Network Monitor 的運作方式。 Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetProbe。

list-monitors

以下程式碼範例顯示如何使用 list-monitors。

AWS CLI

範例 1：列出所有監視器 （ 單一監視器 ）

下列list-monitors範例只會傳回單一監視器的清單。監視器的 state 是 ，ACTIVE且其具有 60 秒aggregationPeriod的 。

```
aws networkmonitor list-monitors
```

輸出：

```
{
    "monitors": [{
            "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor",
            "monitorName": "Example_NetworkMonitor",
            "state": "ACTIVE",
            "aggregationPeriod": 60,
            "tags": {
                "Monitor": "Monitor1"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch Network Monitor 的運作方式。 Amazon CloudWatch

範例 2：列出所有監視器 （ 多個監視器 ）

下列list-monitors範例會傳回三個監視器的清單。一個監視器state的 是 ACTIVE和產生 CloudWatch 指標。其他兩個監視器的狀態為 INACTIVE，而不是產生 CloudWatch 指標。所有三個監視器都使用 60 秒aggregationPeriod的 。

```
aws networkmonitor list-monitors
```

輸出：

```
{
    "monitors": [
        {
            "monitorArn": "arn:aws:networkmonitor:us-east-1:111122223333:monitor/
Example_NetworkMonitor",
            "monitorName": "Example_NetworkMonitor",
            "state": "INACTIVE",
            "aggregationPeriod": 60,
            "tags": {}
        },
        {
            "monitorArn": "arn:aws:networkmonitor:us-east-1:111122223333:monitor/
Example_NetworkMonitor2",
            "monitorName": "Example_NetworkMonitor2",
            "state": "ACTIVE",
```

```
            "aggregationPeriod": 60,
            "tags": {
                "Monitor": "Monitor1"
            }
        },
        {
            "monitorArn": "arn:aws:networkmonitor:us-east-1:111122223333:monitor/
TestNetworkMonitor_CLI",
            "monitorName": "TestNetworkMonitor_CLI",
            "state": "INACTIVE",
            "aggregationPeriod": 60,
            "tags": {}
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch Network Monitor 的運作方式。 Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMonitors。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出資源的標籤

下列`list-tags-for-resource`範例會傳回名為 之監視器的標籤清單`Example_NetworkMonitor`。

```
aws networkmonitor list-tags-for-resource \
    --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor
```

輸出：

```
{
    "tags": {
        "Environment": "Dev",
        "Application": "PetStore"
```

```
    }
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch Network Monitor 的運作方式。 Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`tag-resource`

以下程式碼範例顯示如何使用 `tag-resource`。

AWS CLI

標記資源

下列`tag-resource`範例`Example_NetworkMonitor`使用 `Environment=Dev`和 標籤`Application=PetStore`標記名為 的監視器。

```
aws networkmonitor tag-resource \
    --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor \
    --tags Environment=Dev,Application=PetStore
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch Network Monitor 的運作方式。 Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

`untag-resource`

以下程式碼範例顯示如何使用 `untag-resource`。

AWS CLI

取消標記資源

下列`untag-resource`範例會從`Environment Application`其與名為 之監視器的關聯中移除具有 鍵/值對的`tag-keys`參數`Example_NetworkMonitor`。

```
aws networkmonitor untag-resource \
    --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor \
    --tag-keys Environment Application
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch Network Monitor 的運作方式。 Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-monitor

以下程式碼範例顯示如何使用 update-monitor。

AWS CLI

更新監視器

下列update-monitor範例會將監視器的 aggregationPeriod 從 60 秒變更為 30 秒。

```
aws networkmonitor update-monitor \
    --monitor-name Example_NetworkMonitor \
    --aggregation-period 30
```

輸出：

```
{
    "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor",
    "monitorName": "Example_NetworkMonitor",
    "state": "PENDING",
    "aggregationPeriod": 30,
    "tags": {
        "Monitor": "Monitor1"
    }
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch Network Monitor 的運作方式。 Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateMonitor。

update-probe

以下程式碼範例顯示如何使用 update-probe。

AWS CLI

更新探查

下列update-probe範例會更新探查的原始 destination IP 地址，並將 更新packetSize為 60。

```
aws networkmonitor update-probe \
    --monitor-name Example_NetworkMonitor \
    --probe-id probe-12345 \
    --destination 10.0.0.150 \
    --packet-size 60
```

輸出：

```
{
    "probeId": "probe-12345",
    "probeArn": "arn:aws:networkmonitor:region:012345678910:probe/probe-12345",
    "sourceArn": "arn:aws:ec2:region:012345678910:subnet/subnet-12345",
    "destination": "10.0.0.150",
    "destinationPort": 80,
    "protocol": "TCP",
    "packetSize": 60,
    "addressFamily": "IPV4",
    "vpcId": "vpc-12345",
    "state": "PENDING",
    "createdAt": "2024-03-29T12:41:57.314000-04:00",
    "modifiedAt": "2024-03-29T13:52:23.115000-04:00",
    "tags": {
        "Name": "Probe1"
    }
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 Amazon CloudWatch Network Monitor 的運作方式。 Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateProbe。

使用 的 CloudWatch 可觀測性存取監視器範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 CloudWatch Observability Access Monitor 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-link`

以下程式碼範例顯示如何使用 `create-link`。

AWS CLI

建立連結

下列`create-link`範例會在來源帳戶與您在監控帳戶中建立的接收器之間建立連結。

```
aws oam create-link \
    --label-template sourceAccount \
    --resource-types AWS::CloudWatch::Metric \
    --sink-identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345
```

輸出：

```
{
    "Arn": "arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111",
    "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
```

```
    "Label": "sourceAccount",
    "LabelTemplate": "sourceAccount",
    "ResourceTypes": [
        "AWS::CloudWatch::Metric"
    ],
    "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-
example12345",
    "Tags": {}
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 CloudWatch 跨帳戶可觀測性。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLink。

create-sink

以下程式碼範例顯示如何使用 create-sink。

AWS CLI

建立接收器

下列create-sink範例會在目前帳戶中建立接收器，以便在 CloudWatch 跨帳戶可觀測性中用作監控帳戶。

```
aws oam create-sink \
    --name DemoSink
```

輸出：

```
{
    "Arn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-
example12345",
    "Id": "a1b2c3d4-5678-90ab-cdef-example12345",
    "Name": "DemoSink",
    "Tags": {}
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 CloudWatch 跨帳戶可觀測性。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSink。

delete-link

以下程式碼範例顯示如何使用 delete-link。

AWS CLI

刪除連結

下列delete-link範例會刪除監控帳戶目的地和來源帳戶之間的連結。

```
aws oam delete-link \
    --identifier arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 CloudWatch 跨帳戶可觀測性。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLink。

delete-sink

以下程式碼範例顯示如何使用 delete-sink。

AWS CLI

刪除接收器

下列delete-sink範例會刪除接收器。您必須刪除目的地的所有連結，才能刪除該目的地。

```
aws oam delete-sink \
    --identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 CloudWatch 跨帳戶可觀測性。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSink。

get-link

以下程式碼範例顯示如何使用 get-link。

AWS CLI

傳回一個連結的完整資訊

下列get-link範例會傳回連結的完整資訊。

```
aws oam get-link \
    --identifier arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111
```

輸出：

```
{
    "Arn": "arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111",
    "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
    "Label": "sourceAccount",
    "LabelTemplate": "sourceAccount",
    "ResourceTypes": [
        "AWS::CloudWatch::Metric"
    ],
    "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345",
    "Tags": {}
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 CloudWatch 跨帳戶可觀測性。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLink。

get-sink-policy

以下程式碼範例顯示如何使用 get-sink-policy。

AWS CLI

傳回連接到接收器的目前接收器政策

下列get-sink-policy範例會傳回連接到接收器的目前接收器政策。

```
aws oam get-sink-policy \
    --sink-identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345
```

輸出：

```
{
    "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345",
    "SinkId": "a1b2c3d4-5678-90ab-cdef-example12345",
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789111:root\"},\"Action\":[\"oam:CreateLink\",\"oam:UpdateLink\"],\"Resource\":\"*\",\"Condition\":{\"ForAllValues:StringEquals\":{\"oam:ResourceTypes\":[\"AWS::Logs::LogGroup\",\"AWS::CloudWatch::Metric\",\"AWS::XRay::Trace\",\"AWS::ApplicationInsights::Application\"]}}}]}"
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 CloudWatch 跨帳戶可觀測性。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSinkPolicy。

get-sink

以下程式碼範例顯示如何使用 get-sink。

AWS CLI

傳回一個監控帳戶接收器的完整資訊

下列get-sink範例會傳回監控帳戶目的地的完整資訊。

```
aws oam get-sink \
    --identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345
```

輸出：

```
{
    "Arn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-
example12345",
    "Id": "a1b2c3d4-5678-90ab-cdef-example12345",
    "Name": "DemoSink",
    "Tags": {}
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 CloudWatch 跨帳戶可觀測性。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSink。

`list-attached-links`

以下程式碼範例顯示如何使用 `list-attached-links`。

AWS CLI

傳回連結至此監控帳戶目的地的來源帳戶連結清單

下列`list-attached-links`範例會傳回連結至此監控帳戶目的地的來源帳戶連結清單。

```
aws oam list-attached-links \
    --sink-identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-
cdef-example12345
```

輸出：

```
{
    "Items": [{
        "Label": "Monitoring account",
        "LinkArn": "arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-
example11111",
        "ResourceTypes": [
            "AWS::ApplicationInsights::Application",
            "AWS::Logs::LogGroup",
            "AWS::CloudWatch::Metric",
            "AWS::XRay::Trace"
        ]
```

```
    }]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 CloudWatch 跨帳戶可觀測性。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAttachedLinks。

`list-links`

以下程式碼範例顯示如何使用 `list-links`。

AWS CLI

傳回一個監控帳戶目的地的連結清單

下列`list-links`範例會傳回一個監控帳戶目的地的連結清單。在來源帳戶中執行此操作，以傳回此來源帳戶擁有之監控帳戶目的地的連結清單。

```
aws oam list-links
```

輸出：

```
{
    "Items": [{
        "Arn": "arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-
example11111",
        "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
        "Label": "sourceAccount",
        "ResourceTypes": [
            "AWS::CloudWatch::Metric"
        ],
        "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-
example12345"
    }]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 CloudWatch 跨帳戶可觀測性。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListLinks。

list-sinks

以下程式碼範例顯示如何使用 list-sinks。

AWS CLI

傳回在監控帳戶中建立的目的地清單

下列list-sinks範例會傳回監控帳戶中建立的接收器清單。在監控帳戶中執行此操作。

```
aws oam list-sinks
```

輸出：

```
{
    "Items": [
        {
            "Arn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-
example12345",
            "Id": "a1b2c3d4-5678-90ab-cdef-example12345",
            "Name": "DemoSink"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 CloudWatch 跨帳戶可觀測性。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSinks。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

顯示與資源相關聯的標籤

下列list-tags-for-resource範例顯示與接收器相關聯的標籤。

```
aws oam list-tags-for-resource \
```

```
    --resource-arn arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345
```

輸出：

```
{
    "Tags": {
        "Team": "Devops"
    }
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 CloudWatch 跨帳戶可觀測性。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

put-sink-policy

以下程式碼範例顯示如何使用 put-sink-policy。

AWS CLI

建立或更新資源政策

下列put-sink-policy範例會建立 資源政策，將許可授予來源帳戶，以連結至監控帳戶目的地。

```
aws oam put-sink-policy \
    --policy '{"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":{"AWS":"arn:aws:iam::123456789111:root"},"Action":["oam:CreateLink","oam:UpdateLink"],"Resource":"*","Condition":{"ForAllValues:StringEquals":{"oam:ResourceTypes":["AWS::Logs::LogGroup","AWS::CloudWatch::Metric","AWS::XRay::Trace","AWS::ApplicationInsight
 \
    --sink-identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345
```

輸出：

```
{
    "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345",
    "SinkId": "a1b2c3d4-5678-90ab-cdef-example12345",
```

```
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789111:root\"},
\"Action\":[\"oam:CreateLink\",\"oam:UpdateLink\"],\"Resource\":\"*\",
\"Condition\":{\"ForAllValues:StringEquals\":{\"oam:ResourceTypes\":
[\"AWS::Logs::LogGroup\",\"AWS::CloudWatch::Metric\",\"AWS::XRay::Trace\",
\"AWS::ApplicationInsights::Application\"]}}}]}"
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 CloudWatch 跨帳戶可觀測性。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutSinkPolicy。

`tag-resource`

以下程式碼範例顯示如何使用 `tag-resource`。

AWS CLI

將一或多個標籤指派給指定的資源

下列`tag-resource`範例會標記目的地 `arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345`。

```
aws oam tag-resource \
    --resource-arn arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345 \
    --tags team=Devops
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 CloudWatch 跨帳戶可觀測性。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

`untag-resource`

以下程式碼範例顯示如何使用 `untag-resource`。

AWS CLI

從指定的資源移除一或多個標籤。

下列untag-resource範例team會從目的地 移除具有 金鑰的標籤arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345。

```
aws oam untag-resource \
    --resource-arn arn:aws:oam:us-east-2:123456789012:sink/f3f42f60-f0f2-425c-1234-12347bdd821f \
    --tag-keys team
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 CloudWatch 跨帳戶可觀測性。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-link

以下程式碼範例顯示如何使用 update-link。

AWS CLI

將來源帳戶共用的資料類型變更為其連結的監控帳戶目的地

下列update-link範例會使用資源類型 arn:aws:oam:us-east-2:123456789111:link/0123e691-e7ef-43fa-1234-c57c837fced0 AWS::CloudWatch::Metric和 更新連結AWS::Logs::LogGroup。

```
aws oam update-link \
    --identifier arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111 \
    --resource-types "AWS::CloudWatch::Metric" "AWS::Logs::LogGroup"
```

輸出：

```
{
    "Arn": "arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111",
    "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
    "Label": "sourceAccount",
    "LabelTemplate": "sourceAccount",
    "ResourceTypes": [
```

```
        "AWS::CloudWatch::Metric",
        "AWS::Logs::LogGroup"
    ],
    "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-
example12345",
    "Tags": {}
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 CloudWatch 跨帳戶可觀測性。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateLink。

使用 的 CloudWatch 可觀測性管理員範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 CloudWatch Observability Admin 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`get-telemetry-evaluation-status-for-organization`

以下程式碼範例顯示如何使用 `get-telemetry-evaluation-status-for-organization`。

AWS CLI

取得組織的遙測加入狀態

下列`get-telemetry-evaluation-status-for-organization`範例會傳回組織遙測組態功能的目前加入狀態。

```
aws observabilityadmin get-telemetry-evaluation-status-for-organization
```

輸出：

```
{
    "Status": "RUNNING"
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的稽核 CloudWatch 遙測組態。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTelemetryEvaluationStatusForOrganization。

get-telemetry-evaluation-status

以下程式碼範例顯示如何使用 get-telemetry-evaluation-status。

AWS CLI

取得帳戶的遙測加入狀態

下列get-telemetry-evaluation-status範例會傳回指定帳戶中遙測組態功能的目前加入狀態。

```
aws observabilityadmin get-telemetry-evaluation-status
```

輸出：

```
{
    "Status": "RUNNING"
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的稽核 CloudWatch 遙測組態。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTelemetryEvaluationStatus。

list-resource-telemetry-for-organization

以下程式碼範例顯示如何使用 list-resource-telemetry-for-organization。

AWS CLI

擷取組織的遙測組態

下列list-resource-telemetry-for-organization範例會針對遙測組態支援 AWS 的資源，傳回組織中的遙測組態清單。

```
aws observabilityadmin list-resource-telemetry-for-organization \
    --resource-types  AWS::EC2::Instance
```

輸出：

```
{
    "TelemetryConfigurations": [
        {
            "AccountIdentifier": "111111111111",
            "TelemetryConfigurationState": {
                "Logs": "NotApplicable",
                "Metrics": "Disabled",
                "Traces": "NotApplicable"
            },
            "ResourceType": "AWS::EC2::Instance",
            "ResourceIdentifier": "i-a166400b",
            "ResourceTags": {
                "Name": "dev"
            },
            "LastUpdateTimeStamp": 1733168548521
        },
        {
            "AccountIdentifier": "222222222222",
            "TelemetryConfigurationState": {
                "Logs": "NotApplicable",
                "Metrics": "Disabled",
                "Traces": "NotApplicable"
            },
            "ResourceType": "AWS::EC2::Instance",
            "ResourceIdentifier": "i-b188560f",
            "ResourceTags": {
                "Name": "apache"
            },
            "LastUpdateTimeStamp": 1732744260182
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的稽核 CloudWatch 遙測組態。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourceTelemetryForOrganization。

`list-resource-telemetry`

以下程式碼範例顯示如何使用 `list-resource-telemetry`。

AWS CLI

擷取帳戶的遙測組態

下列`list-resource-telemetry`範例會傳回指定帳戶中遙測組態所支援 AWS 之資源的遙測組態清單。

```
aws observabilityadmin list-resource-telemetry \
    --resource-types  AWS::EC2::Instance
```

輸出：

```
{
    "TelemetryConfigurations": [
        {
            "AccountIdentifier": "111111111111",
            "TelemetryConfigurationState": {
                "Logs": "NotApplicable",
                "Metrics": "Disabled",
                "Traces": "NotApplicable"
            },
            "ResourceType": "AWS::EC2::Instance",
            "ResourceIdentifier": "i-0e979d278b040f856",
            "ResourceTags": {
                "Name": "apache"
            },
            "LastUpdateTimeStamp": 1732744260182
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的稽核 CloudWatch 遙測組態。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourceTelemetry。

start-telemetry-evaluation-for-organization

以下程式碼範例顯示如何使用 start-telemetry-evaluation-for-organization。

AWS CLI

啟用遙測組態功能

下列start-telemetry-evaluation-for-organization範例會啟用組織的遙測組態功能。

```
aws observabilityadmin start-telemetry-evaluation-for-organization
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的開啟 CloudWatch 遙測稽核。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartTelemetryEvaluationForOrganization。

start-telemetry-evaluation

以下程式碼範例顯示如何使用 start-telemetry-evaluation。

AWS CLI

啟用遙測組態功能

下列start-telemetry-evaluation範例會在指定的帳戶中啟用遙測組態功能。

```
aws observabilityadmin start-telemetry-evaluation
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的開啟 CloudWatch 遙測稽核。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartTelemetryEvaluation。

stop-telemetry-evaluation-for-organization

以下程式碼範例顯示如何使用 stop-telemetry-evaluation-for-organization。

AWS CLI

停用遙測組態功能

下列stop-telemetry-evaluation-for-organization範例會停用組織的遙測組態功能。

```
aws observabilityadmin stop-telemetry-evaluation-for-organization
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的關閉 CloudWatch 遙測稽核。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopTelemetryEvaluationForOrganization。

stop-telemetry-evaluation

以下程式碼範例顯示如何使用 stop-telemetry-evaluation。

AWS CLI

停用遙測組態功能

下列stop-telemetry-evaluation範例會停用指定帳戶中的遙測組態功能。

```
aws observabilityadmin stop-telemetry-evaluation
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的關閉 CloudWatch 遙測稽核。Amazon CloudWatch

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopTelemetryEvaluation。

使用 的 CloudWatch Synthetics 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 CloudWatch Synthetics 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-resource`

以下程式碼範例顯示如何使用 `associate-resource`。

AWS CLI

將 Canary 與群組建立關聯

下列`associate-resource`範例會將 Canary 與名為 的群組建立關聯`demo_group`。

```
aws synthetics associate-resource \
    --group-identifier demo_group \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateResource。

`create-canary`

以下程式碼範例顯示如何使用 `create-canary`。

AWS CLI

建立 Canary

下列create-canary範例會建立名為 的 Canarydemo_canary。

```
aws synthetics create-canary \
    --name demo_canary \
    --code '{"S3Bucket": "artifacts3bucket", "S3Key":"demo_canary.zip", "Handler":
 "index.lambda_handler"}' \
    --artifact-s3-location s3://amzn-s3-demo-bucket/demo_canary.zip \
    --execution-role-arn arn:aws:iam::123456789012:role/demo_canary_role \
    --schedule Expression="rate(10 minutes)" \
    --runtime-version syn-nodejs-puppeteer-9.1
```

輸出：

```
{
    "Canary": {
        "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
        "Name": "demo_canary",
        "Code": {
            "Handler": "index.lambda_handler"
        },
        "ExecutionRoleArn": "arn:aws:iam::123456789012:role/demo_canary_role",
        "Schedule": {
            "Expression": "rate(10 minutes)",
            "DurationInSeconds": 0
        },
        "RunConfig": {
            "TimeoutInSeconds": 600,
            "MemoryInMB": 1000,
            "ActiveTracing": false
        },
        "SuccessRetentionPeriodInDays": 31,
        "FailureRetentionPeriodInDays": 31,
        "Status": {
            "State": "CREATING",
            "StateReasonCode": "CREATE_PENDING"
        },
        "Timeline": {
            "Created": "2024-10-15T19:03:08.826000+05:30",
            "LastModified": "2024-10-15T19:03:08.826000+05:30"
        },
        "ArtifactS3Location": "amzn-s3-demo-bucket/demo_canary.zip",
        "RuntimeVersion": "syn-nodejs-puppeteer-9.1",
        "Tags": {}
```

```
    }
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCanary。

create-group

以下程式碼範例顯示如何使用 create-group。

AWS CLI

建立群組

下列create-group範例會建立名為 的群組demo_group。

```
aws synthetics create-group \
    --name demo_group
```

輸出：

```
{
    "Group": {
        "Id": "example123",
        "Name": "demo_group",
        "Arn": "arn:aws:synthetics:us-east-1:123456789012:group:example123",
        "Tags": {},
        "CreatedTime": "2024-10-15T14:47:23.811000+05:30",
        "LastModifiedTime": "2024-10-15T14:47:23.811000+05:30"
    }
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateGroup。

delete-canary

以下程式碼範例顯示如何使用 delete-canary。

AWS CLI

永久刪除 Canary

下列delete-canary範例會刪除名為 的 Canarydemo_canary。

```
aws synthetics delete-canary \
    --name demo_canary
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCanary。

delete-group

以下程式碼範例顯示如何使用 delete-group。

AWS CLI

刪除群組

下列delete-group範例會刪除名為 的群組demo_group。

```
aws synthetics delete-group \
    --group-identifier demo_group
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteGroup。

describe-canaries-last-run

以下程式碼範例顯示如何使用 describe-canaries-last-run。

AWS CLI

查看每個 Canary 最近執行的資訊

下列describe-canaries-last-run範例會傳回您建立的每個 Canary 的最新執行。

```
aws synthetics describe-canaries-last-run
```

輸出：

```
{
    "CanariesLastRun": [
        {
            "CanaryName": "demo_canary",
            "LastRun": {
                "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
                "Name": "demo_canary",
                "Status": {
                    "State": "PASSED",
                    "StateReason": "",
                    "StateReasonCode": ""
                },
                "Timeline": {
                    "Started": "2024-10-15T19:20:39.691000+05:30",
                    "Completed": "2024-10-15T19:20:58.211000+05:30"
                },
                "ArtifactS3Location": "cw-syn-results-123456789012-us-east-1/canary/
us-east-1/demo_canary-abc-example1234/2024/10/15/13/50-39-690"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCanariesLastRun。

describe-canaries

以下程式碼範例顯示如何使用 describe-canaries。

AWS CLI

列出您帳戶中的 Canary

下列describe-canaries範例列出您帳戶中 Canary 的詳細資訊。

```
aws synthetics describe-canaries
```

輸出：

```
{
    "Canaries": [
        {
            "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
            "Name": "demo_canary",
            "Code": {
                "SourceLocationArn": "arn:aws:lambda:us-
east-1:123456789012:layer:cwsyn-demo_canary-a1b2c3d4-5678-90ab-cdef-
example11111b8:1",
                "Handler": "pageLoadBlueprint.handler"
            },
            "ExecutionRoleArn": "arn:aws:iam::123456789012:role/service-role/
CloudWatchSyntheticsRole-demo_canary-a12-a123bc456789",
            "Schedule": {
                "Expression": "rate(5 minutes)",
                "DurationInSeconds": 0
            },
            "RunConfig": {
                "TimeoutInSeconds": 300,
                "MemoryInMB": 1000,
                "ActiveTracing": false
            },
            "SuccessRetentionPeriodInDays": 31,
            "FailureRetentionPeriodInDays": 31,
            "Status": {
            "State": "RUNNING"
            },
            "Timeline": {
                "Created": "2024-10-15T18:55:15.168000+05:30",
                "LastModified": "2024-10-15T18:55:40.540000+05:30",
                "LastStarted": "2024-10-15T18:55:40.540000+05:30"
            },
            "ArtifactS3Location": "cw-syn-results-123456789012-us-east-1/canary/us-
east-1/demo_canary-a12-a123bc456789",
            "EngineArn": "arn:aws:lambda:us-east-1:123456789012:function:cwsyn-
demo_canary-a1b2c3d4-5678-90ab-cdef-example111118:1",
            "RuntimeVersion": "syn-nodejs-puppeteer-9.1",
            "Tags": {
                "blueprint": "heartbeat"
            }
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCanaries。

describe-runtime-versions

以下程式碼範例顯示如何使用 describe-runtime-versions。

AWS CLI

傳回合成 Canary 執行時間版本的清單

下列describe-runtime-versions範例會傳回合成 Canary 執行時間版本的清單。

```
aws synthetics describe-runtime-versions
```

輸出：

```
{
    "RuntimeVersions": [
        {
            "VersionName": "syn-nodejs-puppeteer-9.1",
            "Description": "Security fixes and bug fix for date range error in har.
 Dependencies: Node JS 20.x, Puppeteer-core 22.12.1, Chromium 126.0.6478.126",
            "ReleaseDate": "2024-10-02T05:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-9.0",
            "Description": "Upgraded Chromium and Puppeteer. Dependencies: Node JS
 20.x, Puppeteer-core 22.12.1, Chromium 126.0.6478.126",
            "ReleaseDate": "2024-07-22T05:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-8.0",
            "Description": "Upgraded Chromium and Puppeteer. Dependencies: Node JS
 20.x, Puppeteer-core 22.10.0, Chromium 125.0.6422.112",
            "ReleaseDate": "2024-06-21T05:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-7.0",
```

```
            "Description": "Upgraded Chromium and Puppeteer. Dependencies: Node JS
18.x, Puppeteer-core 21.9.0, Chromium 121.0.6167.139",
            "ReleaseDate": "2024-03-08T05:30:00+05:30"
            },
        {
            "VersionName": "syn-nodejs-puppeteer-6.2",
            "Description": "Updated shared libraries for Chromium and added
ephemeral storage monitoring. Dependencies: Node JS 18.x, Puppeteer-core 19.7.0,
Chromium 111.0.5563.146",
            "ReleaseDate": "2024-02-02T05:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-6.1",
            "Description": "Added puppeteer launch retry. Dependencies: Node JS
18.x, Puppeteer-core 19.7.0, Chromium 111.0.5563.146",
            "ReleaseDate": "2023-11-13T05:30:00+05:30",
            "DeprecationDate": "2024-03-08T13:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-6.0",
            "Description": "Reduced X-Ray traces of a canary run, improved duration
metric and upgraded to NodeJS 18.x. Dependencies: Node JS 18.x, Puppeteer-core
19.7.0, Chromium 111.0.5563.146",
            "ReleaseDate": "2023-09-15T05:30:00+05:30",
            "DeprecationDate": "2024-03-08T13:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-5.2",
            "Description": "Updated shared libraries for Chromium. Dependencies:
Node JS 16.x, Puppeteer-core 19.7.0, Chromium 111.0.5563.146",
            "ReleaseDate": "2024-02-01T05:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-5.1",
            "Description": "Fixes a bug about missing request headers in har.
Dependencies: Node JS 16.x, Puppeteer-core 19.7.0, Chromium 111.0.5563.146",
            "ReleaseDate": "2023-08-09T05:30:00+05:30",
            "DeprecationDate": "2024-03-08T13:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-5.0",
            "Description": "Upgraded Puppeteer and Chromium. Dependencies: Node JS
16.x, Puppeteer-core 19.7.0, Chromium 111.0.5563.146",
            "ReleaseDate": "2023-07-21T05:30:00+05:30",
```

```
            "DeprecationDate": "2024-03-08T13:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-4.0",
            "Description": "Upgraded to NodeJS 16.x. Dependencies: Node JS 16.x,
 Puppeteer-core 5.5.0, Chromium 92.0.4512.0",
            "ReleaseDate": "2023-05-01T05:30:00+05:30",
            "DeprecationDate": "2024-03-08T13:30:00+05:30"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRuntimeVersions。

disassociate-resource

以下程式碼範例顯示如何使用 disassociate-resource。

AWS CLI

從群組中移除 Canary

下列disassociate-resource範例會從名為 的群組中移除 Canarydemo_group。

```
aws synthetics disassociate-resource \
    --group-identifier demo_group \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateResource。

get-canary-runs

以下程式碼範例顯示如何使用 get-canary-runs。

AWS CLI

擷取指定 Canary 的執行清單

下列get-canary-runs範例會擷取名為 之 Canary 的執行清單demo_canary。

```
aws synthetics get-canary-runs \
    --name demo_canary
```

輸出：

```
{
    "CanaryRuns": [
        {
            "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
            "Name": "demo_canary",
            "Status": {
                "State": "PASSED",
                "StateReason": "",
                "StateReasonCode": ""
            },
            "Timeline": {
                "Started": "2024-10-16T10:38:57.013000+05:30",
                "Completed": "2024-10-16T10:39:25.793000+05:30"
            },
            "ArtifactS3Location": "cw-syn-results-123456789012-us-east-1/canary/us-
east-1/demo_canary-abc-example1234/2024/10/15/13/50-39-690"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCanaryRuns。

get-canary

以下程式碼範例顯示如何使用 get-canary。

AWS CLI

擷取有關一個 Canary 的完整資訊

下列get-canary範例會擷取名為 之 Canary 的完整資訊demo_canary。

```
aws synthetics get-canary \
```

```
    --name demo_canary
```

輸出：

```
{
    "Canary": {
        "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
        "Name": "demo_canary",
        "Code": {
            "SourceLocationArn": "arn:aws:lambda:us-east-1:123456789012:layer:cwsyn-
demo_canary-a1b2c3d4-5678-90ab-cdef-example111118:1",
            "Handler": "pageLoadBlueprint.handler"
        },
        "ExecutionRoleArn": "arn:aws:iam::123456789012:role/demo_canary_role",
        "Schedule": {
            "Expression": "rate(10 minutes)",
            "DurationInSeconds": 0
        },
        "RunConfig": {
            "TimeoutInSeconds": 300,
            "MemoryInMB": 1000,
            "ActiveTracing": false
        },
        "SuccessRetentionPeriodInDays": 31,
        "FailureRetentionPeriodInDays": 31,
        "Status": {
            "State": "RUNNING"
        },
        "Timeline": {
            "Created": "2024-10-15T18:55:15.168000+05:30",
            "LastModified": "2024-10-15T18:55:40.540000+05:30",
            "LastStarted": "2024-10-15T18:55:40.540000+05:30"
        },
        "ArtifactS3Location": "cw-syn-results-123456789012-us-east-1/canary/us-
east-1/demo_canary-a12-a123bc456789",
        "EngineArn": "arn:aws:lambda:us-east-1:123456789012:function:cwsyn-
demo_canary-a1b2c3d4-5678-90ab-cdef-example111118:1",
        "RuntimeVersion": "syn-nodejs-puppeteer-9.1",
        "Tags": {
            "blueprint": "heartbeat"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCanary。

get-group

以下程式碼範例顯示如何使用 get-group。

AWS CLI

傳回一個群組的相關資訊

下列get-group範例會傳回名為 之群組的相關資訊demo_group。

```
aws synthetics get-group \
    --group-identifier demo_group
```

輸出：

```
{
    "Group": {
        "Id": "example123",
        "Name": "demo_group",
        "Arn": "arn:aws:synthetics:us-east-1:123456789012:group:example123",
        "Tags": {},
        "CreatedTime": "2024-10-15T14:47:23.811000+05:30",
        "LastModifiedTime": "2024-10-15T14:47:23.811000+05:30"
    }
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetGroup。

list-associated-groups

以下程式碼範例顯示如何使用 list-associated-groups。

AWS CLI

傳回群組清單

下列list-associated-groups範例會傳回與名為 的 Canary 相關聯的群組清單demo_canary。

```
aws synthetics list-associated-groups \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary
```

輸出：

```
{
    "Groups": [
        {
            "Id": "example123",
            "Name": "demo_group",
            "Arn": "arn:aws:synthetics:us-east-1:123456789012:group:example123"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAssociatedGroups。

list-group-resources

以下程式碼範例顯示如何使用 list-group-resources。

AWS CLI

傳回與指定群組相關聯之 Canary ARNs 清單

下列list-group-resources範例會傳回與名為 的群組相關聯的 Canary ARNs 清單demo_group。

```
aws synthetics list-group-resources \
    --group-identifier demo_group
```

輸出：

```
{
    "Resources": [
        "arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary"
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListGroupResources。

`list-groups`

以下程式碼範例顯示如何使用 `list-groups`。

AWS CLI

傳回帳戶中所有群組的清單

下列`list-groups`範例會傳回帳戶中所有群組的清單。

```
aws synthetics list-groups
```

輸出：

```
{
    "Groups": [
        {
            "Id": "example123",
            "Name": "demo_group",
            "Arn": "arn:aws:synthetics:us-east-1:123456789012:group:example123"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListGroups。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

範例 1：顯示與 Canary 相關聯的標籤

下列`list-tags-for-resource`範例會傳回與名為 的 Canary 相關聯的標籤`demo_canary`。

```
aws synthetics list-tags-for-resource \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary
```

輸出：

```
{
    "Tags": {
        "blueprint": "heartbeat"
    }
}
```

範例 2：顯示與群組相關聯的標籤

下列`list-tags-for-resource`範例會傳回與名為 的群組相關聯的標籤`demo_group`。

```
aws  synthetics list-tags-for-resource \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:group:example123
```

輸出：

```
{
    "Tags": {
        "team": "Devops"
    }
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`start-canary`

以下程式碼範例顯示如何使用 `start-canary`。

AWS CLI

執行 Canary

下列`start-canary`範例會執行名為 的 Canary`demo_canary`。

```
aws synthetics start-canary \
```

```
    --name demo_canary
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartCanary。

stop-canary

以下程式碼範例顯示如何使用 stop-canary。

AWS CLI

停止 Canary

下列stop-canary範例會停止名為 的 Canarydemo_canary。

```
aws synthetics stop-canary \
    --name demo_canary
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopCanary。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

範例 1：將標籤指派給 Canary

下列tag-resource範例會將標籤指派給名為 的 Canarydemo_canary。

```
aws synthetics tag-resource \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary \
    --tags blueprint=heartbeat
```

此命令不會產生輸出。

範例 2：將標籤指派給群組

下列tag-resource範例會將標籤指派給名為 的群組demo_group。

```
aws synthetics tag-resource \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:group:example123 \
    --tags team=Devops
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

範例 1：從 Canary 移除標籤

下列untag-resource範例會從名為 的 Canary 移除標籤demo_canary。

```
aws synthetics untag-resource \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary \
    --tag-keys blueprint
```

此命令不會產生輸出。

範例 2：從群組中移除標籤

下列untag-resource範例會指派 ，從名為 的群組中移除標籤demo_group。

```
aws synthetics untag-resource \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:group:example123 \
    --tag-keys team
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-canary

以下程式碼範例顯示如何使用 update-canary。

AWS CLI

更新 Canary

下列update-canary範例會更新名為 的 Canary 組態demo_canary。

```
aws synthetics update-canary \
    --name demo_canary \
    --schedule Expression="rate(15 minutes)"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的合成監控 (Canary)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateCanary。

使用 的 CodeArtifact 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 CodeArtifact 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

associate-external-connection

以下程式碼範例顯示如何使用 associate-external-connection。

AWS CLI

將外部連線新增至儲存庫

下列associate-external-connection範例會將 npmjs.com 的外部連線新增至名為 test-repo 的儲存庫。

```
aws codeartifact associate-external-connection \
    --repository test-repo \
    --domain test-domain \
    --external-connection public:npmjs
```

輸出：

```
{
    "repository": {
        "name": "test-repo",
        "administratorAccount": "111122223333",
        "domainName": "test-domain",
        "domainOwner": "111122223333",
        "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/
test-repo",
        "upstreams": [],
        "externalConnections": [
            {
                "externalConnectionName": "public:npmjs",
                "packageFormat": "npm",
                "status": "AVAILABLE"
            }
        ]
    }
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的新增外部連線。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateExternalConnection。

copy-package-versions

以下程式碼範例顯示如何使用 copy-package-versions。

AWS CLI

將套件版本從一個儲存庫複製到另一個儲存庫

下列 copy-package-versions 會將名為 test-package 的套件版本 4.0.0 和 5.0.0 從 my-repo 移至 test-repo。

```
aws codeartifact copy-package-versions \
    --domain test-domain \
    --source-repository my-repo \
    --destination-repository test-repo \
    --format npm \
    --package test-package \
    --versions '["4.0.0", "5.0.0"]'
```

輸出：

```
{
    "format": "npm",
    "package": "test-package",
    "versions": [
        {
        "version": "5.0.0",
        "revision": "REVISION-1-SAMPLE-6C81EFF7DA55CC",
        "status": "Published"
        },
        {
        "version": "4.0.0",
        "revision": "REVISION-2-SAMPLE-55C752BEE772FC",
        "status": "Published"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的在儲存庫之間複製套件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CopyPackageVersions。

create-domain

以下程式碼範例顯示如何使用 `create-domain`。

AWS CLI

建立網域

下列`create-domain`範例會建立名為 test-domain 的網域。

```
aws codeartifact create-domain \
```

```
    --domain test-domain
```

輸出：

```
{
    "domain": {
        "name": "test-domain",
        "owner": "111122223333",
        "arn": "arn:aws:codeartifact:us-west-2:111122223333:domain/test-domain",
        "status": "Active",
        "createdTime": "2020-10-20T13:16:48.559000-04:00",
        "encryptionKey": "arn:aws:kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "repositoryCount": 0,
        "assetSizeBytes": 0
    }
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的建立網域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDomain。

create-repository

以下程式碼範例顯示如何使用 create-repository。

AWS CLI

建立儲存庫

下列create-repository範例會在名為 test-domain 的網域內建立名為 test-repo 的儲存庫。

```
aws codeartifact create-repository \
    --domain test-domain \
    --domain-owner 111122223333 \
    --repository test-repo \
    --description "This is a test repository."
```

輸出：

```
{
    "repository": {
        "name": "test-repo",
```

```
        "administratorAccount": "111122223333",
        "domainName": "test-domain",
        "domainOwner": "111122223333",
        "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/
test-repo",
        "description": "This is a test repository.",
        "upstreams": [],
        "externalConnections": []
    }
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的建立網域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRepository。

delete-domain-permissions-policy

以下程式碼範例顯示如何使用 delete-domain-permissions-policy。

AWS CLI

從網域刪除許可政策文件

下列delete-domain-permissions-policy範例會從名為 test-domain 的網域刪除許可政策。

```
aws codeartifact delete-domain-permissions-policy \
    --domain test-domain
```

輸出：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "BasicDomainPolicy",
            "Action": [
                "codeartifact:GetDomainPermissionsPolicy",
                "codeartifact:ListRepositoriesInDomain",
                "codeartifact:GetAuthorizationToken",
                "codeartifact:CreateRepository"
            ],
            "Effect": "Allow",
            "Resource": "*",
```

```
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的刪除網域政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDomainPermissionsPolicy。

delete-domain

以下程式碼範例顯示如何使用 delete-domain。

AWS CLI

刪除網域

下列delete-domain範例會刪除名為 的網域test-domain。

```
aws codeartifact delete-domain \
    --domain test-domain
```

輸出：

```
{
    "domain": {
        "name": "test-domain",
        "owner": "417498243647",
        "arn": "arn:aws:codeartifact:us-west-2:417498243647:domain/test-domain",
        "status": "Deleted",
        "createdTime": "2020-10-20T13:16:48.559000-04:00",
        "encryptionKey": "arn:aws:kms:us-west-2:417498243647:key/c9fe2447-0795-4fda-
afbe-8464574ae162",
        "repositoryCount": 0,
        "assetSizeBytes": 0
    }
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的刪除網域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDomain。

delete-package-versions

以下程式碼範例顯示如何使用 delete-package-versions。

AWS CLI

刪除套件版本

下列delete-package-versions範例會刪除名為 test-package 的套件 4.0.0 版。

```
aws codeartifact delete-package-versions \
    --domain test-domain \
    --repo test-repo \
    --format npm \
    --package test-package \
    --versions 4.0.0
```

輸出：

```
{
    "successfulVersions": {
        "4.0.0": {
            "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
            "status": "Deleted"
        }
    },
    "failedVersions": {}
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的刪除套件版本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePackageVersions。

delete-repository-permissions-policy

以下程式碼範例顯示如何使用 delete-repository-permissions-policy。

AWS CLI

從儲存庫刪除許可政策

下列delete-repository-permissions-policy範例會從名為 test-repo 的儲存庫刪除許可政策。

```
aws codeartifact delete-repository-permissions-policy \
    --domain test-domain \
    --repository test-repo
```

輸出：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": [
                "codeartifact:DescribePackageVersion",
                "codeartifact:DescribeRepository",
                "codeartifact:GetPackageVersionReadme",
                "codeartifact:GetRepositoryEndpoint",
                "codeartifact:ListPackages",
                "codeartifact:ListPackageVersions",
                "codeartifact:ListPackageVersionAssets",
                "codeartifact:ListPackageVersionDependencies",
                "codeartifact:ReadFromRepository"
            ],
            "Resource": "*"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的刪除政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRepositoryPermissionsPolicy。

delete-repository

以下程式碼範例顯示如何使用 delete-repository。

AWS CLI

刪除儲存庫

下列delete-repository範例會在名為 的網域test-repo中刪除名為 的儲存庫test-domain。

```
aws codeartifact delete-repository \
    --domain test-domain \
    --repository test-repo
```

輸出：

```
{
    "repository": {
        "name": "test-repo",
        "administratorAccount": "111122223333",
        "domainName": "test-domain",
        "domainOwner": "111122223333",
        "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/
test-repo",
        "description": "This is a test repository",
        "upstreams": [],
        "externalConnections": []
    }
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的刪除儲存庫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRepository。

describe-domain

以下程式碼範例顯示如何使用 describe-domain。

AWS CLI

取得網域的相關資訊

下列describe-domain範例會傳回名為 test-domain 之網域的 DomainDescription 物件。

```
aws codeartifact describe-domain \
    --domain test-domain
```

輸出：

```
{
    "domain": {
        "name": "test-domain",
        "owner": "111122223333",
        "arn": "arn:aws:codeartifact:us-west-2:111122223333:domain/test-domain",
        "status": "Active",
        "createdTime": "2020-10-20T13:16:48.559000-04:00",
        "encryptionKey": "arn:aws:kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "repositoryCount": 2,
        "assetSizeBytes": 0,
        "s3BucketArn": "arn:aws:s3:::assets-111122223333-us-west-2"
    }
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的網域概觀。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDomain。

describe-repository

以下程式碼範例顯示如何使用 `describe-repository`。

AWS CLI

取得儲存庫的相關資訊

下列`describe-repository`範例會傳回名為 test-repo 之儲存庫的 RepositoryDescription 物件。

```
aws codeartifact describe-repository \
    --domain test-domain \
    --repository test-repo
```

輸出：

```
{
    "repository": {
        "name": "test-repo",
        "administratorAccount": "111122223333",
        "domainName": "test-domain",
        "domainOwner": "111122223333",
```

```
        "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/
test-repo",
        "description": "This is a test repository.",
        "upstreams": [],
        "externalConnections": []
    }
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的建立網域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRepository。

`disassociate-external-connection`

以下程式碼範例顯示如何使用 `disassociate-external-connection`。

AWS CLI

從儲存庫移除外部連線

下列`disassociate-external-connection`範例會從名為 test-repo 的儲存庫中移除與 npmjs.com 的外部連線。

```
aws codeartifact disassociate-external-connection \
    --repository test-repo \
    --domain test-domain \
    --external-connection public:npmjs
```

輸出：

```
{
    "repository": {
        "name": "test-repo",
        "administratorAccount": "111122223333",
        "domainName": "test-domain",
        "domainOwner": "111122223333",
        "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/
test-repo",
        "upstreams": [],
        "externalConnections": []
    }
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的移除外部連線。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateExternalConnection。

dispose-package-versions

以下程式碼範例顯示如何使用 dispose-package-versions。

AWS CLI

刪除套件版本的資產，並將其狀態設定為已處置

下列dispose-package-versions範例會刪除 test-package 4.0.0 版的資產，並將其狀態設定為已處置。

```
aws codeartifact dispose-package-versions \
    --domain test-domain \
    --repo test-repo \
    --format npm \
    --package test-package \
    --versions 4.0.0
```

輸出：

```
{
    "successfulVersions": {
        "4.0.0": {
            "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
            "status": "Disposed"
        }
    },
    "failedVersions": {}
}
```

如需詳細資訊，請參閱 CodeArtifact 使用者指南中的在 CodeArtifact 中使用套件。 AWS CodeArtifact

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisposePackageVersions。

get-authorization-token

以下程式碼範例顯示如何使用 get-authorization-token。

AWS CLI

若要取得授權字符

下列get-authorization-token範例會擷取 CodeArtifact 授權字符。

```
aws codeartifact get-authorization-token \
    --domain test-domain \
    --query authorizationToken \
    --output text
```

輸出：

```
This command will return the authorization token. You can store the output in an
 environment variable when calling the command.
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的在沒有登入命令的情況下設定 pip。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAuthorizationToken。

get-domain-permissions-policy

以下程式碼範例顯示如何使用 get-domain-permissions-policy。

AWS CLI

取得網域的許可政策文件

下列get-domain-permissions-policy範例取得連接至名為 test-domain 之網域的許可政策。

```
aws codeartifact get-domain-permissions-policy \
    --domain test-domain
```

輸出：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "BasicDomainPolicy",
            "Action": [
                "codeartifact:GetDomainPermissionsPolicy",
```

```
                "codeartifact:ListRepositoriesInDomain",
                "codeartifact:GetAuthorizationToken",
                "codeartifact:CreateRepository"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的讀取網域政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDomainPermissionsPolicy。

get-package-version-asset

以下程式碼範例顯示如何使用 get-package-version-asset。

AWS CLI

從套件版本取得資產

下列get-package-version-asset範例會擷取名為 test-package 的 npm 套件 4.0.0 版的package.tgz資產。

```
aws codeartifact get-package-version-asset \
    --domain test-domain \
    --repository test-repo \
    --format npm \
    --package test-package \
    --package-version 4.0.0 \
    --asset 'package.tgz' \
    outfileName
```

輸出：

```
The output for this command will also store the raw asset in the file provided in
 place of outfileName.
```

```
{
    "assetName": "package.tgz",
    "packageVersion": "4.0.0",
    "packageVersionRevision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs="
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的列出套件版本資產。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPackageVersionAsset。

`get-package-version-readme`

以下程式碼範例顯示如何使用 `get-package-version-readme`。

AWS CLI

取得套件版本的讀我檔案

下列`get-package-version-readme`範例會擷取名為 test-package 之 npm 套件 4.0.0 版的 readme 檔案。

```
aws codeartifact get-package-version-readme \
    --domain test-domain \
    --repo test-repo \
    --format npm \
    --package test-package \
    --package-version 4.0.0
```

輸出：

```
{
    "format": "npm",
    "package": "test-package",
    "version": "4.0.0",
    "readme": "<div align=\"center\">\n   <a href=\https://github.com/test-package/
testpack\"> ... more content ... \n",
    "versionRevision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs="
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的檢視套件版本讀我檔案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPackageVersionReadme。

get-repository-endpoint

以下程式碼範例顯示如何使用 get-repository-endpoint。

AWS CLI

取得儲存庫的 URL 端點

下列get-repository-endpoint範例會傳回 test-repo 儲存庫的 npm 端點。

```
aws codeartifact get-repository-endpoint \
    --domain test-domain \
    --repository test-repo \
    --format npm
```

輸出：

```
{
    "repositoryEndpoint": "https://test-domain-111122223333.d.codeartifact.us-
west-2.amazonaws.com/npm/test-repo/"
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的連線至儲存庫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRepositoryEndpoint。

get-repository-permissions-policy

以下程式碼範例顯示如何使用 get-repository-permissions-policy。

AWS CLI

取得儲存庫的許可政策文件

下列get-repository-permissions-policy範例取得連接至名為 test-repo 之儲存庫的許可政策。

```
aws codeartifact get-repository-permissions-policy \
    --domain test-domain \
    --repository test-repo
```

輸出：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": [
                "codeartifact:DescribePackageVersion",
                "codeartifact:DescribeRepository",
                "codeartifact:GetPackageVersionReadme",
                "codeartifact:GetRepositoryEndpoint",
                "codeartifact:ListPackages",
                "codeartifact:ListPackageVersions",
                "codeartifact:ListPackageVersionAssets",
                "codeartifact:ListPackageVersionDependencies",
                "codeartifact:ReadFromRepository"
            ],
            "Resource": "*"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的讀取政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRepositoryPermissionsPolicy。

`list-domains`

以下程式碼範例顯示如何使用 `list-domains`。

AWS CLI

列出網域

下列`list-domains`範例會傳回進行呼叫之 AWS 帳戶擁有的所有網域的摘要。

```
aws codeartifact list-domains
```

輸出：

```
{
    "domains": [
        {
            "name": "my-domain",
            "owner": "111122223333",
            "status": "Active",
            "encryptionKey": "arn:aws:kms:us-west-2:111122223333:key/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        },
        {
            "name": "test-domain",
            "owner": "111122223333",
            "status": "Active",
            "encryptionKey": "arn:aws:kms:us-west-2:111122223333:key/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
        }
    ]
}
```

如需詳細資訊，請參閱 CodeArtifact 使用者指南中的在 CodeArtifact 中使用網域。 AWS CodeArtifact

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDomains。

`list-package-version-assets`

以下程式碼範例顯示如何使用 `list-package-version-assets`。

AWS CLI

檢視套件版本的資產

下列`list-package-version-assets`範例會擷取名為 test-package 的 npm 套件 4.0.0 版的資產。

```
aws codeartifact list-package-version-assets \
    --domain test-domain \
    --repo test-repo \
    --format npm \
    --package test-package \
    --package-version 4.0.0
```

輸出：

```
{
    "format": "npm",
    "package": "test-package",
    "version": "4.0.0",
    "versionRevision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
    "assets": [
        {
            "name": "package.tgz",
            "size": 316680,
            "hashes": {
                "MD5": "60078ec6d9e76b89fb55c860832742b2",
                "SHA-1": "b44a9b6297bcb698f1c51a3545a2b3b368d59c52",
                "SHA-256":
 "d2aa8c6afc3c8591765785a37d1c5acae482a8eb3ab9729ed28922692454f2e2",
                "SHA-512":
 "3e585d15c8a594e20d7de57b362ea81754c011acb2641a19f1b72c8531ea39825896bab344ae616a0a5a824cb9
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的列出套件版本資產。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPackageVersionAssets。

list-package-version-dependencies

以下程式碼範例顯示如何使用 list-package-version-dependencies。

AWS CLI

檢視套件版本的相依性

下列list-package-version-dependencies範例會擷取名為 test-package 之 npm 套件 4.0.0 版的相依性。

```
aws codeartifact list-package-version-dependencies \
    --domain test-domain \
    --repo test-repo \
    --format npm \
```

```
    --package test-package \
    --package-version 4.0.0
```

輸出：

```
{
    "format": "npm",
    "package": "test-package",
    "version": "4.0.0",
    "versionRevision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
    "dependencies": [
        {
            "namespace": "testns",
            "package": "testdep1",
            "dependencyType": "regular",
            "versionRequirement": "1.8.5"
        },
        {
            "namespace": "testns",
            "package": "testdep2",
            "dependencyType": "regular",
            "versionRequirement": "1.8.5"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的檢視和更新套件版本詳細資訊和相依性。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPackageVersionDependencies。

`list-package-versions`

以下程式碼範例顯示如何使用 `list-package-versions`。

AWS CLI

列出套件的套件版本

下列`list-package-versions`範例會傳回名為 之套件的套件版本清單`kind-of`。

```
aws codeartifact list-package-versions \
```

```
    --package kind-of \
    --domain test-domain \
    --repository test-repo \
    --format npm
```

輸出：

```
{
    "defaultDisplayVersion": "1.0.1",
    "format": "npm",
    "package": "kind-of",
    "versions": [
        {
            "version": "1.0.1",
            "revision": "REVISION-SAMPLE-1-C7F4S5E9B772FC",
            "status": "Published"
        },
        {
            "version": "1.0.0",
            "revision": "REVISION-SAMPLE-2-C752BEEF6D2CFC",
            "status": "Published"
        },
        {
            "version": "0.1.2",
            "revision": "REVISION-SAMPLE-3-654S65A5C5E1FC",
            "status": "Published"
        },
        {
            "version": "0.1.1",
            "revision": "REVISION-SAMPLE-1-C7F4S5E9B772FC"",
            "status": "Published"
        },
        {
            "version": "0.1.0",
            "revision": "REVISION-SAMPLE-4-AF669139B772FC",
            "status": "Published"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的列出套件版本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPackageVersions。

list-packages

以下程式碼範例顯示如何使用 list-packages。

AWS CLI

列出儲存庫中的套件

下列list-packages範例列出名為 之網域中名為 的儲存庫test-repo中的套件test-domain。

```
aws codeartifact list-packages \
    --domain test-domain \
    --repository test-repo
```

輸出：

```
{
    "packages": [
        {
            "format": "npm",
            "package": "lodash"
        }
        {
            "format": "python",
            "package": "test-package"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的列出套件名稱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPackages。

list-repositories-in-domain

以下程式碼範例顯示如何使用 list-repositories-in-domain。

AWS CLI

列出網域中的儲存庫

下列list-repositories-in-domain範例會傳回 test-domain 網域中所有儲存庫的摘要。

```
aws codeartifact list-repositories-in-domain \
    --domain test-domain
```

輸出：

```
{
    "repositories": [
        {
            "name": "test-repo",
            "administratorAccount": "111122223333",
            "domainName": "test-domain",
            "domainOwner": "111122223333",
            "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-
domain/test-repo",
            "description": "This is a test repository."
        },
        {
            "name": "test-repo2",
            "administratorAccount": "111122223333",
            "domainName": "test-domain",
            "domainOwner": "111122223333",
            "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-
domain/test-repo2",
            "description": "This is a test repository."
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的列出儲存庫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRepositoriesInDomain。

list-repositories

以下程式碼範例顯示如何使用 `list-repositories`。

AWS CLI

列出儲存庫

下列`list-repositories`範例會傳回進行呼叫之 AWS 帳戶所擁有的網域中所有儲存庫的摘要。

```
aws codeartifact list-repositories
```

輸出：

```
{
    "repositories": [
        {
            "name": "npm-store",
            "administratorAccount": "111122223333",
            "domainName": "my-domain",
            "domainOwner": "111122223333",
            "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/my-
domain/npm-store",
            "description": "Provides npm artifacts from npm, Inc."
        },
        {
            "name": "target-repo",
            "administratorAccount": "111122223333",
            "domainName": "my-domain",
            "domainOwner": "111122223333",
            "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/my-
domain/target-repo",
            "description": "test target repo"
        },
        {
            "name": "test-repo2",
            "administratorAccount": "111122223333",
            "domainName": "test-domain",
            "domainOwner": "111122223333",
            "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-
domain/test-repo2",
            "description": "This is a test repository."
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的列出儲存庫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRepositories。

login

以下程式碼範例顯示如何使用 login。

AWS CLI

使用登入命令設定對儲存庫的身分驗證

下列login範例會使用名為 test-repo 的儲存庫，在名為 test-domain 的網域中設定 npm 套件管理員。

```
aws codeartifact login \
    --domain test-domain \
    --repository test-repo \
    --tool npm
```

輸出：

```
Successfully configured npm to use AWS CodeArtifact repository https://test-
domain-111122223333.d.codeartifact.us-west-2.amazonaws.com/npm/test-repo/
Login expires in 12 hours at 2020-11-12 01:53:16-05:00
```

如需詳細資訊，請參閱 CodeArtifact AWS 使用者指南中的 CLI 入門。 AWS CodeArtifact

- 如需 API 詳細資訊，請參閱登入AWS CLI 命令參考。

put-domain-permissions-policy

以下程式碼範例顯示如何使用 put-domain-permissions-policy。

AWS CLI

將許可政策連接至網域

下列put-domain-permissions-policy範例會將 policy.json 檔案中定義的許可政策連接至名為 test-domain 的網域。

```
aws codeartifact put-domain-permissions-policy \
    --domain test-domain \
    --policy-document file://PATH/TO/policy.json
```

輸出：

```
{
    "policy": {
        "resourceArn": "arn:aws:codeartifact:region-id:111122223333:domain/test-
domain",
        "document": "{ ...policy document content...}",
        "revision": "MQlyyTQRASRU3HB58gBtSDHXG7Q3hvxxxxxxx="
    }
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的設定網域政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutDomainPermissionsPolicy。

put-repository-permissions-policy

以下程式碼範例顯示如何使用 put-repository-permissions-policy。

AWS CLI

將許可政策連接至儲存庫

下列put-repository-permissions-policy範例會將 policy.json 檔案中定義的許可政策連接至名為 test-repo 的儲存庫。

```
aws codeartifact put-repository-permissions-policy \
    --domain test-domain \
    --repository test-repo \
    --policy-document file://PATH/TO/policy.json
```

輸出：

```
{
    "policy": {
        "resourceArn": "arn:aws:codeartifact:region-id:111122223333:repository/test-
domain/test-repo",
        "document": "{ ...policy document content...}",
        "revision": "MQlyyTQRASRU3HB58gBtSDHXG7Q3hvxxxxxxx="
    }
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的設定政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutRepositoryPermissionsPolicy。

update-package-versions-status

以下程式碼範例顯示如何使用 update-package-versions-status。

AWS CLI

更新套件版本狀態

下列update-package-versions-status範例會將 test-package 套件 4.0.0 版的狀態更新為已封存。

```
aws codeartifact update-package-versions-status \
    --domain test-domain \
    --repo test-repo \
    --format npm \
    --package test-package \
    --versions 4.0.0 \
    --target-status Archived
```

輸出：

```
{
    "successfulVersions": {
        "4.0.0": {
            "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
            "status": "Archived"
        }
    },
    "failedVersions": {}
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的更新套件版本狀態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePackageVersionsStatus。

update-repository

以下程式碼範例顯示如何使用 update-repository。

AWS CLI

更新儲存庫

下列update-repository範例會將名為 test-repo 的儲存庫在名為 test-domain 的網域中的描述更新為「這是更新的描述」。

```
aws codeartifact update-repository \
    --domain test-domain \
    --repository test-repo \
    --description "this is an updated description"
```

輸出：

```
{
    "repository": {
        "name": "test-repo",
        "administratorAccount": "111122223333",
        "domainName": "test-domain",
        "domainOwner": "111122223333",
        "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/
test-repo",
        "description": "this is an updated description",
        "upstreams": [],
        "externalConnections": []
    }
}
```

如需詳細資訊，請參閱 AWS CodeArtifact 使用者指南中的檢視或修改儲存庫組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRepository。

使用 的 CodeBuild 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 CodeBuild 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

batch-delete-builds

以下程式碼範例顯示如何使用 batch-delete-builds。

AWS CLI

在 AWS CodeBuild 中刪除組建。

下列batch-delete-builds範例會刪除 CodeBuild 中具有指定 IDs組建。

```
aws codebuild batch-delete-builds --ids my-build-project-one:a1b2c3d4-5678-9012-abcd-11111EXAMPLE my-build-project-two:a1b2c3d4-5678-9012-abcd-22222EXAMPLE
```

輸出：

```
{
    "buildsNotDeleted": [
        {
            "id": "arn:aws:codebuild:us-west-2:123456789012:build/my-build-project-one:a1b2c3d4-5678-9012-abcd-11111EXAMPLE",
            "statusCode": "BUILD_IN_PROGRESS"
        }
    ],
    "buildsDeleted": [
        "arn:aws:codebuild:us-west-2:123456789012:build/my-build-project-two:a1b2c3d4-5678-9012-abcd-22222EXAMPLE"
    ]
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的刪除組建 (AWS CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDeleteBuilds。

batch-get-build-batches

以下程式碼範例顯示如何使用 batch-get-build-batches。

AWS CLI

在 AWS CodeBuild 中檢視組建的詳細資訊。

下列batch-get-build-batches範例取得 CodeBuild 中具有指定 IDs之組建批次的相關資訊。

```
aws codebuild batch-get-build-batches \
    --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE
```

輸出：

```
{
    "buildBatches": [
        {
            "id": "codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",
            "arn": "arn:aws:codebuild:us-west-2:123456789012:build-batch/codebuild-
demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",
            "startTime": "2020-11-03T21:52:20.775000+00:00",
            "endTime": "2020-11-03T21:56:59.784000+00:00",
            "currentPhase": "SUCCEEDED",
            "buildBatchStatus": "SUCCEEDED",
            "resolvedSourceVersion": "0a6546f68309560d08a310daac92314c4d378f6b",
            "projectName": "codebuild-demo-project",
            "phases": [
                {
                    "phaseType": "SUBMITTED",
                    "phaseStatus": "SUCCEEDED",
                    "startTime": "2020-11-03T21:52:20.775000+00:00",
                    "endTime": "2020-11-03T21:52:20.976000+00:00",
                    "durationInSeconds": 0
                },
                {
                    "phaseType": "DOWNLOAD_BATCHSPEC",
                    "phaseStatus": "SUCCEEDED",
                    "startTime": "2020-11-03T21:52:20.976000+00:00",
                    "endTime": "2020-11-03T21:52:57.401000+00:00",
                    "durationInSeconds": 36
                },
                {
                    "phaseType": "IN_PROGRESS",
                    "phaseStatus": "SUCCEEDED",
                    "startTime": "2020-11-03T21:52:57.401000+00:00",
                    "endTime": "2020-11-03T21:56:59.751000+00:00",
                    "durationInSeconds": 242
                },
                {
                    "phaseType": "COMBINE_ARTIFACTS",
```

```
                    "phaseStatus": "SUCCEEDED",
                    "startTime": "2020-11-03T21:56:59.751000+00:00",
                    "endTime": "2020-11-03T21:56:59.784000+00:00",
                    "durationInSeconds": 0
                },
                {
                    "phaseType": "SUCCEEDED",
                    "startTime": "2020-11-03T21:56:59.784000+00:00"
                }
            ],
            "source": {
                "type": "GITHUB",
                "location": "https://github.com/my-repo/codebuild-demo-project.git",
                "gitCloneDepth": 1,
                "gitSubmodulesConfig": {
                    "fetchSubmodules": false
                },
                "reportBuildStatus": false,
                "insecureSsl": false
            },
            "secondarySources": [],
            "secondarySourceVersions": [],
            "artifacts": {
                "location": ""
            },
            "secondaryArtifacts": [],
            "cache": {
                "type": "NO_CACHE"
            },
            "environment": {
                "type": "LINUX_CONTAINER",
                "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
                "computeType": "BUILD_GENERAL1_SMALL",
                "environmentVariables": [],
                "privilegedMode": false,
                "imagePullCredentialsType": "CODEBUILD"
            },
            "logConfig": {
                "cloudWatchLogs": {
                    "status": "ENABLED"
                },
                "s3Logs": {
                    "status": "DISABLED",
                    "encryptionDisabled": false
```

```
                }
            },
            "buildTimeoutInMinutes": 60,
            "queuedTimeoutInMinutes": 480,
            "complete": true,
            "initiator": "Strohm",
            "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
            "buildBatchNumber": 6,
            "buildBatchConfig": {
                "serviceRole": "arn:aws:iam::123456789012:role/service-role/
codebuild-demo-project",
                "restrictions": {
                    "maximumBuildsAllowed": 100
                },
                "timeoutInMins": 480
            },
            "buildGroups": [
                {
                    "identifier": "DOWNLOAD_SOURCE",
                    "ignoreFailure": false,
                    "currentBuildSummary": {
                        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/
codebuild-demo-project:379737d8-bc35-48ec-97fd-776d27545315",
                        "requestedOn": "2020-11-03T21:52:21.394000+00:00",
                        "buildStatus": "SUCCEEDED",
                        "primaryArtifact": {
                            "type": "no_artifacts",
                            "identifier": "DOWNLOAD_SOURCE"
                        },
                        "secondaryArtifacts": []
                    }
                },
                {
                    "identifier": "linux_small",
                    "dependsOn": [],
                    "ignoreFailure": false,
                    "currentBuildSummary": {
                        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/
codebuild-demo-project:dd785171-ed84-4bb6-8ede-ceeb86e54bdb",
                        "requestedOn": "2020-11-03T21:52:57.604000+00:00",
                        "buildStatus": "SUCCEEDED",
                        "primaryArtifact": {
                            "type": "no_artifacts",
                            "identifier": "linux_small"
```

```
                        },
                        "secondaryArtifacts": []
                    }
                },
                {
                    "identifier": "linux_medium",
                    "dependsOn": [
                        "linux_small"
                    ],
                    "ignoreFailure": false,
                    "currentBuildSummary": {
                        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/
codebuild-demo-project:97cf7bd4-5313-4786-8243-4aef350a1267",
                        "requestedOn": "2020-11-03T21:54:18.474000+00:00",
                        "buildStatus": "SUCCEEDED",
                        "primaryArtifact": {
                            "type": "no_artifacts",
                            "identifier": "linux_medium"
                        },
                        "secondaryArtifacts": []
                    }
                },
                {
                    "identifier": "linux_large",
                    "dependsOn": [
                        "linux_medium"
                    ],
                    "ignoreFailure": false,
                    "currentBuildSummary": {
                        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/
codebuild-demo-project:60a194cd-0d03-4337-9db1-d41476a17d27",
                        "requestedOn": "2020-11-03T21:55:39.203000+00:00",
                        "buildStatus": "SUCCEEDED",
                        "primaryArtifact": {
                            "type": "no_artifacts",
                            "identifier": "linux_large"
                        },
                        "secondaryArtifacts": []
                    }
                }
            ]
        }
    ],
    "buildBatchesNotFound": []
```

```
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的 CodeBuild <https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html>)__ 中的批次建置。 AWS CodeBuild

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetBuildBatches。

batch-get-builds

以下程式碼範例顯示如何使用 batch-get-builds。

AWS CLI

在 AWS CodeBuild 中檢視組建的詳細資訊。

下列batch-get-builds範例取得 CodeBuild 中具有指定 IDs 之組建的相關資訊。

```
aws codebuild batch-get-builds --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE
```

輸出：

```
{
    "buildsNotFound": [],
    "builds": [
        {
            "artifacts": {
                "md5sum": "0e95edf915048a0c22efe6d139fff837",
                "location": "arn:aws:s3:::codepipeline-us-west-2-820783811474/
CodeBuild-Python-Pip/BuildArtif/6DJsqQa",
                "encryptionDisabled": false,
                "sha256sum":
 "cfa0df33a090966a737f64ae4fe498969fdc842a0c9aec540bf93c37ac0d05a2"
            },
            "logs": {
                "cloudWatchLogs": {
                    "status": "ENABLED"
                },
                "s3Logs": {
                    "status": "DISABLED"
                },
                "streamName": "46472baf-8f6b-43c2-9255-b3b963af2732",
```

```
                "groupName": "/aws/codebuild/codebuild-demo-project",
                "deepLink": "https://console.aws.amazon.com/cloudwatch/
home?region=us-west-2#logEvent:group=/aws/codebuild/codebuild-demo-
project;stream=46472baf-8f6b-43c2-9255-b3b963af2732"
            },
            "timeoutInMinutes": 60,
            "environment": {
                "privilegedMode": false,
                "computeType": "BUILD_GENERAL1_MEDIUM",
                "image": "aws/codebuild/windows-base:1.0",
                "environmentVariables": [],
                "type": "WINDOWS_CONTAINER"
            },
            "projectName": "codebuild-demo-project",
            "buildComplete": true,
            "source": {
                "gitCloneDepth": 1,
                "insecureSsl": false,
                "type": "CODEPIPELINE"
            },
            "buildStatus": "SUCCEEDED",
            "secondaryArtifacts": [],
            "phases": [
                {
                    "durationInSeconds": 0,
                    "startTime": 1548717462.122,
                    "phaseType": "SUBMITTED",
                    "endTime": 1548717462.484,
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 0,
                    "startTime": 1548717462.484,
                    "phaseType": "QUEUED",
                    "endTime": 1548717462.775,
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 34,
                    "endTime": 1548717496.909,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
```

```
                    }
                ],
                "startTime": 1548717462.775,
                "phaseType": "PROVISIONING",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 15,
                "endTime": 1548717512.555,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548717496.909,
                "phaseType": "DOWNLOAD_SOURCE",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 0,
                "endTime": 1548717512.734,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548717512.555,
                "phaseType": "INSTALL",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 0,
                "endTime": 1548717512.924,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548717512.734,
                "phaseType": "PRE_BUILD",
                "phaseStatus": "SUCCEEDED"
```

```
                },
                {
                    "durationInSeconds": 9,
                    "endTime": 1548717522.254,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548717512.924,
                    "phaseType": "BUILD",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 3,
                    "endTime": 1548717525.498,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548717522.254,
                    "phaseType": "POST_BUILD",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 9,
                    "endTime": 1548717534.646,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548717525.498,
                    "phaseType": "UPLOAD_ARTIFACTS",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 2,
                    "endTime": 1548717536.846,
                    "contexts": [
```

```
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548717534.646,
                    "phaseType": "FINALIZING",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "startTime": 1548717536.846,
                    "phaseType": "COMPLETED"
                }
            ],
            "startTime": 1548717462.122,
            "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
            "initiator": "codepipeline/CodeBuild-Pipeline",
            "secondarySources": [],
            "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-
codebuild-service-role",
            "currentPhase": "COMPLETED",
            "id": "codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",
            "cache": {
                "type": "NO_CACHE"
            },
            "sourceVersion": "arn:aws:s3:::codepipeline-us-west-2-820783811474/
CodeBuild-Python-Pip/SourceArti/1TspnN3.zip",
            "endTime": 1548717536.846,
            "arn": "arn:aws:codebuild:us-west-2:123456789012:build/codebuild-demo-
project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",
            "queuedTimeoutInMinutes": 480,
            "resolvedSourceVersion": "f2194c1757bbdcb0f8f229254a4b3c8b27d43e0b"
        },
        {
            "artifacts": {
                "md5sum": "",
                "overrideArtifactName": false,
                "location": "arn:aws:s3:::my-artifacts/codebuild-demo-project",
                "encryptionDisabled": false,
                "sha256sum": ""
            },
            "logs": {
                "cloudWatchLogs": {
                    "status": "ENABLED"
```

```
                },
                "s3Logs": {
                    "status": "DISABLED"
                },
                "streamName": "4dea3ca4-20ec-4898-b22a-a9eb9292775d",
                "groupName": "/aws/codebuild/codebuild-demo-project",
                "deepLink": "https://console.aws.amazon.com/cloudwatch/
home?region=us-west-2#logEvent:group=/aws/codebuild/codebuild-demo-
project;stream=4dea3ca4-20ec-4898-b22a-a9eb9292775d"
            },
            "timeoutInMinutes": 60,
            "environment": {
                "privilegedMode": false,
                "computeType": "BUILD_GENERAL1_MEDIUM",
                "image": "aws/codebuild/windows-base:1.0",
                "environmentVariables": [],
                "type": "WINDOWS_CONTAINER"
            },
            "projectName": "codebuild-demo-project",
            "buildComplete": true,
            "source": {
                "gitCloneDepth": 1,
                "location": "https://github.com/my-repo/codebuild-demo-project.git",
                "insecureSsl": false,
                "reportBuildStatus": false,
                "type": "GITHUB"
            },
            "buildStatus": "SUCCEEDED",
            "secondaryArtifacts": [],
            "phases": [
                {
                    "durationInSeconds": 0,
                    "startTime": 1548716241.89,
                    "phaseType": "SUBMITTED",
                    "endTime": 1548716242.241,
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 0,
                    "startTime": 1548716242.241,
                    "phaseType": "QUEUED",
                    "endTime": 1548716242.536,
                    "phaseStatus": "SUCCEEDED"
                },
```

```
            {
                "durationInSeconds": 33,
                "endTime": 1548716276.171,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548716242.536,
                "phaseType": "PROVISIONING",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 15,
                "endTime": 1548716291.809,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548716276.171,
                "phaseType": "DOWNLOAD_SOURCE",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 0,
                "endTime": 1548716291.993,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548716291.809,
                "phaseType": "INSTALL",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 0,
                "endTime": 1548716292.191,
                "contexts": [
                    {
```

```
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548716291.993,
                "phaseType": "PRE_BUILD",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 9,
                "endTime": 1548716301.622,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548716292.191,
                "phaseType": "BUILD",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 3,
                "endTime": 1548716304.783,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548716301.622,
                "phaseType": "POST_BUILD",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 8,
                "endTime": 1548716313.775,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548716304.783,
```

```
                    "phaseType": "UPLOAD_ARTIFACTS",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 2,
                    "endTime": 1548716315.935,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548716313.775,
                    "phaseType": "FINALIZING",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "startTime": 1548716315.935,
                    "phaseType": "COMPLETED"
                }
            ],
            "startTime": 1548716241.89,
            "secondarySourceVersions": [],
            "initiator": "my-codebuild-project",
            "arn": "arn:aws:codebuild:us-west-2:123456789012:build/codebuild-demo-
project:815e755f-bade-4a7e-80f0-efe51EXAMPLE",
            "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
            "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-
codebuild-service-role",
            "currentPhase": "COMPLETED",
            "id": "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE",
            "cache": {
                "type": "NO_CACHE"
            },
            "endTime": 1548716315.935,
            "secondarySources": [],
            "queuedTimeoutInMinutes": 480,
            "resolvedSourceVersion": "f2194c1757bbdcb0f8f229254a4b3c8b27d43e0b"
        }
    ]
}
```


如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的檢視建置詳細資訊 (AWS CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetBuilds。

batch-get-projects

以下程式碼範例顯示如何使用 batch-get-projects。

AWS CLI

取得 AWS CodeBuild 組建專案名稱的清單。

下列batch-get-projects範例會取得依名稱指定的 CodeBuild 組建專案清單。

```
aws codebuild batch-get-projects --names codebuild-demo-project codebuild-demo-project2 my-other-demo-project
```

在下列輸出中，projectsNotFound陣列會列出已指定但找不到的任何建置專案名稱。projects陣列會列出每個找到資訊的組建專案詳細資訊。

```
{
    "projectsNotFound": [],
    "projects": [
        {
            "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
            "name": "codebuild-demo-project2",
            "queuedTimeoutInMinutes": 480,
            "timeoutInMinutes": 60,
            "source": {
                "buildspec": "version: 0.2\n\n#env:\n  #variables:\n     # key: \"value\"\n     # key: \"value\"\n  #parameter-store:\n     # key: \"value\"\n  # key:\"value\"\n\nphases:\n  #install:\n    #commands:\n      # - command\n  # - command\n  #pre_build:\n    #commands:\n      # - command\n      # - command\n  build:\n    commands:\n      # - command\n      # - command\n  #post_build:\n    #commands:\n      # - command\n      # - command\n#artifacts:\n  #files:\n    # - location\n    # - location\n  #name: $(date +%Y-%m-%d)\n  #discard-paths: yes\n  #base-directory: location\n#cache:\n  #paths:\n    # - paths",
                "type": "NO_SOURCE",
                "insecureSsl": false,
                "gitCloneDepth": 1
            },
            "artifacts": {
                "type": "NO_ARTIFACTS"
            },
            "badge": {
```

```
                "badgeEnabled": false
            },
            "lastModified": 1540588091.108,
            "created": 1540588091.108,
            "arn": "arn:aws:codebuild:us-west-2:123456789012:project/test-for-
sample",
            "secondarySources": [],
            "secondaryArtifacts": [],
            "cache": {
                "type": "NO_CACHE"
            },
            "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-test-
role",
            "environment": {
                "image": "aws/codebuild/java:openjdk-8",
                "privilegedMode": true,
                "type": "LINUX_CONTAINER",
                "computeType": "BUILD_GENERAL1_SMALL",
                "environmentVariables": []
            },
            "tags": []
        },
        {
            "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
            "name": "my-other-demo-project",
            "queuedTimeoutInMinutes": 480,
            "timeoutInMinutes": 60,
            "source": {
                "location": "https://github.com/iversonic/codedeploy-sample.git",
                "reportBuildStatus": false,
                "buildspec": "buildspec.yml",
                "insecureSsl": false,
                "gitCloneDepth": 1,
                "type": "GITHUB",
                "auth": {
                    "type": "OAUTH"
                }
            },
            "artifacts": {
                "type": "NO_ARTIFACTS"
            },
            "badge": {
                "badgeEnabled": false
            },
```

```
            "lastModified": 1523401711.73,
            "created": 1523401711.73,
            "arn": "arn:aws:codebuild:us-west-2:123456789012:project/Project2",
            "cache": {
                "type": "NO_CACHE"
            },
            "serviceRole": "arn:aws:iam::123456789012:role/service-role/codebuild-
Project2-service-role",
            "environment": {
                "image": "aws/codebuild/nodejs:4.4.7",
                "privilegedMode": false,
                "type": "LINUX_CONTAINER",
                "computeType": "BUILD_GENERAL1_SMALL",
                "environmentVariables": []
            },
            "tags": []
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的檢視組建專案的詳細資訊 (AWS CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetProjects。

batch-get-report-groups

以下程式碼範例顯示如何使用 batch-get-report-groups。

AWS CLI

取得 AWS CodeBuild 中一或多個報告群組的相關資訊。

下列batch-get-report-groups範例會使用指定的 ARN 擷取報告群組的相關資訊。

```
aws codebuild batch-get-report-groups \
    --report-group-arns arn:aws:codebuild:<region-ID>:<user-ID>:report-group/
<report-group-name>
```

輸出：

```
{
    "reportGroups": [
        {
```

```
            "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-
group-name>",
            "name": "report-group-name",
            "type": "TEST",
            "exportConfig": {
                "exportConfigType": "NO_EXPORT"
            },
            "created": "2020-10-01T18:04:08.466000+00:00",
            "lastModified": "2020-10-01T18:04:08.466000+00:00",
            "tags": []
        }
    ],
    "reportGroupsNotFound": []
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的使用報告群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetReportGroups。

batch-get-reports

以下程式碼範例顯示如何使用 batch-get-reports。

AWS CLI

取得 AWS CodeBuild 中一或多個報告的相關資訊。

下列batch-get-reports範例會使用指定的 ARNs 擷取報告的相關資訊。

```
aws codebuild batch-get-reports \
    --report-arns arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-
name>:<report 1 ID> arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-
name>:<report 2 ID>
```

輸出：

```
{
    "reports": [
        {
            "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-
name>:<report 1 ID>",
            "type": "TEST",
            "name": "<report-group-name>",
```

```
            "reportGroupArn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/
<report-group-name>",
            "executionId": "arn:aws:codebuild:<region-ID>:<user-ID>:build/test-
reports:<ID>",
            "status": "FAILED",
            "created": "2020-10-01T11:25:22.531000-07:00",
            "expired": "2020-10-31T11:25:22-07:00",
            "exportConfig": {
                "exportConfigType": "NO_EXPORT"
            },
            "truncated": false,
            "testSummary": {
                "total": 28,
                "statusCounts": {
                    "ERROR": 5,
                    "FAILED": 1,
                    "SKIPPED": 4,
                    "SUCCEEDED": 18,
                    "UNKNOWN": 0
                },
                "durationInNanoSeconds": 94000000
            }
        },
        {
            "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-
name>:<report 2 ID>",
            "type": "TEST",
            "name": "<report-group-name>",
            "reportGroupArn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/
<report-group-name>",
            "executionId": "arn:aws:codebuild:<region-ID>:<user-ID>:build/test-
reports:<ID>",
            "status": "FAILED",
            "created": "2020-10-01T11:13:05.816000-07:00",
            "expired": "2020-10-31T11:13:05-07:00",
            "exportConfig": {
                "exportConfigType": "NO_EXPORT"
            },
            "truncated": false,
            "testSummary": {
                "total": 28,
                "statusCounts": {
                    "ERROR": 5,
                    "FAILED": 1,
```

```
                    "SKIPPED": 4,
                    "SUCCEEDED": 18,
                    "UNKNOWN": 0
                },
                "durationInNanoSeconds": 94000000
            }
        }
    ],
    "reportsNotFound": []
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的使用報告。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetReports。

create-project

以下程式碼範例顯示如何使用 create-project。

AWS CLI

範例 1：建立 AWS CodeBuild 組建專案

下列create-project範例使用來自 S3 儲存貯體的來源檔案建立 CodeBuild 建置專案

```
aws codebuild create-project \
    --name "my-demo-project" \
    --source "{\"type\": \"S3\",\"location\": \"codebuild-us-west-2-123456789012-
input-bucket/my-source.zip\"}" \
    --artifacts {"\"type\": \"S3\",\"location\": \"codebuild-us-west-2-123456789012-
output-bucket\""} \
    --environment "{\"type\": \"LINUX_CONTAINER\",\"image\": \"aws/codebuild/
standard:1.0\",\"computeType\": \"BUILD_GENERAL1_SMALL\"}" \
    --service-role "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role"
```

輸出：

```
{
    "project": {
        "arn": "arn:aws:codebuild:us-west-2:123456789012:project/my-demo-project",
        "name": "my-cli-demo-project",
        "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
```

```
        "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role",
        "lastModified": 1556839783.274,
        "badge": {
            "badgeEnabled": false
        },
        "queuedTimeoutInMinutes": 480,
        "environment": {
            "image": "aws/codebuild/standard:1.0",
            "computeType": "BUILD_GENERAL1_SMALL",
            "type": "LINUX_CONTAINER",
            "imagePullCredentialsType": "CODEBUILD",
            "privilegedMode": false,
            "environmentVariables": []
        },
        "artifacts": {
            "location": "codebuild-us-west-2-123456789012-output-bucket",
            "name": "my-cli-demo-project",
            "namespaceType": "NONE",
            "type": "S3",
            "packaging": "NONE",
            "encryptionDisabled": false
        },
        "source": {
            "type": "S3",
            "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source.zip",
            "insecureSsl": false
        },
        "timeoutInMinutes": 60,
        "cache": {
            "type": "NO_CACHE"
        },
        "created": 1556839783.274
    }
}
```

範例 2：使用參數的 JSON 輸入檔案建立 AWS CodeBuild 組建專案

下列`create-project`範例會透過在 JSON 輸入檔案中傳遞所有必要參數來建立 CodeBuild 建置專案。僅使用 執行 命令來建立輸入檔案範本`--generate-cli-skeleton parameter`。

```
aws codebuild create-project --cli-input-json file://create-project.json
```

輸入 JSON 檔案`create-project.json`包含下列內容：

```
{
    "name": "codebuild-demo-project",
    "source": {
        "type": "S3",
        "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
    },
    "artifacts": {
        "type": "S3",
        "location": "codebuild-region-ID-account-ID-output-bucket"
    },
    "environment": {
        "type": "LINUX_CONTAINER",
        "image": "aws/codebuild/standard:1.0",
        "computeType": "BUILD_GENERAL1_SMALL"
    },
    "serviceRole": "serviceIAMRole"
}
```

輸出：

```
{
    "project": {
        "name": "codebuild-demo-project",
        "serviceRole": "serviceIAMRole",
        "tags": [],
        "artifacts": {
            "packaging": "NONE",
            "type": "S3",
            "location": "codebuild-region-ID-account-ID-output-bucket",
            "name": "message-util.zip"
        },
        "lastModified": 1472661575.244,
        "timeoutInMinutes": 60,
        "created": 1472661575.244,
        "environment": {
            "computeType": "BUILD_GENERAL1_SMALL",
            "image": "aws/codebuild/standard:1.0",
            "type": "LINUX_CONTAINER",
            "environmentVariables": []
        },
        "source": {
```

```
            "type": "S3",
            "location": "codebuild-region-ID-account-ID-input-bucket/
MessageUtil.zip"
        },
        "encryptionKey": "arn:aws:kms:region-ID:account-ID:alias/aws/s3",
        "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-
project"
    }
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的建立建置專案 (AWS CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateProject。

create-report-group

以下程式碼範例顯示如何使用 create-report-group。

AWS CLI

在 AWS CodeBuild 中建立報告群組。

下列create-report-group範例會建立新的報告群組。

```
aws codebuild create-report-group \
    --cli-input-json file://create-report-group-source.json
```

create-report-group-source.json 的內容：

```
{
    "name": "cli-created-report-group",
    "type": "TEST",
    "exportConfig": {
        "exportConfigType": "S3",
        "s3Destination": {
            "bucket": "amzn-s3-demo-bucket",
            "path": "",
            "packaging": "ZIP",
            "encryptionDisabled": true
        }
    }
}
```

輸出：

```
{
    "reportGroup": {
        "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/cli-created-
report-group",
        "name": "cli-created-report-group",
        "type": "TEST",
        "exportConfig": {
            "exportConfigType": "S3",
            "s3Destination": {
                "bucket": "amzn-s3-demo-bucket",
                "path": "",
                "packaging": "ZIP",
                "encryptionDisabled": true
            }
        },
        "created": 1602020026.775,
        "lastModified": 1602020026.775
    }
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的使用報告群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateReportGroup。

create-webhook

以下程式碼範例顯示如何使用 create-webhook。

AWS CLI

為 AWS CodeBuild 專案建立 Webhook 篩選條件

下列create-webhook範例會為名為 的 CodeBuild 專案建立 Webhookmy-project，該專案有兩個篩選群組。第一個篩選群組指定在分支上建立、更新或重新開啟的提取請求，並且這些分支的 Git 參考名稱符合規則表達式 ^refs/heads/master$，而標頭參考符合 ^refs/heads/myBranch$。第二個篩選條件群組指定分支上的推送請求，其 Git 參考名稱不符合規則表達式 ^refs/heads/myBranch$。

```
aws codebuild create-webhook \
    --project-name my-project \
```

```
    --filter-groups "[[{\"type\":\"EVENT\",\"pattern\":\"PULL_REQUEST_CREATED,
 PULL_REQUEST_UPDATED, PULL_REQUEST_REOPENED\"},{\"type\":\"HEAD_REF\",\"pattern
\":\"^refs/heads/myBranch$\",\"excludeMatchedPattern\":true},{\"type\":\"BASE_REF
\",\"pattern\":\"^refs/heads/master$\",\"excludeMatchedPattern\":true}],[{\"type\":
\"EVENT\",\"pattern\":\"PUSH\"},{\"type\":\"HEAD_REF\",\"pattern\":\"^refs/heads/
myBranch$\",\"excludeMatchedPattern\":true}]]"
```

輸出：

```
{
    "webhook": {
        "payloadUrl": "https://codebuild.us-west-2.amazonaws.com/webhooks?
t=eyJlbmNyeXB0ZWREYXRhIjoiVVl5MGtoeGRwSzZFRXl2Wnh4bld1Z0tKZ291TVpQNEtFamQ3RDlDYWpRaGIreVFrdm
        "url": "https://api.github.com/repos/iversonic/codedeploy-sample/
hooks/105190656",
        "lastModifiedSecret": 1556311319.069,
        "filterGroups": [
            [
                {
                    "type": "EVENT",
                    "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
 PULL_REQUEST_REOPENED",
                    "excludeMatchedPattern": false
                },
                {
                    "type": "HEAD_REF",
                    "pattern": "refs/heads/myBranch$",
                    "excludeMatchedPattern": true
                },
                {
                    "type": "BASE_REF",
                    "pattern": "refs/heads/master$",
                    "excludeMatchedPattern": true
                }
            ],
            [
                {
                    "type": "EVENT",
                    "pattern": "PUSH",
                    "excludeMatchedPattern": false
                },
                {
                    "type": "HEAD_REF",
```

```
                    "pattern": "refs/heads/myBranch$",
                    "excludeMatchedPattern": true
                }
            ]
        ]
    }
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的篩選 GitHub Webhook 事件 (SDK)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateWebhook。

delete-build-batch

以下程式碼範例顯示如何使用 delete-build-batch。

AWS CLI

在 AWS CodeBuild 中刪除批次建置。

下列delete-build-batch範例會刪除指定的批次建置。

```
aws codebuild delete-build-batch \
    --id <project-name>:<batch-ID>
```

輸出：

```
{
    "statusCode": "BATCH_DELETED",
    "buildsDeleted": [
        "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-name>:<build-
ID>",
        "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-name>:<build-
ID>",
        "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-name>:<build-
ID>",
        "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-name>:<build-ID>"
    ],
    "buildsNotDeleted": []
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的 Batch builds inCodeBuild。 AWS CodeBuild

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBuildBatch。

delete-project

以下程式碼範例顯示如何使用 delete-project。

AWS CLI

刪除 AWS CodeBuild 組建專案

下列delete-project範例會刪除指定的 CodeBuild 組建專案。

```
aws codebuild delete-project --name my-project
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的刪除組建專案 (AWS CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteProject。

delete-report-group

以下程式碼範例顯示如何使用 delete-report-group。

AWS CLI

在 AWS CodeBuild 中刪除報告群組。

下列delete-report-group範例會刪除具有指定 ARN 的報告群組。

```
aws codebuild delete-report-group \
    --arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-group-name>
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的使用報告群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteReportGroup。

delete-report

以下程式碼範例顯示如何使用 delete-report。

AWS CLI

在 AWS CodeBuild 中刪除報告。

下列delete-report範例會刪除指定的報告。

```
aws codebuild delete-report \
    --arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的使用報告。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteReport。

delete-source-credentials

以下程式碼範例顯示如何使用 delete-source-credentials。

AWS CLI

中斷與來源提供者的連線，並移除其存取權杖。

下列delete-source-credentials範例會中斷與來源提供者的連線，並移除其權杖。用於連線至來源提供者的來源登入資料的 ARN 會決定哪些來源登入資料。

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

輸出：

```
{
    "arn": "arn:aws:codebuild:your-region:your-account-id:token/your-server-type"
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的使用存取字符 (CLI) 連接來源提供者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSourceCredentials。

delete-webhook

以下程式碼範例顯示如何使用 delete-webhook。

AWS CLI

從 AWS CodeBuild 專案刪除 Webhook 篩選條件

下列delete-webhook範例會從指定的 CodeBuild 專案中刪除 Webhook。

```
aws codebuild delete-webhook --project-name my-project
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的停止自動執行組建 (AWS CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteWebhook。

describe-code-coverages

以下程式碼範例顯示如何使用 describe-code-coverages。

AWS CLI

若要取得 AWS CodeBuild 中程式碼涵蓋範圍測試結果的詳細資訊。

下列describe-code-coverages範例會取得指定報告中程式碼涵蓋範圍測試結果的相關資訊。

```
aws codebuild describe-code-coverages \
    --report-arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>
```

輸出：

```
{
    "codeCoverages": [
        {
            "id": "20a0adcc-db13-4b66-804b-ecaf9f852855",
            "reportARN": "arn:aws:codebuild:<region-ID>:972506530580:report/<report-group-name>:<report-ID>",
            "filePath": "<source-file-1-path>",
            "lineCoveragePercentage": 83.33,
            "linesCovered": 5,
            "linesMissed": 1,
            "branchCoveragePercentage": 50.0,
            "branchesCovered": 1,
            "branchesMissed": 1,
```

```
            "expired": "2020-11-20T21:22:45+00:00"
        },
        {
            "id": "0887162d-bf57-4cf1-a164-e432373d1a83",
            "reportARN": "arn:aws:codebuild:<region-ID>:972506530580:report/<report-
group-name>:<report-ID>",
            "filePath": "<source-file-2-path>",
            "lineCoveragePercentage": 90.9,
            "linesCovered": 10,
            "linesMissed": 1,
            "branchCoveragePercentage": 50.0,
            "branchesCovered": 1,
            "branchesMissed": 1,
            "expired": "2020-11-20T21:22:45+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的程式碼涵蓋範圍報告。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCodeCoverages。

describe-test-cases

以下程式碼範例顯示如何使用 `describe-test-cases`。

AWS CLI

取得 AWS CodeBuild 中測試案例的詳細資訊。

下列`describe-test-cases`範例會取得指定報告中測試案例的相關資訊。

```
aws codebuild describe-test-cases \
    --report-arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-
name>:<report-ID>
```

輸出：

```
{
    "testCases": [
        {
            "reportArn": "arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-
group-name>:<report-ID>",
```

```
            "testRawDataPath": "<test-report-path>",
            "prefix": "NUnit.Tests.Assemblies.MockTestFixture",
            "name": "NUnit.Tests.Assemblies.MockTestFixture.NotRunnableTest",
            "status": "ERROR",
            "durationInNanoSeconds": 0,
            "message": "No arguments were provided\n",
            "expired": "2020-11-20T17:52:10+00:00"
        },
        {
            "reportArn": "arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-
group-name>:<report-ID>",
            "testRawDataPath": "<test-report-path>",
            "prefix": "NUnit.Tests.Assemblies.MockTestFixture",
            "name": "NUnit.Tests.Assemblies.MockTestFixture.TestWithException",
            "status": "ERROR",
            "durationInNanoSeconds": 0,
            "message": "System.ApplicationException : Intentional Exception
\nat NUnit.Tests.Assemblies.MockTestFixture.MethodThrowsException()\nat
 NUnit.Tests.Assemblies.MockTestFixture.TestWithException()\n\n",
            "expired": "2020-11-20T17:52:10+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的使用測試報告 inCodeBuildAWS CodeBuild。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTestCases。

import-source-credentials

以下程式碼範例顯示如何使用 import-source-credentials。

AWS CLI

透過匯入來源提供者的登入資料，將 AWS CodeBuild 使用者連線至來源提供者。

下列import-source-credentials範例會為使用 BASIC_AUTH 進行身分驗證類型的 Bitbucket 儲存庫匯入字符。

```
aws codebuild import-source-credentials --server-type BITBUCKET --auth-
type BASIC_AUTH --token my-Bitbucket-password --username my-Bitbucket-username
```

輸出：

```
{
    "arn": "arn:aws:codebuild:us-west-2:123456789012:token/bitbucket"
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的使用存取字符 (CLI) 連接來源提供者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ImportSourceCredentials。

invalidate-project-cache

以下程式碼範例顯示如何使用 invalidate-project-cache。

AWS CLI

重設 an AWS CodeBuild 組建專案的快取。

下列invalidate-project-cache範例會重設指定 CodeBuild 專案的快取。

```
aws codebuild invalidate-project-cache --project-name my-project
```

此命令不會產生輸出。

如需詳細資訊，請參閱 CodeBuild 使用者指南中的在 CodeBuild 中建置快取。 AWS CodeBuild

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 InvalidateProjectCache。

list-build-batches-for-project

以下程式碼範例顯示如何使用 list-build-batches-for-project。

AWS CLI

列出 AWS CodeBuild 中特定組建專案的批次組建。

下列list-build-batches-for-project範例列出指定專案的 CodeBuild 批次建置。

```
aws codebuild list-build-batches-for-project \
    --project-name "<project-name>"
```

輸出：

```
{
    "ids": [
        "<project-name>:<batch-ID>",
        "<project-name>:<batch-ID>"
    ]
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的 Batch builds inCodeBuild。 AWS CodeBuild

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBuildBatchesForProject。

`list-build-batches`

以下程式碼範例顯示如何使用 `list-build-batches`。

AWS CLI

在 AWS CodeBuild 中列出批次組建。

下列`list-build-batches`範例列出目前帳戶的 CodeBuild 批次建置。

```
aws codebuild list-build-batches
```

輸出：

```
{
    "ids": [
        "<project-name>:<batch-ID>",
        "<project-name>:<batch-ID>"
    ]
}
```

如需詳細資訊，請參閱 AWS CodeBuild AWS 使用者指南中的 CodeBuild <https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html>)__ 中的批次建置。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBuildBatches。

`list-builds-for-project`

以下程式碼範例顯示如何使用 `list-builds-for-project`。

AWS CLI

檢視 an AWS CodeBuild 組建專案的組建清單。

下列`list-builds-for-project`範例會依遞減順序列出指定 CodeBuild 組建專案的組建 IDs。

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --sort-order DESCENDING
```

輸出：

```
{
    "ids": [
        "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-11111example",
        "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-22222example",
        "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-33333example",
        "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-44444example",
        "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-55555example"
    ]
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的檢視組建IDs 清單 (AWS CLI)

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBuildsForProject。

`list-builds`

以下程式碼範例顯示如何使用 `list-builds`。

AWS CLI

取得 AWS CodeBuild 組建 IDs的清單。

下列`list-builds`範例會取得依遞增順序排序的 CodeBuild IDs清單。

```
aws codebuild list-builds --sort-order ASCENDING
```

輸出包含一個`nextToken`值，指出有更多輸出可用。

```
{
    "nextToken": "4AEA6u7J...The full token has been omitted for
 brevity...MzY2OA==",
```

```
    "ids": [
        "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE"
        "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE"
            ... The full list of build IDs has been omitted for brevity ...
        "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
    ]
}
```

再次執行此命令，並提供上一個回應中的 nextToken值做為參數，以取得輸出的下一個部分。重複此動作，直到您未在回應中收到nextToken值為止。

```
aws codebuild list-builds --sort-order ASCENDING --next-
token 4AEA6u7J...The full token has been omitted for brevity...MzY2OA==
```

輸出的下一個部分：

```
{
    "ids": [
        "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
        "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
            ... The full list of build IDs has been omitted for brevity ...
        "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
    ]
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的檢視組建 IDs 清單 (AWS CLI)

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBuilds。

list-curated-environment-images

以下程式碼範例顯示如何使用 list-curated-environment-images。

AWS CLI

若要取得 AWS CodeBuild 管理的 Docker 映像清單，您可以用於建置。

下列list-curated-environment-images範例列出 CodeBuild 管理的 Docker 映像，可用於組建。

```
aws codebuild list-curated-environment-images
```

輸出：

```
{
    "platforms": [
        {
            "platform": "AMAZON_LINUX",
            "languages": [
                {
                    "language": "JAVA",
                    "images": [
                        {
                            "description": "AWS ElasticBeanstalk - Java 7 Running on
 Amazon Linux 64bit v2.1.3",
                            "name": "aws/codebuild/eb-java-7-amazonlinux-64:2.1.3",
                            "versions": [
                                "aws/codebuild/eb-java-7-amazonlinux-64:2.1.3-1.0.0"
                            ]
                        },
                        {
                            "description": "AWS ElasticBeanstalk - Java 8 Running on
 Amazon Linux 64bit v2.1.3",
                            "name": "aws/codebuild/eb-java-8-amazonlinux-64:2.1.3",
                            "versions": [
                                "aws/codebuild/eb-java-8-amazonlinux-64:2.1.3-1.0.0"
                            ]
                        },
                        ... LIST TRUNCATED FOR BREVITY ...
                    ]
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱 CodeBuild 使用者指南中的 CodeBuild 提供的 Docker 影像 AWS CodeBuild

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCuratedEnvironmentImages。

list-projects

以下程式碼範例顯示如何使用 list-projects。

AWS CLI

取得 AWS CodeBuild 組建專案名稱的清單。

下列`list-projects`範例會取得依名稱遞增排序的 CodeBuild 組建專案清單。

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

輸出包含一個`nextToken`值，指出有更多輸出可用。

```
{
    "nextToken": "Ci33ACF6...The full token has been omitted for brevity...U
+AkMx8=",
    "projects": [
        "codebuild-demo-project",
        "codebuild-demo-project2",
            ... The full list of build project names has been omitted for
 brevity ...
        "codebuild-demo-project99"
    ]
}
```

再次執行此命令，並提供先前回應`nextToken`的值做為參數，以取得輸出的下一個部分。重複此動作，直到您未在回應中收到`nextToken`值為止。

```
aws codebuild list-projects  --sort-by NAME --sort-order ASCENDING --next-
token Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=

{
    "projects": [
        "codebuild-demo-project100",
        "codebuild-demo-project101",

  ... The full list of build project names has been omitted for brevity ...
        "codebuild-demo-project122"

  ]
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的檢視組建專案名稱清單 (AWS CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListProjects。

list-report-groups

以下程式碼範例顯示如何使用 list-report-groups。

AWS CLI

取得 AWS CodeBuild 中的報告群組 ARNs 清單。

下列list-report-groups範例會擷取 區域中帳戶的報告群組 ARNs。

```
aws codebuild list-report-groups
```

輸出：

```
{
    "reportGroups": [
        "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-1",
        "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-2",
        "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-3"
    ]
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的使用報告群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListReportGroups。

list-reports-for-report-group

以下程式碼範例顯示如何使用 list-reports-for-report-group。

AWS CLI

若要取得報告群組 in AWS CodeBuild 中的報告清單。

下列list-report-for-report-groups範例會擷取 區域中帳戶在指定報告群組中的報告。

```
aws codebuild list-reports-for-report-group \
    --report-group-arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-group-name>
```

輸出：

```
{
```

```
    "reports": [
        "arn:aws:codebuild:<region-ID>:<user-ID>:report/report-1",
        "arn:aws:codebuild:<region-ID>:<user-ID>:report/report-2",
        "arn:aws:codebuild:<region-ID>:<user-ID>:report/report-3"
    ]
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的使用報告群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListReportsForReportGroup。

`list-reports`

以下程式碼範例顯示如何使用 `list-reports`。

AWS CLI

取得目前帳戶 in AWS CodeBuild 的報告清單。

下列`list-reports`範例會擷取目前帳戶的報告 ARNs。

```
aws codebuild list-reports
```

輸出：

```
{
    "reports": [
        "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report
 ID>",
        "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report
 ID>",
        "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report
 ID>"
    ]
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的使用報告。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListReports。

`list-shared-projects`

以下程式碼範例顯示如何使用 `list-shared-projects`。

AWS CLI

在 AWS CodeBuild 中列出共用專案。

下列list-shared-projects範例列出目前帳戶可用的 CodeBuild 共用專案。

```
aws codebuild list-shared-projects
```

輸出：

```
{
    "projects": [
        "arn:aws:codebuild:<region-ID>:<account-ID>:project/<shared-project-
name-1>",
        "arn:aws:codebuild:<region-ID>:<account-ID>:project/<shared-project-name-2>"
    ]
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的使用共用專案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSharedProjects。

list-shared-report-groups

以下程式碼範例顯示如何使用 list-shared-report-groups。

AWS CLI

若要取得 AWS CodeBuild 中共用報告群組 ARNs的清單。

下列list-shared-report-groups範例會擷取 區域中帳戶的報告群組 ARNs。

```
aws codebuild list-shared-report-groups
```

輸出：

```
{
    "reportGroups": [
        "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-1",
        "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-2",
```

```
        "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-3"
    ]
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的使用報告群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSharedReportGroups。

`list-source-credentials`

以下程式碼範例顯示如何使用 `list-source-credentials`。

AWS CLI

檢視 sourceCredentialsObjects 的清單

下列`list-source-credentials`範例列出 AWS 連線至一個 Bitbucket 帳戶和一個 GitHub 帳戶的 權杖。回應中的每個`sourceCredentialsInfos`物件都包含連線的來源登入資料資訊。

```
aws codebuild list-source-credentials
```

輸出：

```
{
    "sourceCredentialsInfos": [
        {
            "serverType": "BITBUCKET",
            "arn": "arn:aws:codebuild:us-west-2:123456789012:token/bitbucket",
            "authType": "BASIC_AUTH"
        },
        {
            "serverType": "GITHUB",
            "arn": "arn:aws:codebuild:us-west-2:123456789012:token/github",
            "authType": "OAUTH"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的使用存取字符 (CLI) 連接來源提供者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSourceCredentials。

retry-build-batch

以下程式碼範例顯示如何使用 retry-build-batch。

AWS CLI

在 AWS CodeBuild 中重試失敗的批次建置。

下列retry-build-batch範例會重新啟動指定的批次建置。

```
aws codebuild retry-build-batch \
    --id <project-name>:<batch-ID>
```

輸出：

```
{
    "buildBatch": {
        "id": "<project-name>:<batch-ID>",
        "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build-batch/<project-
name>:<batch-ID>",
        "startTime": "2020-10-21T17:26:23.099000+00:00",
        "currentPhase": "SUBMITTED",
        "buildBatchStatus": "IN_PROGRESS",
        "resolvedSourceVersion": "3a9e11cb419e8fff14b03883dc4e64f6155aaa7e",
        "projectName": "<project-name>",
        "phases": [
            {
                "phaseType": "SUBMITTED",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2020-10-21T17:26:23.099000+00:00",
                "endTime": "2020-10-21T17:26:23.457000+00:00",
                "durationInSeconds": 0
            },
            {
                "phaseType": "DOWNLOAD_BATCHSPEC",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2020-10-21T17:26:23.457000+00:00",
                "endTime": "2020-10-21T17:26:54.902000+00:00",
                "durationInSeconds": 31
            },
            {
                "phaseType": "IN_PROGRESS",
                "phaseStatus": "CLIENT_ERROR",
                "startTime": "2020-10-21T17:26:54.902000+00:00",
```

```
                "endTime": "2020-10-21T17:28:16.060000+00:00",
                "durationInSeconds": 81
            },
            {
                "phaseType": "FAILED",
                "phaseStatus": "RETRY",
                "startTime": "2020-10-21T17:28:16.060000+00:00",
                "endTime": "2020-10-21T17:29:39.709000+00:00",
                "durationInSeconds": 83
            },
            {
                "phaseType": "SUBMITTED",
                "startTime": "2020-10-21T17:29:39.709000+00:00"
            }
        ],
        "source": {
            "type": "GITHUB",
            "location": "https://github.com/strohm-a/<project-name>-graph.git",
            "gitCloneDepth": 1,
            "gitSubmodulesConfig": {
                "fetchSubmodules": false
            },
            "reportBuildStatus": false,
            "insecureSsl": false
        },
        "secondarySources": [],
        "secondarySourceVersions": [],
        "artifacts": {
            "location": ""
        },
        "secondaryArtifacts": [],
        "cache": {
            "type": "NO_CACHE"
        },
        "environment": {
            "type": "LINUX_CONTAINER",
            "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
            "computeType": "BUILD_GENERAL1_SMALL",
            "environmentVariables": [],
            "privilegedMode": false,
            "imagePullCredentialsType": "CODEBUILD"
        },
        "logConfig": {
            "cloudWatchLogs": {
```

```
                "status": "ENABLED"
            },
            "s3Logs": {
                "status": "DISABLED",
                "encryptionDisabled": false
            }
        },
        "buildTimeoutInMinutes": 60,
        "queuedTimeoutInMinutes": 480,
        "complete": false,
        "initiator": "<username>",
        "encryptionKey": "arn:aws:kms:<region-ID>:<account-ID>:alias/aws/s3",
        "buildBatchNumber": 4,
        "buildBatchConfig": {
            "serviceRole": "arn:aws:iam::<account-ID>:role/service-role/<project-
name>",
            "restrictions": {
                "maximumBuildsAllowed": 100
            },
            "timeoutInMins": 480
        },
        "buildGroups": [
            {
                "identifier": "DOWNLOAD_SOURCE",
                "ignoreFailure": false,
                "currentBuildSummary": {
                    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
                    "requestedOn": "2020-10-21T17:26:23.889000+00:00",
                    "buildStatus": "SUCCEEDED",
                    "primaryArtifact": {
                        "type": "no_artifacts",
                        "identifier": "DOWNLOAD_SOURCE"
                    },
                    "secondaryArtifacts": []
                }
            },
            {
                "identifier": "linux_small",
                "dependsOn": [],
                "ignoreFailure": false,
                "currentBuildSummary": {
                    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
```

```
                    "requestedOn": "2020-10-21T17:26:55.115000+00:00",
                    "buildStatus": "FAILED",
                    "primaryArtifact": {
                        "type": "no_artifacts",
                        "identifier": "linux_small"
                    },
                    "secondaryArtifacts": []
                }
            },
            {
                "identifier": "linux_medium",
                "dependsOn": [
                    "linux_small"
                ],
                "ignoreFailure": false,
                "currentBuildSummary": {
                    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
                    "requestedOn": "2020-10-21T17:26:54.594000+00:00",
                    "buildStatus": "STOPPED"
                }
            },
            {
                "identifier": "linux_large",
                "dependsOn": [
                    "linux_medium"
                ],
                "ignoreFailure": false,
                "currentBuildSummary": {
                    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
                    "requestedOn": "2020-10-21T17:26:54.701000+00:00",
                    "buildStatus": "STOPPED"
                }
            }
        ]
    }
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的 Batch builds inCodeBuild。 AWS CodeBuild

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RetryBuildBatch。

retry-build

以下程式碼範例顯示如何使用 retry-build。

AWS CLI

在 AWS CodeBuild 中重試失敗的組建。

下列retry-build範例會重新啟動指定的組建。

```
aws codebuild retry-build \
    --id <project-name>:<build-ID>
```

輸出：

```
{
    "build": {
        "id": "<project-name>:<build-ID>",
        "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-
name>:<build-ID>",
        "buildNumber": 9,
        "startTime": "2020-10-21T17:51:38.161000+00:00",
        "currentPhase": "QUEUED",
        "buildStatus": "IN_PROGRESS",
        "projectName": "<project-name>",
        "phases": [
            {
                "phaseType": "SUBMITTED",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2020-10-21T17:51:38.161000+00:00",
                "endTime": "2020-10-21T17:51:38.210000+00:00",
                "durationInSeconds": 0
            },
            {
                "phaseType": "QUEUED",
                "startTime": "2020-10-21T17:51:38.210000+00:00"
            }
        ],
        "source": {
            "type": "GITHUB",
            "location": "<GitHub-repo-URL>",
            "gitCloneDepth": 1,
            "gitSubmodulesConfig": {
                "fetchSubmodules": false
```

```
            },
            "reportBuildStatus": false,
            "insecureSsl": false
        },
        "secondarySources": [],
        "secondarySourceVersions": [],
        "artifacts": {
            "location": ""
        },
        "secondaryArtifacts": [],
        "cache": {
            "type": "NO_CACHE"
        },
        "environment": {
            "type": "LINUX_CONTAINER",
            "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
            "computeType": "BUILD_GENERAL1_SMALL",
            "environmentVariables": [],
            "privilegedMode": false,
            "imagePullCredentialsType": "CODEBUILD"
        },
        "serviceRole": "arn:aws:iam::<account-ID>:role/service-role/<service-role-
name>",
        "logs": {
            "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=<region-ID>#logEvent:group=null;stream=null",
            "cloudWatchLogsArn": "arn:aws:logs:<region-ID>:<account-ID>:log-
group:null:log-stream:null",
            "cloudWatchLogs": {
                "status": "ENABLED"
            },
            "s3Logs": {
                "status": "DISABLED",
                "encryptionDisabled": false
            }
        },
        "timeoutInMinutes": 60,
        "queuedTimeoutInMinutes": 480,
        "buildComplete": false,
        "initiator": "<username>",
        "encryptionKey": "arn:aws:kms:<region-ID>:<account-ID>:alias/aws/s3"
    }
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的 Batch builds inCodeBuild。 AWS CodeBuild

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RetryBuild。

start-build-batch

以下程式碼範例顯示如何使用 start-build-batch。

AWS CLI

在 AWS CodeBuild 中啟動批次建置。

下列start-build-batch範例會啟動指定專案的批次建置。

```
aws codebuild start-build-batch \
    --project-name <project-name>
```

輸出：

```
{
    "buildBatch": {
        "id": "<project-name>:<batch-ID>",
        "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build-batch/<project-
name>:<batch-ID>",
        "startTime": "2020-10-21T16:54:24.740000+00:00",
        "currentPhase": "SUBMITTED",
        "buildBatchStatus": "IN_PROGRESS",
        "projectName": "<project-name>",
        "source": {
            "type": "GITHUB",
            "location": "<GitHub-repo-URL>",
            "gitCloneDepth": 1,
            "gitSubmodulesConfig": {
                "fetchSubmodules": false
            },
            "reportBuildStatus": false,
            "insecureSsl": false
        },
        "secondarySources": [],
        "secondarySourceVersions": [],
        "artifacts": {
```

```
            "location": ""
        },
        "secondaryArtifacts": [],
        "cache": {
            "type": "NO_CACHE"
        },
        "environment": {
            "type": "LINUX_CONTAINER",
            "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
            "computeType": "BUILD_GENERAL1_SMALL",
            "environmentVariables": [],
            "privilegedMode": false,
            "imagePullCredentialsType": "CODEBUILD"
        },
        "logConfig": {
            "cloudWatchLogs": {
                "status": "ENABLED"
            },
            "s3Logs": {
                "status": "DISABLED",
                "encryptionDisabled": false
            }
        },
        "buildTimeoutInMinutes": 60,
        "queuedTimeoutInMinutes": 480,
        "complete": false,
        "initiator": "<username>",
        "encryptionKey": "arn:aws:kms:<region-ID>:<account-ID>:alias/aws/s3",
        "buildBatchNumber": 3,
        "buildBatchConfig": {
            "serviceRole": "arn:aws:iam::<account-ID>:role/service-role/<service-
role-name>",
            "restrictions": {
                "maximumBuildsAllowed": 100
            },
            "timeoutInMins": 480
        }
    }
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的 Batch builds inCodeBuild。 AWS CodeBuild

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartBuildBatch。

start-build

以下程式碼範例顯示如何使用 start-build。

AWS CLI

開始執行建置 an AWS CodeBuild 組建專案。

下列start-build範例會啟動指定 CodeBuild 專案的組建。建置會覆寫專案的設定，在建置逾時之前允許佇列的分鐘數，以及專案的成品設定。

```
aws codebuild start-build \
    --project-name "my-demo-project" \
    --queued-timeout-in-minutes-override 5 \
    --artifacts-override {"\"type\": \"S3\",\"location\": \"arn:aws:s3:::artifacts-
override\",\"overrideArtifactName\":true"}
```

輸出：

```
{
    "build": {
        "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role",
        "buildStatus": "IN_PROGRESS",
        "buildComplete": false,
        "projectName": "my-demo-project",
        "timeoutInMinutes": 60,
        "source": {
            "insecureSsl": false,
            "type": "S3",
            "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source.zip"
        },
        "queuedTimeoutInMinutes": 5,
        "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
        "currentPhase": "QUEUED",
        "startTime": 1556905683.568,
        "environment": {
            "computeType": "BUILD_GENERAL1_MEDIUM",
            "environmentVariables": [],
            "type": "LINUX_CONTAINER",
            "privilegedMode": false,
            "image": "aws/codebuild/standard:1.0",
```

```
            "imagePullCredentialsType": "CODEBUILD"
        },
        "phases": [
            {
                "phaseStatus": "SUCCEEDED",
                "startTime": 1556905683.568,
                "phaseType": "SUBMITTED",
                "durationInSeconds": 0,
                "endTime": 1556905684.524
            },
            {
                "startTime": 1556905684.524,
                "phaseType": "QUEUED"
            }
        ],
        "logs": {
            "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logEvent:group=null;stream=null"
        },
        "artifacts": {
            "encryptionDisabled": false,
            "location": "arn:aws:s3:::artifacts-override/my-demo-project",
            "overrideArtifactName": true
        },
        "cache": {
            "type": "NO_CACHE"
        },
        "id": "my-demo-project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE",
        "initiator": "my-aws-account-name",
        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-
project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE"
    }
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的執行組建 (AWS CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartBuild。

stop-build-batch

以下程式碼範例顯示如何使用 stop-build-batch。

AWS CLI

在 AWS CodeBuild 中停止進行中批次建置。

下列stop-build-batch範例會停止指定的批次建置。

```
aws codebuild stop-build-batch \
    --id <project-name>:<batch-ID>
```

輸出：

```
{
    "buildBatch": {
        "id": "<project-name>:<batch-ID>",
        "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build-batch/<project-
name>:<batch-ID>",
        "startTime": "2020-10-21T16:54:24.740000+00:00",
        "endTime": "2020-10-21T16:56:05.152000+00:00",
        "currentPhase": "STOPPED",
        "buildBatchStatus": "STOPPED",
        "resolvedSourceVersion": "aef7744ed069c51098e15c360f4102cd2cd1ad64",
        "projectName": "<project-name>",
        "phases": [
            {
                "phaseType": "SUBMITTED",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2020-10-21T16:54:24.740000+00:00",
                "endTime": "2020-10-21T16:54:25.039000+00:00",
                "durationInSeconds": 0
            },
            {
                "phaseType": "DOWNLOAD_BATCHSPEC",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2020-10-21T16:54:25.039000+00:00",
                "endTime": "2020-10-21T16:54:56.583000+00:00",
                "durationInSeconds": 31
            },
            {
                "phaseType": "IN_PROGRESS",
                "phaseStatus": "STOPPED",
                "startTime": "2020-10-21T16:54:56.583000+00:00",
                "endTime": "2020-10-21T16:56:05.152000+00:00",
                "durationInSeconds": 68
```

```
            },
            {
                "phaseType": "STOPPED",
                "startTime": "2020-10-21T16:56:05.152000+00:00"
            }
        ],
        "source": {
            "type": "GITHUB",
            "location": "<GitHub-repo-URL>",
            "gitCloneDepth": 1,
            "gitSubmodulesConfig": {
                "fetchSubmodules": false
            },
            "reportBuildStatus": false,
            "insecureSsl": false
        },
        "secondarySources": [],
        "secondarySourceVersions": [],
        "artifacts": {
            "location": ""
        },
        "secondaryArtifacts": [],
        "cache": {
            "type": "NO_CACHE"
        },
        "environment": {
            "type": "LINUX_CONTAINER",
            "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
            "computeType": "BUILD_GENERAL1_SMALL",
            "environmentVariables": [],
            "privilegedMode": false,
            "imagePullCredentialsType": "CODEBUILD"
        },
        "logConfig": {
            "cloudWatchLogs": {
                "status": "ENABLED"
            },
            "s3Logs": {
                "status": "DISABLED",
                "encryptionDisabled": false
            }
        },
        "buildTimeoutInMinutes": 60,
        "queuedTimeoutInMinutes": 480,
```

```
        "complete": true,
        "initiator": "Strohm",
        "encryptionKey": "arn:aws:kms:<region-ID>:<account-ID>:alias/aws/s3",
        "buildBatchNumber": 3,
        "buildBatchConfig": {
            "serviceRole": "arn:aws:iam::<account-ID>:role/service-role/<project-
name>",
            "restrictions": {
                "maximumBuildsAllowed": 100
            },
            "timeoutInMins": 480
        },
        "buildGroups": [
            {
                "identifier": "DOWNLOAD_SOURCE",
                "ignoreFailure": false,
                "currentBuildSummary": {
                    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
                    "requestedOn": "2020-10-21T16:54:25.468000+00:00",
                    "buildStatus": "SUCCEEDED",
                    "primaryArtifact": {
                        "type": "no_artifacts",
                        "identifier": "DOWNLOAD_SOURCE"
                    },
                    "secondaryArtifacts": []
                }
            },
            {
                "identifier": "linux_small",
                "dependsOn": [],
                "ignoreFailure": false,
                "currentBuildSummary": {
                    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
                    "requestedOn": "2020-10-21T16:54:56.833000+00:00",
                    "buildStatus": "IN_PROGRESS"
                }
            },
            {
                "identifier": "linux_medium",
                "dependsOn": [
                    "linux_small"
                ],
```

```
                "ignoreFailure": false,
                "currentBuildSummary": {
                    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
                    "requestedOn": "2020-10-21T16:54:56.211000+00:00",
                    "buildStatus": "PENDING"
                }
            },
            {
                "identifier": "linux_large",
                "dependsOn": [
                    "linux_medium"
                ],
                "ignoreFailure": false,
                "currentBuildSummary": {
                    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
                    "requestedOn": "2020-10-21T16:54:56.330000+00:00",
                    "buildStatus": "PENDING"
                }
            }
        ]
    }
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的 Batch builds inCodeBuild。 AWS CodeBuild

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopBuildBatch。

stop-build

以下程式碼範例顯示如何使用 stop-build。

AWS CLI

停止 AWS CodeBuild 組建專案的組建。

下列stop-build範例會停止指定的 CodeBuild 組建。

```
aws codebuild stop-build --id my-demo-project:12345678-a1b2-c3d4-e5f6-11111EXAMPLE
```

輸出：

```
{
    "build": {
        "startTime": 1556906956.318,
        "initiator": "my-aws-account-name",
        "projectName": "my-demo-project",
        "currentPhase": "COMPLETED",
        "cache": {
            "type": "NO_CACHE"
        },
        "source": {
            "insecureSsl": false,
            "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source.zip",
            "type": "S3"
        },
        "id": "my-demo-project:1a2b3c4d-5678-90ab-cdef-11111EXAMPLE",
        "endTime": 1556906974.781,
        "phases": [
            {
                "durationInSeconds": 0,
                "phaseType": "SUBMITTED",
                "endTime": 1556906956.935,
                "phaseStatus": "SUCCEEDED",
                "startTime": 1556906956.318
            },
            {
                "durationInSeconds": 1,
                "phaseType": "QUEUED",
                "endTime": 1556906958.272,
                "phaseStatus": "SUCCEEDED",
                "startTime": 1556906956.935
            },
            {
                "phaseType": "PROVISIONING",
                "phaseStatus": "SUCCEEDED",
                "durationInSeconds": 14,
                "contexts": [
                    {
                        "message": "",
                        "statusCode": ""
                    }
                ],
                "endTime": 1556906972.847,
```

```
            "startTime": 1556906958.272
        },
        {
            "phaseType": "DOWNLOAD_SOURCE",
            "phaseStatus": "SUCCEEDED",
            "durationInSeconds": 0,
            "contexts": [
                    {
                    "message": "",
                    "statusCode": ""
                }
            ],
            "endTime": 1556906973.552,
            "startTime": 1556906972.847
        },
        {
            "phaseType": "INSTALL",
            "phaseStatus": "SUCCEEDED",
            "durationInSeconds": 0,
            "contexts": [
                {
                    "message": "",
                    "statusCode": ""
                }
            ],
            "endTime": 1556906973.75,
            "startTime": 1556906973.552
        },
        {
            "phaseType": "PRE_BUILD",
            "phaseStatus": "SUCCEEDED",
            "durationInSeconds": 0,
            "contexts": [
                {
                    "message": "",
                    "statusCode": ""
                }
            ],
            "endTime": 1556906973.937,
            "startTime": 1556906973.75
        },
        {
            "durationInSeconds": 0,
            "phaseType": "BUILD",
```

```
                "endTime": 1556906974.781,
                "phaseStatus": "STOPPED",
                "startTime": 1556906973.937
            },
            {
                "phaseType": "COMPLETED",
                "startTime": 1556906974.781
            }
        ],
        "artifacts": {
            "location": "arn:aws:s3:::artifacts-override/my-demo-project",
            "encryptionDisabled": false,
            "overrideArtifactName": true
        },
        "buildComplete": true,
        "buildStatus": "STOPPED",
        "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
        "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role",
        "queuedTimeoutInMinutes": 5,
        "timeoutInMinutes": 60,
        "environment": {
            "type": "LINUX_CONTAINER",
            "environmentVariables": [],
            "computeType": "BUILD_GENERAL1_MEDIUM",
            "privilegedMode": false,
            "image": "aws/codebuild/standard:1.0",
            "imagePullCredentialsType": "CODEBUILD"
        },
        "logs": {
            "streamName": "1a2b3c4d-5678-90ab-cdef-11111EXAMPLE",
            "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logEvent:group=/aws/codebuild/my-demo-project;stream=1a2b3c4d-5678-90ab-
cdef-11111EXAMPLE",
            "groupName": "/aws/codebuild/my-demo-project"
        },
        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-
project:1a2b3c4d-5678-90ab-cdef-11111EXAMPLE"
    }
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的停止建置 (AWS CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopBuild。

update-project

以下程式碼範例顯示如何使用 update-project。

AWS CLI

變更 AWS CodeBuild 組建專案的設定。

下列update-project範例會變更名為 my-demo-project 之指定 CodeBuild 組建專案的設定。

```
aws codebuild update-project --name "my-demo-project" \
    --description "This project is updated" \
    --source "{\"type\": \"S3\",\"location\": \"codebuild-us-west-2-123456789012-input-bucket/my-source-2.zip\"}" \
    --artifacts {"\"type\": \"S3\",\"location\": \"codebuild-us-west-2-123456789012-output-bucket-2\""} \
    --environment "{\"type\": \"LINUX_CONTAINER\",\"image\": \"aws/codebuild/standard:1.0\",\"computeType\": \"BUILD_GENERAL1_MEDIUM\"}" \
    --service-role "arn:aws:iam::123456789012:role/service-role/my-codebuild-service-role"
```

輸出會顯示更新的設定。

```
{
    "project": {
        "arn": "arn:aws:codebuild:us-west-2:123456789012:project/my-demo-project",
        "environment": {
            "privilegedMode": false,
            "environmentVariables": [],
            "type": "LINUX_CONTAINER",
            "image": "aws/codebuild/standard:1.0",
            "computeType": "BUILD_GENERAL1_MEDIUM",
            "imagePullCredentialsType": "CODEBUILD"
        },
        "queuedTimeoutInMinutes": 480,
        "description": "This project is updated",
        "artifacts": {
            "packaging": "NONE",
            "name": "my-demo-project",
            "type": "S3",
            "namespaceType": "NONE",
            "encryptionDisabled": false,
            "location": "codebuild-us-west-2-123456789012-output-bucket-2"
```

```
        },
        "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
        "badge": {
            "badgeEnabled": false
        },
        "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role",
        "lastModified": 1556840545.967,
        "tags": [],
        "timeoutInMinutes": 60,
        "created": 1556839783.274,
        "name": "my-demo-project",
        "cache": {
            "type": "NO_CACHE"
        },
        "source": {
            "type": "S3",
            "insecureSsl": false,
            "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source-2.zip"
        }
    }
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的變更建置專案的設定 (AWS CLI)

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateProject。

update-report-group

以下程式碼範例顯示如何使用 update-report-group。

AWS CLI

在 AWS CodeBuild 中更新報告群組。

下列update-report-group範例會將報告群組的匯出類型變更為 "NO_EXPORT"。

```
aws codebuild update-report-group \
    --arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/cli-created-report-
group \
    --export-config="exportConfigType=NO_EXPORT"
```

輸出：

```
{
    "reportGroup": {
        "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/cli-created-
report-group",
        "name": "cli-created-report-group",
        "type": "TEST",
        "exportConfig": {
            "exportConfigType": "NO_EXPORT"
        },
        "created": 1602020686.009,
        "lastModified": 1602021033.454,
        "tags": []
    }
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的使用報告群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateReportGroup。

update-webhook

以下程式碼範例顯示如何使用 update-webhook。

AWS CLI

更新 an AWS CodeBuild 專案的 Webhook

下列update-webhook範例會使用兩個篩選群組更新指定 CodeBuild 專案的 Webhook。--rotate-secret 參數指定每次程式碼變更觸發組建時，GitHub 都會輪換專案的私密金鑰。第一個篩選群組指定在分支上建立、更新或重新開啟的提取請求，並且這些分支的 Git 參考名稱符合規則表達式 ^refs/heads/master$，而標頭參考符合 ^refs/heads/myBranch$。第二個篩選條件群組指定分支上的推送請求，其 Git 參考名稱不符合規則表達式 ^refs/heads/myBranch$。

```
aws codebuild update-webhook \
    --project-name Project2 \
    --rotate-secret \
    --filter-groups "[[{\"type\":\"EVENT\",\"pattern\":\"PULL_REQUEST_CREATED,
 PULL_REQUEST_UPDATED, PULL_REQUEST_REOPENED\"},{\"type\":\"HEAD_REF\",\"pattern
\":\"^refs/heads/myBranch$\",\"excludeMatchedPattern\":true},{\"type\":\"BASE_REF
\",\"pattern\":\"^refs/heads/master$\",\"excludeMatchedPattern\":true}],[{\"type\":
```

```
\"EVENT\",\"pattern\":\"PUSH\"},{\"type\":\"HEAD_REF\",\"pattern\":\"^refs/heads/
myBranch$\",\"excludeMatchedPattern\":true}]]"
```

輸出：

```
{
    "webhook": {
        "filterGroups": [
            [
                {
                    "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
 PULL_REQUEST_REOPENED",
                    "type": "EVENT"
                },
                {
                    "excludeMatchedPattern": true,
                    "pattern": "refs/heads/myBranch$",
                    "type": "HEAD_REF"
                },
                {
                    "excludeMatchedPattern": true,
                    "pattern": "refs/heads/master$",
                    "type": "BASE_REF"
                }
            ],
            [
                {
                    "pattern": "PUSH",
                    "type": "EVENT"
                },
                {
                    "excludeMatchedPattern": true,
                    "pattern": "refs/heads/myBranch$",
                    "type": "HEAD_REF"
                }
            ]
        ],
        "lastModifiedSecret": 1556312220.133
    }
}
```

如需詳細資訊，請參閱 AWS CodeBuild 使用者指南中的變更組建專案的設定 (AWS CLI)

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateWebhook。

使用 的 CodeCommit 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 CodeCommit 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-approval-rule-template-with-repository`

以下程式碼範例顯示如何使用 `associate-approval-rule-template-with-repository`。

AWS CLI

將核准規則範本與儲存庫建立關聯

下列`associate-approval-rule-template-with-repository`範例會將指定的核准規則範本與名為 的儲存庫建立關聯`MyDemoRepo`。

```
aws codecommit associate-approval-rule-template-with-repository \
    --repository-name MyDemoRepo  \
    --approval-rule-template-name 2-approver-rule-for-main
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的將核准規則範本與儲存庫建立關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateApprovalRuleTemplateWithRepository。

`batch-associate-approval-rule-template-with-repositories`

以下程式碼範例顯示如何使用 `batch-associate-approval-rule-template-with-repositories`。

AWS CLI

在單一操作中將核准規則範本與多個儲存庫建立關聯

下列batch-associate-approval-rule-template-with-repositories範例會將指定的核准規則範本與名為 MyDemoRepo和 的儲存庫建立關聯MyOtherDemoRepo。

注意：核准規則範本專屬於建立規則的 AWS 區域。它們只能與該 AWS 區域中的儲存庫相關聯。

```
aws codecommit batch-associate-approval-rule-template-with-repositories \
    --repository-names MyDemoRepo, MyOtherDemoRepo  \
    --approval-rule-template-name 2-approver-rule-for-main
```

輸出：

```
{
    "associatedRepositoryNames": [
        "MyDemoRepo",
        "MyOtherDemoRepo"
    ],
    "errors": []
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的將核准規則範本與儲存庫建立關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchAssociateApprovalRuleTemplateWithRepositories。

batch-describe-merge-conflicts

以下程式碼範例顯示如何使用 batch-describe-merge-conflicts。

AWS CLI

取得兩個遞交指標之間合併中所有檔案或檔案子集之合併衝突的相關資訊

下列batch-describe-merge-conflicts範例決定合併名為 的來源分支feature-randomizationfeature與main使用名為 之儲存庫中的THREE_WAY_MERGE策略名為 的目的地分支的合併衝突MyDemoRepo。

```
aws codecommit batch-describe-merge-conflicts \
```

```
    --source-commit-specifier feature-randomizationfeature \
    --destination-commit-specifier main \
    --merge-option THREE_WAY_MERGE \
    --repository-name MyDemoRepo
```

輸出：

```
{
    "conflicts": [
        {
            "conflictMetadata": {
                "filePath": "readme.md",
                "fileSizes": {
                    "source": 139,
                    "destination": 230,
                    "base": 85
                },
                "fileModes": {
                    "source": "NORMAL",
                    "destination": "NORMAL",
                    "base": "NORMAL"
                },
                "objectTypes": {
                    "source": "FILE",
                    "destination": "FILE",
                    "base": "FILE"
                },
                "numberOfConflicts": 1,
                "isBinaryFile": {
                    "source": false,
                    "destination": false,
                    "base": false
                },
                "contentConflict": true,
                "fileModeConflict": false,
                "objectTypeConflict": false,
                "mergeOperations": {
                    "source": "M",
                    "destination": "M"
                }
            },
            "mergeHunks": [
                {
```

```
                    "isConflict": true,
                    "source": {
                        "startLine": 0,
                        "endLine": 3,
                        "hunkContent": "VGhpcyBpEXAMPLE=="
                    },
                    "destination": {
                        "startLine": 0,
                        "endLine": 1,
                        "hunkContent": "VXNlIHRoEXAMPLE="
                    }
                }
            ]
        }
    ],
    "errors": [],
    "destinationCommitId": "86958e0aEXAMPLE",
    "sourceCommitId": "6ccd57fdEXAMPLE",
    "baseCommitId": "767b6958EXAMPLE"
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的解決提取請求中的衝突。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDescribeMergeConflicts。

batch-disassociate-approval-rule-template-from-repositories

以下程式碼範例顯示如何使用 batch-disassociate-approval-rule-template-from-repositories。

AWS CLI

在單一操作中取消核准規則範本與多個儲存庫的關聯

下列batch-disassociate-approval-rule-template-from-repositories範例會將指定的核准規則範本與名為 MyDemoRepo和 的儲存庫取消關聯MyOtherDemoRepo。

```
aws codecommit batch-disassociate-approval-rule-template-from-repositories \
    --repository-names MyDemoRepo, MyOtherDemoRepo  \
    --approval-rule-template-name 1-approval-rule-for-all pull requests
```

輸出：

```
{
    "disassociatedRepositoryNames": [
        "MyDemoRepo",
        "MyOtherDemoRepo"
    ],
    "errors": []
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的取消與核准規則範本的關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDisassociateApprovalRuleTemplateFromRepositories。

batch-get-commits

以下程式碼範例顯示如何使用 batch-get-commits。

AWS CLI

檢視多個遞交的相關資訊

下列batch-get-commits範例顯示指定遞交的詳細資訊。

```
aws codecommit batch-get-commits  \
    --repository-name MyDemoRepo  \
    --commit-ids 317f8570EXAMPLE 4c925148EXAMPLE
```

輸出：

```
{
    "commits": [
      {
        "additionalData": "",
        "committer": {
            "date": "1508280564 -0800",
            "name": "Mary Major",
            "email": "mary_major@example.com"
        },
        "author": {
            "date": "1508280564 -0800",
            "name": "Mary Major",
```

```
            "email": "mary_major@example.com"
        },
        "commitId": "317f8570EXAMPLE",
        "treeId": "1f330709EXAMPLE",
        "parents": [
            "6e147360EXAMPLE"
        ],
        "message": "Change variable name and add new response element"
    },
    {
        "additionalData": "",
        "committer": {
            "date": "1508280542 -0800",
            "name": "Li Juan",
            "email": "li_juan@example.com"
        },
        "author": {
            "date": "1508280542 -0800",
            "name": "Li Juan",
            "email": "li_juan@example.com"
        },
        "commitId": "4c925148EXAMPLE",
        "treeId": "1f330709EXAMPLE",
        "parents": [
            "317f8570EXAMPLE"
        ],
        "message": "Added new class"
    }
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的檢視遞交詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetCommits。

batch-get-repositories

以下程式碼範例顯示如何使用 batch-get-repositories。

AWS CLI

檢視多個儲存庫的詳細資訊

此範例顯示多個 AWS CodeCommit 儲存庫的詳細資訊。

```
aws codecommit batch-get-repositories \
    --repository-names MyDemoRepo MyOtherDemoRepo
```

輸出：

```
{
    "repositoriesNotFound": [],
    "repositories": [
         {
            "creationDate": 1429203623.625,
            "defaultBranch": "main",
            "repositoryName": "MyDemoRepo",
            "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/
MyDemoRepo",
            "lastModifiedDate": 1430783812.0869999,
            "repositoryDescription": "My demonstration repository",
            "cloneUrlHttp": "https://codecommit.us-east-2.amazonaws.com/v1/repos/
MyDemoRepo",
            "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
            "Arn": "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo"
            "accountId": "111111111111"
        },
        {
            "creationDate": 1429203623.627,
            "defaultBranch": "main",
            "repositoryName": "MyOtherDemoRepo",
            "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/
MyOtherDemoRepo",
            "lastModifiedDate": 1430783812.0889999,
            "repositoryDescription": "My other demonstration repository",
            "cloneUrlHttp": "https://codecommit.us-east-2.amazonaws.com/v1/repos/
MyOtherDemoRepo",
            "repositoryId": "cfc29ac4-b0cb-44dc-9990-f6f51EXAMPLE",
            "Arn": "arn:aws:codecommit:us-east-2:111111111111:MyOtherDemoRepo"
            "accountId": "111111111111"
        }
    ],
    "repositoriesNotFound": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetRepositories。

create-approval-rule-template

以下程式碼範例顯示如何使用 create-approval-rule-template。

AWS CLI

建立核准規則範本

下列create-approval-rule-template範例會建立名為 的核准規則範本2-approver-rule-for-main ``. The template requires two users who assume the role of ``CodeCommitReview，以核准任何提取請求，然後才能合併到main分支。

```
aws codecommit create-approval-rule-template \
    --approval-rule-template-name 2-approver-rule-for-main \
    --approval-rule-template-description  "Requires two developers from the team to
 approve the pull request if the destination branch is main" \
    --approval-rule-template-content "{\"Version\": \"2018-11-08\",
\"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type
\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}"
```

輸出：

```
{
    "approvalRuleTemplate": {
        "approvalRuleTemplateName": "2-approver-rule-for-main",
        "creationDate": 1571356106.936,
        "approvalRuleTemplateId": "dd8b17fe-EXAMPLE",
        "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\",
\"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type
\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "approvalRuleTemplateDescription": "Requires two developers from the team to
 approve the pull request if the destination branch is main",
        "lastModifiedDate": 1571356106.936,
        "ruleContentSha256": "4711b576EXAMPLE"
    }
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的建立核准規則範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateApprovalRuleTemplate。

create-branch

以下程式碼範例顯示如何使用 create-branch。

AWS CLI

建立分支

此範例會在 AWS CodeCommit 儲存庫中建立分支。只有在發生錯誤時，此命令才會產生輸出。

命令：

```
aws codecommit create-branch --repository-name MyDemoRepo --branch-name MyNewBranch
 --commit-id 317f8570EXAMPLE
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateBranch。

create-commit

以下程式碼範例顯示如何使用 create-commit。

AWS CLI

建立遞交

下列create-commit範例示範如何為將readme.md檔案新增至main分支MyDemoRepo中名為 的儲存庫的儲存庫建立初始遞交。

```
aws codecommit create-commit \
    --repository-name MyDemoRepo \
    --branch-name main \
    --put-files "filePath=readme.md,fileContent='Welcome to our team repository.'"
```

輸出：

```
{
    "filesAdded": [
        {
```

```
            "blobId": "5e1c309d-EXAMPLE",
            "absolutePath": "readme.md",
            "fileMode": "NORMAL"
        }
    ],
    "commitId": "4df8b524-EXAMPLE",
    "treeId": "55b57003-EXAMPLE",
    "filesDeleted": [],
    "filesUpdated": []
}
```

如需詳細資訊，請參閱 CodeCommit 使用者指南中的在 AWS CodeCommit 中建立遞交。 AWS CodeCommit

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCommit。

create-pull-request-approval-rule

以下程式碼範例顯示如何使用 create-pull-request-approval-rule。

AWS CLI

為提取請求建立核准規則

下列create-pull-request-approval-rule範例Require two approved approvers會為指定的提取請求建立名為 的核准規則。規則指定需要兩個核准集區中的核准。集區包含透過CodeCommitReview在123456789012 AWS 帳戶中擔任 角色來存取 CodeCommit 的所有使用者。它還包括Nikhil_Jayashankar來自相同 AWS 帳戶的 IAM 使用者或名為 的聯合身分使用者。

```
aws codecommit create-pull-request-approval-rule  \
    --approval-rule-name "Require two approved approvers"  \
    --approval-rule-content "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers
\": [\"CodeCommitApprovers:123456789012:Nikhil_Jayashankar\",
 \"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}"
```

輸出：

```
{
    "approvalRule": {
        "approvalRuleName": "Require two approved approvers",
```

```
        "lastModifiedDate": 1570752871.932,
        "ruleContentSha256": "7c44e6ebEXAMPLE",
        "creationDate": 1570752871.932,
        "approvalRuleId": "aac33506-EXAMPLE",
        "approvalRuleContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers
\": [\"CodeCommitApprovers:123456789012:Nikhil_Jayashankar\",
 \"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major"
    }
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的建立核准規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePullRequestApprovalRule。

`create-pull-request`

以下程式碼範例顯示如何使用 `create-pull-request`。

AWS CLI

建立提取請求

下列create-pull-request範例會建立名為「發音困難分析器」的提取請求，其描述為「請檢閱週二的這些變更，該變更以「jane-branch」來源分支為目標，並將合併到名為「MyDemoRepo」的 AWS CodeCommit 儲存庫中的預設分支「主要」。

```
aws codecommit create-pull-request \
    --title "My Pull Request" \
    --description "Please review these changes by Tuesday" \
    --client-request-token 123Example \
    --targets repositoryName=MyDemoRepo,sourceReference=MyNewBranch
```

輸出：

```
{
    "pullRequest": {
        "approvalRules": [
            {
                "approvalRuleContent": "{\"Version\": \"2018-11-08\",
\"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type
```

```
\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
                "approvalRuleId": "dd8b17fe-EXAMPLE",
                "approvalRuleName": "2-approver-rule-for-main",
                "creationDate": 1571356106.936,
                "lastModifiedDate": 571356106.936,
                "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
                "originApprovalRuleTemplate": {
                    "approvalRuleTemplateId": "dd3d22fe-EXAMPLE",
                    "approvalRuleTemplateName": "2-approver-rule-for-main"
                },
                "ruleContentSha256": "4711b576EXAMPLE"
            }
        ],
        "authorArn": "arn:aws:iam::111111111111:user/Jane_Doe",
        "description": "Please review these changes by Tuesday",
        "title": "Pronunciation difficulty analyzer",
        "pullRequestTargets": [
            {
                "destinationCommit": "5d036259EXAMPLE",
                "destinationReference": "refs/heads/main",
                "repositoryName": "MyDemoRepo",
                "sourceCommit": "317f8570EXAMPLE",
                "sourceReference": "refs/heads/jane-branch",
                "mergeMetadata": {
                    "isMerged": false
                }
            }
        ],
        "lastActivityDate": 1508962823.285,
        "pullRequestId": "42",
        "clientRequestToken": "123Example",
        "pullRequestStatus": "OPEN",
        "creationDate": 1508962823.285
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePullRequest。

create-repository

以下程式碼範例顯示如何使用 create-repository。

AWS CLI

建立儲存庫

此範例會建立儲存庫，並將其與使用者帳戶建立關聯 AWS 。

命令：

```
aws codecommit create-repository --repository-name MyDemoRepo --repository-
description "My demonstration repository"
```

輸出：

```
{
    "repositoryMetadata": {
        "repositoryName": "MyDemoRepo",
                "cloneUrlSsh": "ssh://git-codecommit.us-east-1.amazonaws.com/v1/
repos/MyDemoRepo",
                "lastModifiedDate": 1444766838.027,
        "repositoryDescription": "My demonstration repository",
                "cloneUrlHttp": "https://git-codecommit.us-east-1.amazonaws.com/v1/
repos/MyDemoRepo",
        "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
                "Arn": "arn:aws:codecommit:us-
east-1:111111111111EXAMPLE:MyDemoRepo",
        "accountId": "111111111111"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRepository。

create-unreferenced-merge-commit

以下程式碼範例顯示如何使用 create-unreferenced-merge-commit。

AWS CLI

建立未參考的遞交，代表合併兩個遞交指標的結果

下列create-unreferenced-merge-commit範例會建立遞交，代表名為 的來源分支bugfix-1234與名為 的目的地分支之間的合併結果，該分支main使用名為 的儲存庫中的 THREE_WAY_MERGE 策略MyDemoRepo。

```
aws codecommit create-unreferenced-merge-commit \
    --source-commit-specifier bugfix-1234 \
    --destination-commit-specifier main \
    --merge-option THREE_WAY_MERGE \
    --repository-name MyDemoRepo \
    --name "Maria Garcia" \
    --email "maria_garcia@example.com" \
    --commit-message "Testing the results of this merge."
```

輸出：

```
{
    "commitId": "4f178133EXAMPLE",
    "treeId": "389765daEXAMPLE"
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的解決提取請求中的衝突。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateUnreferencedMergeCommit。

credential-helper

以下程式碼範例顯示如何使用 credential-helper。

AWS CLI

使用 AWS CodeCommit 設定 CLI AWS 中包含的登入資料協助程式

credential-helper 公用程式的設計並非直接從 CLI AWS 呼叫。反之，它旨在搭配 git config命令做為參數使用，以設定您的本機電腦。每當 Git 需要向 驗證 AWS 才能與 CodeCommit 儲存庫互動時，它可讓 Git 使用 HTTPS 和密碼編譯簽署的 IAM 使用者憑證或 Amazon EC2 執行個體角色版本。

```
git config --global credential.helper '!aws codecommit credential-helper $@'
git config --global credential.UseHttpPath true
```

輸出：

```
[credential]
    helper = !aws codecommit credential-helper $@
    UseHttpPath = true
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的AWS 使用其他方法設定 CodeCommit。仔細檢閱內容，然後遵循下列其中一個主題中的程序：適用於 Linux、macOS 或 Unix 上的 HTTPS 連線，或 AWS CodeCommit 使用者指南中的適用於 Windows 上的 HTTPS 連線。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CredentialHelper。

delete-approval-rule-template

以下程式碼範例顯示如何使用 delete-approval-rule-template。

AWS CLI

刪除核准規則範本

下列delete-approval-rule-template範例會刪除指定的核准規則範本。

```
aws codecommit delete-approval-rule-template  \
    --approval-rule-template-name 1-approver-for-all-pull-requests
```

輸出：

```
{
    "approvalRuleTemplateId": "41de97b7-EXAMPLE"
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的刪除核准規則範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteApprovalRuleTemplate。

delete-branch

以下程式碼範例顯示如何使用 delete-branch。

AWS CLI

刪除分支

此範例說明如何刪除 AWS CodeCommit 儲存庫中的分支。

命令：

```
aws codecommit delete-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

輸出：

```
{
  "branch": {
      "commitId": "317f8570EXAMPLE",
      "branchName": "MyNewBranch"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBranch。

delete-comment-content

以下程式碼範例顯示如何使用 delete-comment-content。

AWS CLI

刪除評論的內容

您只能刪除您所建立評論的評論內容。此範例示範如何使用系統產生的 ID 刪除評論的內容ff30b348EXAMPLEb9aa670f。

```
aws codecommit delete-comment-content \
    --comment-id ff30b348EXAMPLEb9aa670f
```

輸出：

```
{
    "comment": {
        "creationDate": 1508369768.142,
        "deleted": true,
        "lastModifiedDate": 1508369842.278,
        "clientRequestToken": "123Example",
        "commentId": "ff30b348EXAMPLEb9aa670f",
        "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
        "callerReactions": [],
        "reactionCounts":
        {
```

```
            "CLAP" : 1
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCommentContent。

delete-file

以下程式碼範例顯示如何使用 delete-file。

AWS CLI

刪除檔案

下列delete-file範例示範如何README.md從名為 的分支中刪除名為 的檔案，main該分支在名為 的儲存庫c5709475EXAMPLE中具有 的最新遞交 IDMyDemoRepo。

```
aws codecommit delete-file \
    --repository-name MyDemoRepo \
    --branch-name main \
    --file-path README.md \
    --parent-commit-id c5709475EXAMPLE
```

輸出：

```
{
    "blobId":"559b44fEXAMPLE",
    "commitId":"353cf655EXAMPLE",
    "filePath":"README.md",
    "treeId":"6bc824cEXAMPLE"
}
```

如需詳細資訊，請參閱 AWS CodeCommit API 參考指南中的編輯或刪除檔案 inCodeCommitAWS CodeCommit。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFile。

delete-pull-request-approval-rule

以下程式碼範例顯示如何使用 delete-pull-request-approval-rule。

AWS CLI

刪除提取請求的核准規則

下列delete-pull-request-approval-rule範例會刪除指定提取請求My Approval Rule名為 的核准規則。

```
aws codecommit delete-pull-request-approval-rule  \
    --approval-rule-name "My Approval Rule"  \
    --pull-request-id 15
```

輸出：

```
{
    "approvalRuleId": "077d8e8a8-EXAMPLE"
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的編輯或刪除核准規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePullRequestApprovalRule。

delete-repository

以下程式碼範例顯示如何使用 delete-repository。

AWS CLI

刪除儲存庫

此範例示範如何刪除 AWS CodeCommit 儲存庫。

命令：

```
aws codecommit delete-repository --repository-name MyDemoRepo
```

輸出：

```
{
  "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRepository。

describe-merge-conflicts

以下程式碼範例顯示如何使用 describe-merge-conflicts。

AWS CLI

取得合併衝突的詳細資訊

下列describe-merge-conflicts範例會使用 THREE_WAY_MERGE 策略，判斷指定來源分支和目的地分支readme.md中名為 之檔案的合併衝突。

```
aws codecommit describe-merge-conflicts \
    --source-commit-specifier feature-randomizationfeature \
    --destination-commit-specifier main \
    --merge-option THREE_WAY_MERGE \
    --file-path readme.md \
    --repository-name MyDemoRepo
```

輸出：

```
{
    "conflictMetadata": {
        "filePath": "readme.md",
        "fileSizes": {
            "source": 139,
            "destination": 230,
            "base": 85
        },
        "fileModes": {
            "source": "NORMAL",
            "destination": "NORMAL",
            "base": "NORMAL"
        },
        "objectTypes": {
            "source": "FILE",
            "destination": "FILE",
            "base": "FILE"
        },
        "numberOfConflicts": 1,
        "isBinaryFile": {
```

```
            "source": false,
            "destination": false,
            "base": false
        },
        "contentConflict": true,
        "fileModeConflict": false,
        "objectTypeConflict": false,
        "mergeOperations": {
            "source": "M",
            "destination": "M"
        }
    },
    "mergeHunks": [
        {
            "isConflict": true,
            "source": {
                "startLine": 0,
                "endLine": 3,
                "hunkContent": "VGhpcyBpEXAMPLE="
            },
            "destination": {
                "startLine": 0,
                "endLine": 1,
                "hunkContent": "VXNlIHRoEXAMPLE="
            }
        }
    ],
    "destinationCommitId": "86958e0aEXAMPLE",
    "sourceCommitId": "6ccd57fdEXAMPLE",
    "baseCommitId": "767b69580EXAMPLE"
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的解決提取請求中的衝突。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeMergeConflicts。

describe-pull-request-events

以下程式碼範例顯示如何使用 `describe-pull-request-events`。

AWS CLI

檢視提取請求中的事件

下列describe-pull-request-events範例會擷取 ID 為 '8' 之提取請求的事件。

```
aws codecommit describe-pull-request-events --pull-request-id 8
```

輸出：

```
{
    "pullRequestEvents": [
        {
            "pullRequestId": "8",
            "pullRequestEventType": "PULL_REQUEST_CREATED",
            "eventDate": 1510341779.53,
            "actor": "arn:aws:iam::111111111111:user/Zhang_Wei"
        },
        {
            "pullRequestStatusChangedEventMetadata": {
                "pullRequestStatus": "CLOSED"
            },
            "pullRequestId": "8",
            "pullRequestEventType": "PULL_REQUEST_STATUS_CHANGED",
            "eventDate": 1510341930.72,
            "actor": "arn:aws:iam::111111111111:user/Jane_Doe"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePullRequestEvents。

disassociate-approval-rule-template-from-repository

以下程式碼範例顯示如何使用 disassociate-approval-rule-template-from-repository。

AWS CLI

取消核准規則範本與儲存庫的關聯

下列disassociate-approval-rule-template-from-repository範例會將指定的核准規則範本與名為 的儲存庫取消關聯MyDemoRepo。

```
aws codecommit disassociate-approval-rule-template-from-repository \
    --repository-name MyDemoRepo  \
    --approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的取消與核准規則範本的關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateApprovalRuleTemplateFromRepository。

evaluate-pull-request-approval-rules

以下程式碼範例顯示如何使用 evaluate-pull-request-approval-rules。

AWS CLI

評估提取請求是否滿足其所有核准規則

下列evaluate-pull-request-approval-rules範例會評估指定提取請求的核准規則狀態。在此範例中，提取請求尚未滿足核准規則，因此命令的輸出會顯示 approved的值false。

```
aws codecommit evaluate-pull-request-approval-rules \
    --pull-request-id 27  \
    --revision-id 9f29d167EXAMPLE
```

輸出：

```
{
    "evaluation": {
        "approved": false,
        "approvalRulesNotSatisfied": [
            "Require two approved approvers"
        ],
        "overridden": false,
        "approvalRulesSatisfied": []
    }
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的合併提取請求。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EvaluatePullRequestApprovalRules。

get-approval-rule-template

以下程式碼範例顯示如何使用 get-approval-rule-template。

AWS CLI

取得核准規則範本的內容

下列get-approval-rule-template範例取得名為 的核准規則範本內容1-approver-rule-for-all-pull-requests。

```
aws codecommit get-approval-rule-template \
    --approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

輸出：

```
{
    "approvalRuleTemplate": {
        "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 1,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
        "ruleContentSha256": "621181bbEXAMPLE",
        "lastModifiedDate": 1571356106.936,
        "creationDate": 1571356106.936,
        "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Li_Juan",
        "approvalRuleTemplateId": "a29abb15-EXAMPLE",
        "approvalRuleTemplateDescription": "All pull requests must be approved by
 one developer on the team."
    }
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的管理核准規則範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetApprovalRuleTemplate。

get-blob

以下程式碼範例顯示如何使用 get-blob。

AWS CLI

檢視 Git Blob 物件的相關資訊

下列get-blob範例會在名為 'MyDemoRepo' 的CodeCommit 儲存庫中，擷取 ID 為 '2eb4af3bEXAMPLE' 的 Git Blob 相關資訊。 AWS CodeCommit

```
aws codecommit get-blob  --repository-name MyDemoRepo  --blob-id 2eb4af3bEXAMPLE
```

輸出：

```
{
    "content": "QSBCaW5hcnkgTGFyToEXAMPLE="
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBlob。

get-branch

以下程式碼範例顯示如何使用 get-branch。

AWS CLI

取得分支的相關資訊

此範例會取得 AWS CodeCommit 儲存庫中分支的相關資訊。

命令：

```
aws codecommit get-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

輸出：

```
{
  "BranchInfo": {
        "commitID": "317f8570EXAMPLE",
                "branchName": "MyNewBranch"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBranch。

get-comment-reactions

以下程式碼範例顯示如何使用 get-comment-reactions。

AWS CLI

檢視對註解的表情符號反應

下列get-comment-reactions範例會列出 ID 為 之註解的所有表情符號反應abcd1234EXAMPLEb5678efgh。如果您的 shell 字型支援顯示表情符號 1.0 版，則會emoji在表情符號的輸出中顯示 。

```
aws codecommit get-comment-reactions \
    --comment-id abcd1234EXAMPLEb5678efgh
```

輸出：

```
{
    "reactionsForComment": {
        [
           {
               "reaction": {
                    "emoji:"??",
                    "shortCode": "thumbsup",
                    "unicode": "U+1F44D"
                },
                "users": [
                    "arn:aws:iam::123456789012:user/Li_Juan",
                    "arn:aws:iam::123456789012:user/Mary_Major",
                    "arn:aws:iam::123456789012:user/Jorge_Souza"
                ]
            },
            {
                "reaction": {
                    "emoji": "??",
                    "shortCode": "thumbsdown",
                    "unicode": "U+1F44E"
                },
                "users": [
                    "arn:aws:iam::123456789012:user/Nikhil_Jayashankar"
                ]
            },
            {
                "reaction": {
                    "emoji": "??",
                    "shortCode": "confused",
                    "unicode": "U+1F615"
```

```
                },
                "users": [
                    "arn:aws:iam::123456789012:user/Saanvi_Sarkar"
                ]
            }
        ]
    }
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的對遞交 inCodeCommit 的註解。 AWS CodeCommit

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCommentReactions。

get-comment

以下程式碼範例顯示如何使用 get-comment。

AWS CLI

檢視註解的詳細資訊

此範例示範如何使用系統產生的註解 ID 來檢視註解的詳細資訊ff30b348EXAMPLEb9aa670f。

```
aws codecommit get-comment \
    --comment-id ff30b348EXAMPLEb9aa670f
```

輸出：

```
{
    "comment": {
        "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
        "clientRequestToken": "123Example",
        "commentId": "ff30b348EXAMPLEb9aa670f",
        "content": "Whoops - I meant to add this comment to the line, but I don't
 see how to delete it.",
        "creationDate": 1508369768.142,
        "deleted": false,
        "commentId": "",
        "lastModifiedDate": 1508369842.278,
        "callerReactions": [],
        "reactionCounts":
        {
```

```
            "SMILE" : 6,
            "THUMBSUP" : 1
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetComment。

get-comments-for-compared-commit

以下程式碼範例顯示如何使用 get-comments-for-compared-commit。

AWS CLI

檢視遞交的評論

此範例示範如何檢視對名為 的儲存庫中兩個遞交之間的比較所做的評論MyDemoRepo。

```
aws codecommit get-comments-for-compared-commit \
    --repository-name MyDemoRepo \
    --before-commit-ID 6e147360EXAMPLE \
    --after-commit-id 317f8570EXAMPLE
```

輸出：

```
{
    "commentsForComparedCommitData": [
        {
            "afterBlobId": "1f330709EXAMPLE",
            "afterCommitId": "317f8570EXAMPLE",
            "beforeBlobId": "80906a4cEXAMPLE",
            "beforeCommitId": "6e147360EXAMPLE",
            "comments": [
                {
                    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
                    "clientRequestToken": "123Example",
                    "commentId": "ff30b348EXAMPLEb9aa670f",
                    "content": "Whoops - I meant to add this comment to the line,
 not the file, but I don't see how to delete it.",
                    "creationDate": 1508369768.142,
                    "deleted": false,
                    "CommentId": "123abc-EXAMPLE",
                    "lastModifiedDate": 1508369842.278,
```

```
                    "callerReactions": [],
                    "reactionCounts":
                    {
                        "SMILE" : 6,
                        "THUMBSUP" : 1
                    }
                },
                {
                    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
                    "clientRequestToken": "123Example",
                    "commentId": "553b509bEXAMPLE56198325",
                    "content": "Can you add a test case for this?",
                    "creationDate": 1508369612.240,
                    "deleted": false,
                    "commentId": "456def-EXAMPLE",
                    "lastModifiedDate": 1508369612.240,
                    "callerReactions": [],
                    "reactionCounts":
                    {
                        "THUMBSUP" : 2
                    }
                }
            ],
            "location": {
                "filePath": "cl_sample.js",
                "filePosition": 1232,
                "relativeFileVersion": "after"
            },
            "repositoryName": "MyDemoRepo"
        }
    ],
    "nextToken": "exampleToken"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCommentsForComparedCommit。

get-comments-for-pull-request

以下程式碼範例顯示如何使用 get-comments-for-pull-request。

AWS CLI

檢視提取請求的註解

此範例示範如何在名為 的儲存庫中檢視提取請求的註解MyDemoRepo。

```
aws codecommit get-comments-for-pull-request \
    --repository-name MyDemoRepo \
    --before-commit-ID 317f8570EXAMPLE \
    --after-commit-id 5d036259EXAMPLE
```

輸出：

```
{
    "commentsForPullRequestData": [
        {
            "afterBlobId": "1f330709EXAMPLE",
            "afterCommitId": "5d036259EXAMPLE",
            "beforeBlobId": "80906a4cEXAMPLE",
            "beforeCommitId": "317f8570EXAMPLE",
            "comments": [
                {
                    "authorArn": "arn:aws:iam::111111111111:user/Saanvi_Sarkar",
                    "clientRequestToken": "",
                    "commentId": "abcd1234EXAMPLEb5678efgh",
                    "content": "These don't appear to be used anywhere. Can we
 remove them?",
                    "creationDate": 1508369622.123,
                    "deleted": false,
                    "lastModifiedDate": 1508369622.123,
                    "callerReactions": [],
                    "reactionCounts":
                    {
                        "THUMBSUP" : 6,
                        "CONFUSED" : 1
                    }
                },
                {
                    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
                    "clientRequestToken": "",
                    "commentId": "442b498bEXAMPLE5756813",
                    "content": "Good catch. I'll remove them.",
                    "creationDate": 1508369829.104,
                    "deleted": false,
                    "lastModifiedDate": 150836912.273,
                    "callerReactions": ["THUMBSUP"]
                    "reactionCounts":
```

```
                    {
                        "THUMBSUP" : 14
                    }
                }
            ],
            "location": {
                "filePath": "ahs_count.py",
                "filePosition": 367,
                "relativeFileVersion": "AFTER"
            },
            "repositoryName": "MyDemoRepo",
            "pullRequestId": "42"
        }
    ],
    "nextToken": "exampleToken"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCommentsForPullRequest。

get-commit

以下程式碼範例顯示如何使用 get-commit。

AWS CLI

檢視儲存庫中遞交的相關資訊

此範例顯示在名為 'MyDemoRepo' 的 AWS CodeCommit 儲存庫中，系統產生 ID 為 '7e9fd3091thisisanexamplethisisanexample1' 的遞交詳細資訊。

命令：

```
aws codecommit get-commit --repository-name MyDemoRepo --commit-
id 7e9fd3091thisisanexamplethisisanexample1
```

輸出：

```
{
  "commit": {
      "additionalData": "",
      "committer": {
          "date": "1484167798 -0800",
          "name": "Mary Major",
```

```
            "email": "mary_major@example.com"
        },
        "author": {
            "date": "1484167798 -0800",
            "name": "Mary Major",
            "email": "mary_major@example.com"
        },
        "treeId": "347a3408thisisanexampletreeidexample",
        "parents": [
            "7aa87a031thisisanexamplethisisanexample1"
        ],
        "message": "Fix incorrect variable name"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCommit。

get-differences

以下程式碼範例顯示如何使用 get-differences。

AWS CLI

取得儲存庫中遞交指標差異的相關資訊

此範例顯示在 AWS CodeCommit 名為 MyDemoRepo 的重新命名資料夾中，檢視兩個遞交指標（分支、標籤、HEAD 或其他完整參考，例如遞交 IDs) 之間變更的中繼資料資訊。此範例包含幾個不需要的選項，包括 --before-commit-specifier、--before-path 和 --after-path，以便更完整地說明如何使用這些選項來限制結果。回應包含檔案模式許可。

命令：

```
aws codecommit get-differences --repository-name MyDemoRepo --before-
commit-specifier 955bba12thisisanexamplethisisanexample --after-commit-
specifier 14a95463thisisanexamplethisisanexample --before-path tmp/example-folder --
after-path tmp/renamed-folder
```

輸出：

```
{
  "differences": [
      {
```

```
            "afterBlob": {
                "path": "blob.txt",
                "blobId": "2eb4af3b1thisisanexamplethisisanexample1",
                "mode": "100644"
            },
            "changeType": "M",
            "beforeBlob": {
                "path": "blob.txt",
                "blobId": "bf7fcf281thisisanexamplethisisanexample1",
                "mode": "100644"
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDifferences。

get-file

以下程式碼範例顯示如何使用 get-file。

AWS CLI

取得 AWS CodeCommit 儲存庫中檔案的 base-64 編碼內容

下列get-file範例示範如何README.md從名為 的儲存庫main中名為 的分支取得名為 之檔案的 base-64 編碼內容MyDemoRepo。

```
aws codecommit get-file \
    --repository-name MyDemoRepo \
    --commit-specifier main \
    --file-path README.md
```

輸出：

```
{
    "blobId":"559b44fEXAMPLE",
    "commitId":"c5709475EXAMPLE",
    "fileContent":"IyBQaHVzEXAMPLE",
    "filePath":"README.md",
    "fileMode":"NORMAL",
    "fileSize":1563
```

```
}
```

如需詳細資訊，請參閱 AWS CodeCommit API 參考指南中的 GetFile。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFile。

get-folder

以下程式碼範例顯示如何使用 get-folder。

AWS CLI

取得 AWS CodeCommit 儲存庫中資料夾的內容

下列get-folder範例示範如何從名為 的儲存庫取得最上層資料夾的內容MyDemoRepo。

```
aws codecommit get-folder --repository-name MyDemoRepo --folder-path ""
```

輸出：

```
{
    "commitId":"c5709475EXAMPLE",
    "files":[
        {
            "absolutePath":".gitignore",
            "blobId":"74094e8bEXAMPLE",
            "fileMode":"NORMAL",
            "relativePath":".gitignore"
        },
        {
            "absolutePath":"Gemfile",
            "blobId":"9ceb72f6EXAMPLE",
            "fileMode":"NORMAL",
            "relativePath":"Gemfile"
        },
        {
            "absolutePath":"Gemfile.lock",
            "blobId":"795c4a2aEXAMPLE",
            "fileMode":"NORMAL",
            "relativePath":"Gemfile.lock"
        },
        {
            "absolutePath":"LICENSE.txt",
```

```
            "blobId":"0c7932c8EXAMPLE",
            "fileMode":"NORMAL",
            "relativePath":"LICENSE.txt"
        },
        {
            "absolutePath":"README.md",
            "blobId":"559b44feEXAMPLE",
            "fileMode":"NORMAL",
            "relativePath":"README.md"
        }
    ],
    "folderPath":"",
    "subFolders":[
        {
            "absolutePath":"public",
            "relativePath":"public",
            "treeId":"d5e92ae3aEXAMPLE"
        },
        {
            "absolutePath":"tmp",
            "relativePath":"tmp",
            "treeId":"d564d0bcEXAMPLE"
        }
    ],
    "subModules":[],
    "symbolicLinks":[],
    "treeId":"7b3c4dadEXAMPLE"
}
```

如需詳細資訊，請參閱 AWS CodeCommit API 參考指南中的 GetFolder。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFolder。

get-merge-commit

以下程式碼範例顯示如何使用 get-merge-commit。

AWS CLI

取得合併遞交的詳細資訊

下列get-merge-commit範例顯示名為 之來源分支與名為 之bugfix-bug1234儲存庫main中目的地分支合併遞交的詳細資訊MyDemoRepo。

```
aws codecommit get-merge-commit \
    --source-commit-specifier bugfix-bug1234 \
    --destination-commit-specifier main \
    --repository-name MyDemoRepo
```

輸出：

```
{
    "sourceCommitId": "c5709475EXAMPLE",
    "destinationCommitId": "317f8570EXAMPLE",
    "baseCommitId": "fb12a539EXAMPLE",
    "mergeCommitId": "ffc4d608eEXAMPLE"
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的檢視遞交詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMergeCommit。

get-merge-conflicts

以下程式碼範例顯示如何使用 get-merge-conflicts。

AWS CLI

檢視提取請求是否有任何合併衝突

下列get-merge-conflicts範例顯示名為 的來源分支尖端feature-randomizationfeature與名為 'main' 的目的地分支在名為 的儲存庫中是否有任何合併衝突MyDemoRepo。

```
aws codecommit get-merge-conflicts \
    --repository-name MyDemoRepo \
    --source-commit-specifier feature-randomizationfeature \
    --destination-commit-specifier main \
    --merge-option THREE_WAY_MERGE
```

輸出：

```
{
    "mergeable": false,
```

```
    "destinationCommitId": "86958e0aEXAMPLE",
    "sourceCommitId": "6ccd57fdEXAMPLE",
    "baseCommitId": "767b6958EXAMPLE",
    "conflictMetadataList": [
        {
            "filePath": "readme.md",
            "fileSizes": {
                "source": 139,
                "destination": 230,
                "base": 85
            },
            "fileModes": {
                "source": "NORMAL",
                "destination": "NORMAL",
                "base": "NORMAL"
            },
            "objectTypes": {
                "source": "FILE",
                "destination": "FILE",
                "base": "FILE"
            },
            "numberOfConflicts": 1,
            "isBinaryFile": {
                "source": false,
                "destination": false,
                "base": false
            },
            "contentConflict": true,
            "fileModeConflict": false,
            "objectTypeConflict": false,
            "mergeOperations": {
                "source": "M",
                "destination": "M"
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMergeConflicts。

get-merge-options

以下程式碼範例顯示如何使用 get-merge-options。

AWS CLI

取得可用於合併兩個指定分支之合併選項的相關資訊

下列get-merge-options範例會決定可用於合併名為 的來源分支bugfix-bug1234與名為 的儲存庫main中名為 的目的地分支的合併選項MyDemoRepo。

```
aws codecommit get-merge-options \
    --source-commit-specifier bugfix-bug1234 \
    --destination-commit-specifier main \
    --repository-name MyDemoRepo
```

輸出：

```
{
    "mergeOptions": [
        "FAST_FORWARD_MERGE",
        "SQUASH_MERGE",
        "THREE_WAY_MERGE"
    ],
    "sourceCommitId": "18059494EXAMPLE",
    "destinationCommitId": "ffd3311dEXAMPLE",
    "baseCommitId": "ffd3311dEXAMPLE"
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的解決提取請求中的衝突。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMergeOptions。

get-pull-request-approval-states

以下程式碼範例顯示如何使用 get-pull-request-approval-states。

AWS CLI

檢視提取請求的核准

下列get-pull-request-approval-states範例會傳回指定提取請求的核准。

```
aws codecommit get-pull-request-approval-states \
    --pull-request-id 8 \
```

```
    --revision-id 9f29d167EXAMPLE
```

輸出：

```
{
    "approvals": [
        {
            "userArn": "arn:aws:iam::123456789012:user/Mary_Major",
            "approvalState": "APPROVE"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的檢視提取請求。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPullRequestApprovalStates。

get-pull-request-override-state

以下程式碼範例顯示如何使用 get-pull-request-override-state。

AWS CLI

取得提取請求覆寫狀態的相關資訊

下列get-pull-request-override-state範例會傳回指定提取請求的覆寫狀態。在此範例中，提取請求的核准規則已由名為 Mary Major 的使用者覆寫，因此輸出會傳回 的值true。：

```
aws codecommit get-pull-request-override-state \
    --pull-request-id 34  \
    --revision-id 9f29d167EXAMPLE
```

輸出：

```
{
    "overridden": true,
    "overrider": "arn:aws:iam::123456789012:user/Mary_Major"
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的覆寫提取請求上的核准規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPullRequestOverrideState。

get-pull-request

以下程式碼範例顯示如何使用 get-pull-request。

AWS CLI

檢視提取請求的詳細資訊

此範例示範如何檢視 ID 為 之提取請求的相關資訊27。

```
aws codecommit get-pull-request \
    --pull-request-id 27
```

輸出：

```
{
    "pullRequest": {
        "approvalRules": [
            {
                "approvalRuleContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
                "approvalRuleId": "dd8b17fe-EXAMPLE",
                "approvalRuleName": "2-approver-rule-for-main",
                "creationDate": 1571356106.936,
                "lastModifiedDate": 571356106.936,
                "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
                "ruleContentSha256": "4711b576EXAMPLE"
            }
        ],
        "lastActivityDate": 1562619583.565,
        "pullRequestTargets": [
            {
                "sourceCommit": "ca45e279EXAMPLE",
                "sourceReference": "refs/heads/bugfix-1234",
                "mergeBase": "a99f5ddbEXAMPLE",
                "destinationReference": "refs/heads/main",
                "mergeMetadata": {
                    "isMerged": false
                },
                "destinationCommit": "2abfc6beEXAMPLE",
                "repositoryName": "MyDemoRepo"
            }
        ],
```

```
        "revisionId": "e47def21EXAMPLE",
        "title": "Quick fix for bug 1234",
        "authorArn": "arn:aws:iam::123456789012:user/Nikhil_Jayashankar",
        "clientRequestToken": "d8d7612e-EXAMPLE",
        "creationDate": 1562619583.565,
        "pullRequestId": "27",
        "pullRequestStatus": "OPEN"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPullRequest。

get-repository-triggers

以下程式碼範例顯示如何使用 get-repository-triggers。

AWS CLI

取得儲存庫中觸發程序的相關資訊

此範例顯示針對名為 的 AWS CodeCommit 儲存庫設定觸發的詳細資訊MyDemoRepo。

```
aws codecommit get-repository-triggers \
    --repository-name MyDemoRepo
```

輸出：

```
{
    "configurationId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
    "triggers": [
        {
            "destinationArn": "arn:aws:sns:us-
east-1:111111111111:MyCodeCommitTopic",
            "branches": [
                "main",
                "preprod"
            ],
            "name": "MyFirstTrigger",
            "customData": "",
            "events": [
                "all"
            ]
        },
```

```
        {
            "destinationArn": "arn:aws:lambda:us-
east-1:111111111111:function:MyCodeCommitPythonFunction",
            "branches": [],
            "name": "MySecondTrigger",
            "customData": "EXAMPLE",
            "events": [
                "all"
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRepositoryTriggers。

get-repository

以下程式碼範例顯示如何使用 get-repository。

AWS CLI

取得儲存庫的相關資訊

此範例顯示有關 AWS CodeCommit 儲存庫的詳細資訊。

```
aws codecommit get-repository \
    --repository-name MyDemoRepo
```

輸出：

```
{
    "repositoryMetadata": {
        "creationDate": 1429203623.625,
        "defaultBranch": "main",
        "repositoryName": "MyDemoRepo",
        "cloneUrlSsh": "ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/v1/
repos/MyDemoRepo",
        "lastModifiedDate": 1430783812.0869999,
        "repositoryDescription": "My demonstration repository",
        "cloneUrlHttp": "https://codecommit.us-east-1.amazonaws.com/v1/repos/
MyDemoRepo",
        "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
        "Arn": "arn:aws:codecommit:us-east-1:80398EXAMPLE:MyDemoRepo
```

```
        "accountId": "111111111111"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRepository。

`list-approval-rule-templates`

以下程式碼範例顯示如何使用 `list-approval-rule-templates`。

AWS CLI

列出 AWS 區域中的所有核准規則範本

下列`list-approval-rule-templates`範例列出指定區域中的所有核准規則範本。如果未將 AWS 區域指定為參數，則命令會傳回用於執行命令之 CLI AWS 設定檔中指定區域的核准規則範本。

```
aws codecommit list-approval-rule-templates \
    --region us-east-2
```

輸出：

```
{
    "approvalRuleTemplateNames": [
        "2-approver-rule-for-main",
        "1-approver-rule-for-all-pull-requests"
    ]
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的管理核准規則範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListApprovalRuleTemplates。

`list-associated-approval-rule-templates-for-repository`

以下程式碼範例顯示如何使用 `list-associated-approval-rule-templates-for-repository`。

AWS CLI

列出與儲存庫相關聯的所有範本

下列list-associated-approval-rule-templates-for-repository範例列出與名為 的儲存庫相關聯的所有核准規則範本MyDemoRepo。

```
aws codecommit list-associated-approval-rule-templates-for-repository \
    --repository-name MyDemoRepo
```

輸出：

```
{
  "approvalRuleTemplateNames": [
    "2-approver-rule-for-main",
    "1-approver-rule-for-all-pull-requests"
  ]
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的管理核准規則範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAssociatedApprovalRuleTemplatesForRepository。

list-branches

以下程式碼範例顯示如何使用 list-branches。

AWS CLI

檢視分支名稱清單

此範例列出 AWS CodeCommit 儲存庫中的所有分支名稱。

```
aws codecommit list-branches \
    --repository-name MyDemoRepo
```

輸出：

```
{
    "branches": [
        "MyNewBranch",
        "main"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBranches。

`list-pull-requests`

以下程式碼範例顯示如何使用 `list-pull-requests`。

AWS CLI

檢視儲存庫中的提取請求清單

此範例示範如何在名為 'MyDemoRepo' 的 AWS CodeCommit 儲存庫中，列出 ARN 為 'arn：aws：iam：：111111111111：user/Li_Juan' 且狀態為 'CLOSED' 的 IAM 使用者所建立的提取請求：

```
aws codecommit list-pull-requests --author-arn arn:aws:iam::111111111111:user/
Li_Juan --pull-request-status CLOSED --repository-name MyDemoRepo
```

輸出：

```
{
 "nextToken": "",
 "pullRequestIds": ["2","12","16","22","23","35","30","39","47"]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPullRequests。

`list-repositories-for-approval-rule-template`

以下程式碼範例顯示如何使用 `list-repositories-for-approval-rule-template`。

AWS CLI

列出與範本相關聯的所有儲存庫

下列`list-repositories-for-approval-rule-template`範例列出與指定核准規則範本相關聯的所有儲存庫。

```
aws codecommit list-repositories-for-approval-rule-template \
    --approval-rule-template-name 2-approver-rule-for-main
```

輸出：

```
{
    "repositoryNames": [
        "MyDemoRepo",
        "MyClonedRepo"
    ]
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的管理核准規則範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRepositoriesForApprovalRuleTemplate。

list-repositories

以下程式碼範例顯示如何使用 list-repositories。

AWS CLI

檢視儲存庫清單

此範例列出與使用者帳戶相關聯的所有 AWS CodeCommit 儲存庫 AWS 。

命令：

```
aws codecommit list-repositories
```

輸出：

```
{
  "repositories": [
      {
         "repositoryName": "MyDemoRepo"
         "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
      },
      {
         "repositoryName": "MyOtherDemoRepo"
         "repositoryId": "cfc29ac4-b0cb-44dc-9990-f6f51EXAMPLE"
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRepositories。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

檢視儲存庫的 AWS 標籤

下列`list-tags-for-resource`範例列出指定儲存庫的標籤索引鍵和標籤值。

```
aws codecommit list-tags-for-resource \
    --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo
```

輸出：

```
{
    "tags": {
        "Status": "Secret",
        "Team": "Saanvi"
    }
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的檢視儲存庫的標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`merge-branches-by-fast-forward`

以下程式碼範例顯示如何使用 `merge-branches-by-fast-forward`。

AWS CLI

使用快轉合併策略合併兩個分支

下列`merge-branches-by-fast-forward`範例會將指定的來源分支與名為 的儲存庫中的指定目的地分支合併MyDemoRepo。

```
aws codecommit merge-branches-by-fast-forward \
    --source-commit-specifier bugfix-bug1234 \
```

```
    --destination-commit-specifier bugfix-bug1233 \
    --repository-name MyDemoRepo
```

輸出：

```
{
    "commitId": "4f178133EXAMPLE",
    "treeId": "389765daEXAMPLE"
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的比較和合併分支。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 MergeBranchesByFastForward。

merge-branches-by-squash

以下程式碼範例顯示如何使用 merge-branches-by-squash。

AWS CLI

使用 squash 合併策略合併兩個分支

下列merge-branches-by-squash範例會將指定的來源分支與名為 的儲存庫中的指定目的地分支合併MyDemoRepo。

```
aws codecommit merge-branches-by-squash \
    --source-commit-specifier bugfix-bug1234 \
    --destination-commit-specifier bugfix-bug1233 \
    --author-name "Maria Garcia" \
    --email "maria_garcia@example.com" \
    --commit-message "Merging two fix branches to prepare for a general patch." \
    --repository-name MyDemoRepo
```

輸出：

```
{
    "commitId": "4f178133EXAMPLE",
    "treeId": "389765daEXAMPLE"
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的比較和合併分支。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 MergeBranchesBySquash。

merge-branches-by-three-way

以下程式碼範例顯示如何使用 merge-branches-by-three-way。

AWS CLI

使用三向合併策略合併兩個分支

下列merge-branches-by-three-way範例會將指定的來源分支與名為 的儲存庫中的指定目的地分支合併MyDemoRepo。

```
aws codecommit merge-branches-by-three-way \
    --source-commit-specifier main \
    --destination-commit-specifier bugfix-bug1234 \
    --author-name "Jorge Souza" --email "jorge_souza@example.com" \
    --commit-message "Merging changes from main to bugfix branch before additional
 testing." \
    --repository-name MyDemoRepo
```

輸出：

```
{
    "commitId": "4f178133EXAMPLE",
    "treeId": "389765daEXAMPLE"
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的比較和合併分支。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 MergeBranchesByThreeWay。

merge-pull-request-by-fast-forward

以下程式碼範例顯示如何使用 merge-pull-request-by-fast-forward。

AWS CLI

合併並關閉提取請求

此範例示範如何在名為 的儲存庫中合併和關閉 ID 為 '47' 和來源遞交 ID 為 '99132ab0EXAMPLE' 的提取請求MyDemoRepo。

```
aws codecommit merge-pull-request-by-fast-forward \
    --pull-request-id 47 \
    --source-commit-id 99132ab0EXAMPLE \
    --repository-name MyDemoRepo
```

輸出：

```
{
    "pullRequest": {
        "approvalRules": [
            {
                "approvalRuleContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 1,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
                "approvalRuleId": "dd8b17fe-EXAMPLE",
                "approvalRuleName": "I want one approver for this pull request",
                "creationDate": 1571356106.936,
                "lastModifiedDate": 571356106.936,
                "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
                "ruleContentSha256": "4711b576EXAMPLE"
            }
        ],
        "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
        "clientRequestToken": "",
        "creationDate": 1508530823.142,
        "description": "Review the latest changes and updates to the global
 variables",
        "lastActivityDate": 1508887223.155,
        "pullRequestId": "47",
        "pullRequestStatus": "CLOSED",
        "pullRequestTargets": [
            {
                "destinationCommit": "9f31c968EXAMPLE",
                "destinationReference": "refs/heads/main",
                "mergeMetadata": {
                    "isMerged": true,
                    "mergedBy": "arn:aws:iam::123456789012:user/Mary_Major"
                },
                "repositoryName": "MyDemoRepo",
                "sourceCommit": "99132ab0EXAMPLE",
                "sourceReference": "refs/heads/variables-branch"
            }
        ],
```

```
        "title": "Consolidation of global variables"
    }
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的合併提取請求。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 MergePullRequestByFastForward。

merge-pull-request-by-squash

以下程式碼範例顯示如何使用 merge-pull-request-by-squash。

AWS CLI

使用 squash 合併策略合併提取請求

下列merge-pull-request-by-squash範例會使用名為 的儲存庫中 ACCEPT_SOURCE 的衝突解決策略合併和關閉指定的提取請求MyDemoRepo。

```
aws codecommit merge-pull-request-by-squash \
    --pull-request-id 47 \
    --source-commit-id 99132ab0EXAMPLE \
    --repository-name MyDemoRepo \
    --conflict-detail-level LINE_LEVEL \
    --conflict-resolution-strategy ACCEPT_SOURCE \
    --name "Jorge Souza" --email "jorge_souza@example.com" \
    --commit-message "Merging pull request 47 by squash and accepting source in
 merge conflicts"
```

輸出：

```
{
    "pullRequest": {
        "approvalRules": [
            {
                "approvalRuleContent": "{\"Version\": \"2018-11-08\",
\"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type
\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
                "approvalRuleId": "dd8b17fe-EXAMPLE",
                "approvalRuleName": "2-approver-rule-for-main",
                "creationDate": 1571356106.936,
                "lastModifiedDate": 571356106.936,
```

```
                "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
                "originApprovalRuleTemplate": {
                    "approvalRuleTemplateId": "dd8b17fe-EXAMPLE",
                    "approvalRuleTemplateName": "2-approver-rule-for-main"
                },
                "ruleContentSha256": "4711b576EXAMPLE"
            }
        ],
        "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
        "clientRequestToken": "",
        "creationDate": 1508530823.142,
        "description": "Review the latest changes and updates to the global
 variables",
        "lastActivityDate": 1508887223.155,
        "pullRequestId": "47",
        "pullRequestStatus": "CLOSED",
        "pullRequestTargets": [
            {
                "destinationCommit": "9f31c968EXAMPLE",
                "destinationReference": "refs/heads/main",
                "mergeMetadata": {
                    "isMerged": true,
                    "mergedBy": "arn:aws:iam::123456789012:user/Mary_Major"
                },
                "repositoryName": "MyDemoRepo",
                "sourceCommit": "99132ab0EXAMPLE",
                "sourceReference": "refs/heads/variables-branch"
            }
        ],
        "title": "Consolidation of global variables"
    }
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的合併提取請求。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 MergePullRequestBySquash。

merge-pull-request-by-three-way

以下程式碼範例顯示如何使用 merge-pull-request-by-three-way。

AWS CLI

使用三向合併策略合併提取請求

下列merge-pull-request-by-three-way範例會使用名為 的儲存庫中衝突詳細資訊和衝突解決策略的預設選項，合併和關閉指定的提取請求MyDemoRepo。

```
aws codecommit merge-pull-request-by-three-way \
    --pull-request-id 47 \
    --source-commit-id 99132ab0EXAMPLE \
    --repository-name MyDemoRepo \
    --name "Maria Garcia" \
    --email "maria_garcia@example.com" \
    --commit-message "Merging pull request 47 by three-way with default options"
```

輸出：

```
{
    "pullRequest": {
        "approvalRules": [
            {
                "approvalRuleContent": "{\"Version\": \"2018-11-08\",
\"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type
\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
                "approvalRuleId": "dd8b17fe-EXAMPLE",
                "approvalRuleName": "2-approver-rule-for-main",
                "creationDate": 1571356106.936,
                "lastModifiedDate": 571356106.936,
                "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
                "originApprovalRuleTemplate": {
                    "approvalRuleTemplateId": "dd8b17fe-EXAMPLE",
                    "approvalRuleTemplateName": "2-approver-rule-for-main"
                },
                "ruleContentSha256": "4711b576EXAMPLE"
            }
        ],
        "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
        "clientRequestToken": "",
        "creationDate": 1508530823.142,
        "description": "Review the latest changes and updates to the global
 variables",
        "lastActivityDate": 1508887223.155,
        "pullRequestId": "47",
        "pullRequestStatus": "CLOSED",
        "pullRequestTargets": [
            {
```

```
                "destinationCommit": "9f31c968EXAMPLE",
                "destinationReference": "refs/heads/main",
                "mergeMetadata": {
                    "isMerged": true,
                    "mergedBy": "arn:aws:iam::123456789012:user/Mary_Major"
                },
                "repositoryName": "MyDemoRepo",
                "sourceCommit": "99132ab0EXAMPLE",
                "sourceReference": "refs/heads/variables-branch"
            }
        ],
        "title": "Consolidation of global variables"
    }
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的合併提取請求。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 MergePullRequestByThreeWay。

override-pull-request-approval-rules

以下程式碼範例顯示如何使用 override-pull-request-approval-rules。

AWS CLI

覆寫提取請求的核准規則需求

下列override-pull-request-approval-rules範例會覆寫指定提取請求的核准規則。若要撤銷覆寫，請將 --override-status 參數值設定為 REVOKE。

```
aws codecommit override-pull-request-approval-rules \
    --pull-request-id 34  \
    --revision-id 927df8d8EXAMPLE \
    --override-status OVERRIDE
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的覆寫提取請求的核准規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 OverridePullRequestApprovalRules。

post-comment-for-compared-commit

以下程式碼範例顯示如何使用 post-comment-for-compared-commit。

AWS CLI

建立遞交的註解

此範例示範如何在名為 的儲存庫中兩個遞交之間的比較中，將變更"Can you add a test case for this?"的註解新增至 cl_sample.js 檔案MyDemoRepo。

```
aws codecommit post-comment-for-compared-commit \
    --repository-name MyDemoRepo \
    --before-commit-id 317f8570EXAMPLE \
    --after-commit-id 5d036259EXAMPLE \
    --client-request-token 123Example \
    --content "Can you add a test case for this?" \
    --location filePath=cl_sample.js,filePosition=1232,relativeFileVersion=AFTER
```

輸出：

```
{
    "afterBlobId": "1f330709EXAMPLE",
    "afterCommitId": "317f8570EXAMPLE",
    "beforeBlobId": "80906a4cEXAMPLE",
    "beforeCommitId": "6e147360EXAMPLE",
    "comment": {
        "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
        "clientRequestToken": "",
        "commentId": "553b509bEXAMPLE56198325",
        "content": "Can you add a test case for this?",
        "creationDate": 1508369612.203,
        "deleted": false,
        "commentId": "abc123-EXAMPLE",
        "lastModifiedDate": 1508369612.203,
        "callerReactions": [],
        "reactionCounts": []
    },
    "location": {
        "filePath": "cl_sample.js",
        "filePosition": 1232,
        "relativeFileVersion": "AFTER"
    ,
```

```
    "repositoryName": "MyDemoRepo"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PostCommentForComparedCommit。

post-comment-for-pull-request

以下程式碼範例顯示如何使用 post-comment-for-pull-request。

AWS CLI

將註解新增至提取請求

以下post-comment-for-pull-request範例新增了註解「這些似乎沒有在任何地方使用。我們是否可以移除它們？」 在名為 的儲存庫中 ID 為 的提取請求47中變更 ahs_count.py 檔案MyDemoRepo。

```
aws codecommit post-comment-for-pull-request \
    --pull-request-id "47" \
    --repository-name MyDemoRepo \
    --before-commit-id 317f8570EXAMPLE \
    --after-commit-id 5d036259EXAMPLE \
    --client-request-token 123Example \
    --content "These don't appear to be used anywhere. Can we remove them?" \
    --location filePath=ahs_count.py,filePosition=367,relativeFileVersion=AFTER
```

輸出：

```
{
     "afterBlobId": "1f330709EXAMPLE",
     "afterCommitId": "5d036259EXAMPLE",
     "beforeBlobId": "80906a4cEXAMPLE",
     "beforeCommitId": "317f8570EXAMPLE",
     "comment": {
            "authorArn": "arn:aws:iam::111111111111:user/Saanvi_Sarkar",
            "clientRequestToken": "123Example",
            "commentId": "abcd1234EXAMPLEb5678efgh",
            "content": "These don't appear to be used anywhere. Can we remove
 them?",
            "creationDate": 1508369622.123,
            "deleted": false,
```

```
            "CommentId": "",
            "lastModifiedDate": 1508369622.123,
            "callerReactions": [],
            "reactionCounts": []
        },
        "location": {
            "filePath": "ahs_count.py",
            "filePosition": 367,
            "relativeFileVersion": "AFTER"
         },
     "repositoryName": "MyDemoRepo",
     "pullRequestId": "47"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PostCommentForPullRequest。

post-comment-reply

以下程式碼範例顯示如何使用 post-comment-reply。

AWS CLI

回覆遞交或提取請求中的註解

此範例示範如何使用系統產生的 ID "Good catch. I'll remove them."將回覆新增至註解abcd1234EXAMPLEb5678efgh。

```
aws codecommit post-comment-reply \
    --in-reply-to abcd1234EXAMPLEb5678efgh \
    --content "Good catch. I'll remove them." \
    --client-request-token 123Example
```

輸出：

```
{
    "comment": {
        "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
        "clientRequestToken": "123Example",
        "commentId": "442b498bEXAMPLE5756813",
        "content": "Good catch. I'll remove them.",
        "creationDate": 1508369829.136,
        "deleted": false,
```

```
        "CommentId": "abcd1234EXAMPLEb5678efgh",
        "lastModifiedDate": 150836912.221,
        "callerReactions": [],
        "reactionCounts": []
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PostCommentReply。

put-comment-reaction

以下程式碼範例顯示如何使用 put-comment-reaction。

AWS CLI

使用表情符號回覆遞交的評論

下列put-comment-reaction範例會回覆 ID 為 且表情符號反應值abcd1234EXAMPLEb5678efgh為 的註解:thumbsup:。

```
aws codecommit put-comment-reaction \
    --comment-id abcd1234EXAMPLEb5678efgh \
    --reaction-value :thumbsup:
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的對遞交 inCodeCommit 的註解。 AWS CodeCommit

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutCommentReaction。

put-file

以下程式碼範例顯示如何使用 put-file。

AWS CLI

將檔案新增至儲存庫

下列put-file範例會將名為 'ExampleSolution.py' 的檔案新增至名為 'MyDemoRepo' 的儲存庫中名為 'feature-randomizationfeature' 的分支，其最近遞交的 ID 為 '4c925148EXAMPLE'。

```
aws codecommit put-file \
    --repository-name MyDemoRepo \
    --branch-name feature-randomizationfeature \
    --file-content file://MyDirectory/ExampleSolution.py \
    --file-path /solutions/ExampleSolution.py \
    --parent-commit-id 4c925148EXAMPLE \
    --name "Maria Garcia" \
    --email "maria_garcia@example.com" \
    --commit-message "I added a third randomization routine."
```

輸出：

```
{
    "blobId": "2eb4af3bEXAMPLE",
    "commitId": "317f8570EXAMPLE",
    "treeId": "347a3408EXAMPLE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutFile。

put-repository-triggers

以下程式碼範例顯示如何使用 put-repository-triggers。

AWS CLI

在儲存庫中新增或更新觸發

此範例示範如何使用已建立的 JSON 檔案 （ 此處名為 MyTriggers.json) 更新名為 'MyFirstTrigger' 和 'MySecondTrigger' 的觸發條件，其中包含名為 MyDemoRepo 之儲存庫的所有觸發條件結構。若要了解如何取得現有觸發條件的 JSON，請參閱 get-repository-triggers 命令。

```
aws codecommit put-repository-triggers \
    --repository-name MyDemoRepo file://MyTriggers.json
```

MyTriggers.json 的內容：

```
{
    "repositoryName": "MyDemoRepo",
    "triggers": [
        {
```

```
            "destinationArn": "arn:aws:sns:us-
east-1:80398EXAMPLE:MyCodeCommitTopic",
            "branches": [
                "main",
                "preprod"
            ],
            "name": "MyFirstTrigger",
            "customData": "",
            "events": [
                "all"
            ]
        },
        {
            "destinationArn": "arn:aws:lambda:us-
east-1:111111111111:function:MyCodeCommitPythonFunction",
            "branches": [],
            "name": "MySecondTrigger",
            "customData": "EXAMPLE",
            "events": [
                "all"
            ]
        }
    ]
}
```

輸出：

```
{
    "configurationId": "6fa51cd8-35c1-EXAMPLE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutRepositoryTriggers。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將 AWS 標籤新增至現有的儲存庫

下列tag-resource範例使用兩個標籤來標記指定的儲存庫。

```
aws codecommit tag-resource \
    --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo \
    --tags Status=Secret,Team=Saanvi
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的將標籤新增至儲存庫。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

`test-repository-triggers`

以下程式碼範例顯示如何使用 `test-repository-triggers`。

AWS CLI

在儲存庫中測試觸發

此範例示範如何在名為 MyDemoRepo 的CodeCommit 儲存庫中測試名為 'MyFirstTrigger' 的觸發。AWS CodeCommit MyDemoRepo 在此範例中，儲存庫中的事件會從 Amazon Simple Notification Service (Amazon SNS) 主題觸發通知。

命令：

```
aws codecommit test-repository-triggers --repository-name MyDemoRepo
 --triggers name=MyFirstTrigger,destinationArn=arn:aws:sns:us-
east-1:111111111111:MyCodeCommitTopic,branches=mainline,preprod,events=all
```

輸出：

```
{
  "successfulExecutions": [
      "MyFirstTrigger"
  ],
  "failedExecutions": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TestRepositoryTriggers。

`untag-resource`

以下程式碼範例顯示如何使用 `untag-resource`。

AWS CLI

從儲存庫移除 AWS 標籤

下列untag-resource範例會從名為 的儲存庫中移除具有指定金鑰的標籤MyDemoRepo。

```
aws codecommit untag-resource \
    --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo \
    --tag-keys Status
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的從儲存庫移除標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-approval-rule-template-content

以下程式碼範例顯示如何使用 update-approval-rule-template-content。

AWS CLI

更新核准規則範本的內容

下列update-approval-rule-template-content範例會變更指定核准規則範本的內容，將核准集區重新定義為擔任 角色的使用者CodeCommitReview。

```
aws codecommit update-approval-rule-template-content \
    --approval-rule-template-name 1-approver-rule  \
    --new-rule-content "{\"Version\": \"2018-11-08\",\"DestinationReferences
\": [\"refs/heads/main\"],\"Statements\": [{\"Type\": \"Approvers
\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}"
```

輸出：

```
{
    "approvalRuleTemplate": {
        "creationDate": 1571352720.773,
        "approvalRuleTemplateDescription": "Requires 1 approval for all pull
 requests from the CodeCommitReview pool",
```

```
        "lastModifiedDate": 1571358728.41,
        "approvalRuleTemplateId": "41de97b7-EXAMPLE",
        "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 1,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
        "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
        "ruleContentSha256": "2f6c21a5EXAMPLE",
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Li_Juan"
    }
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的管理核准規則範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateApprovalRuleTemplateContent。

update-approval-rule-template-description

以下程式碼範例顯示如何使用 update-approval-rule-template-description。

AWS CLI

更新核准規則範本的描述

下列update-approval-rule-template-description範例會將指定核准規則範本的描述變更為 Requires 1 approval for all pull requests from the CodeCommitReview pool。：

```
aws codecommit update-approval-rule-template-description \
    --approval-rule-template-name 1-approver-rule-for-all-pull-requests  \
    --approval-rule-template-description "Requires 1 approval for all pull requests
 from the CodeCommitReview pool"
```

輸出：

```
{
  "approvalRuleTemplate": {
    "creationDate": 1571352720.773,
    "approvalRuleTemplateDescription": "Requires 1 approval for all pull requests
 from the CodeCommitReview pool",
    "lastModifiedDate": 1571358728.41,
    "approvalRuleTemplateId": "41de97b7-EXAMPLE",
```

```
    "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 1,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
    "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
    "ruleContentSha256": "2f6c21a5EXAMPLE",
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Li_Juan"
  }
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的管理核准規則範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateApprovalRuleTemplateDescription。

update-approval-rule-template-name

以下程式碼範例顯示如何使用 update-approval-rule-template-name。

AWS CLI

更新核准規則範本的名稱

下列update-approval-rule-template-name範例會將核准規則範本的名稱從 變更為 1-approver-rule 1-approver-rule-for-all-pull-requests`。

```
aws codecommit update-approval-rule-template-name \
    --old-approval-rule-template-name 1-approver-rule  \
    --new-approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

輸出：

```
{
  "approvalRuleTemplate": {
    "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
    "lastModifiedDate": 1571358241.619,
    "approvalRuleTemplateId": "41de97b7-EXAMPLE",
    "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 1,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
    "creationDate": 1571352720.773,
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
    "approvalRuleTemplateDescription": "All pull requests must be approved by one
 developer on the team.",
```

```
    "ruleContentSha256": "2f6c21a5cEXAMPLE"
  }
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的管理核准規則範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateApprovalRuleTemplateName。

update-comment

以下程式碼範例顯示如何使用 update-comment。

AWS CLI

更新遞交的註解

此範例示範如何將內容新增至 ID 為 的"Fixed as requested. I'll update the pull request."評論442b498bEXAMPLE5756813。

```
aws codecommit update-comment \
    --comment-id 442b498bEXAMPLE5756813 \
    --content "Fixed as requested. I'll update the pull request."
```

輸出：

```
{
    "comment": {
        "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
        "clientRequestToken": "",
        "commentId": "442b498bEXAMPLE5756813",
        "content": "Fixed as requested. I'll update the pull request.",
        "creationDate": 1508369929.783,
        "deleted": false,
        "lastModifiedDate": 1508369929.287,
        "callerReactions": [],
        "reactionCounts":
            {
                "THUMBSUP" : 2
            }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateComment。

update-default-branch

以下程式碼範例顯示如何使用 update-default-branch。

AWS CLI

變更儲存庫的預設分支

此範例會變更 AWS CodeCommit 儲存庫的預設分支。只有在發生錯誤時，此命令才會產生輸出。

命令：

```
aws codecommit update-default-branch --repository-name MyDemoRepo --default-branch-
name MyNewBranch
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDefaultBranch。

update-pull-request-approval-rule-content

以下程式碼範例顯示如何使用 update-pull-request-approval-rule-content。

AWS CLI

編輯提取請求的核准規則

下列update-pull-request-approval-rule-content範例會更新她指定的核准規則，要求核准集區中包含123456789012 AWS 帳戶中任何 IAM 使用者的使用者核准。

```
aws codecommit update-pull-request-approval-rule-content \
    --pull-request-id 27  \
    --approval-rule-name "Require two approved approvers" \
    --approval-rule-content "{Version: 2018-11-08, Statements: [{Type:
 \"Approvers\", NumberOfApprovalsNeeded: 1, ApprovalPoolMembers:
[\"CodeCommitApprovers:123456789012:user/*\"]}]}"
```

輸出：

```
{
    "approvalRule": {
        "approvalRuleContent": "{Version: 2018-11-08, Statements:
 [{Type: \"Approvers\", NumberOfApprovalsNeeded: 1, ApprovalPoolMembers:
[\"CodeCommitApprovers:123456789012:user/*\"]}]}",
        "approvalRuleId": "aac33506-EXAMPLE",
        "originApprovalRuleTemplate": {},
        "creationDate": 1570752871.932,
        "lastModifiedDate": 1570754058.333,
        "approvalRuleName": Require two approved approvers",
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "ruleContentSha256": "cd93921cEXAMPLE",
    }
}
```

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的編輯或刪除核准規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePullRequestApprovalRuleContent。

update-pull-request-approval-state

以下程式碼範例顯示如何使用 update-pull-request-approval-state。

AWS CLI

核准或撤銷提取請求的核准

下列update-pull-request-approval-state範例會核准 ID 為 27且修訂 ID 為 的提取請求9f29d167EXAMPLE。如果您想要撤銷核准，請將 --approval-state 參數值設定為 REVOKE。

```
aws codecommit update-pull-request-approval-state \
    --pull-request-id 27  \
    --revision-id 9f29d167EXAMPLE  \
    --approval-state "APPROVE"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CodeCommit 使用者指南中的檢閱提取請求。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePullRequestApprovalState。

update-pull-request-description

以下程式碼範例顯示如何使用 update-pull-request-description。

AWS CLI

變更提取請求的描述

此範例示範如何變更 ID 為 的提取請求描述47。

```
aws codecommit update-pull-request-description \
    --pull-request-id 47 \
    --description "Updated the pull request to remove unused global variable."
```

輸出：

```
{
    "pullRequest": {
        "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
        "clientRequestToken": "",
        "creationDate": 1508530823.155,
        "description": "Updated the pull request to remove unused global variable.",
        "lastActivityDate": 1508372423.204,
        "pullRequestId": "47",
        "pullRequestStatus": "OPEN",
        "pullRequestTargets": [
            {
                "destinationCommit": "9f31c968EXAMPLE",
                "destinationReference": "refs/heads/main",
                "mergeMetadata": {
                    "isMerged": false,
                },
                "repositoryName": "MyDemoRepo",
                "sourceCommit": "99132ab0EXAMPLE",
                "sourceReference": "refs/heads/variables-branch"
            }
        ],
        "title": "Consolidation of global variables"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePullRequestDescription。

update-pull-request-status

以下程式碼範例顯示如何使用 update-pull-request-status。

AWS CLI

變更提取請求的狀態

此範例示範如何在名為 的 AWS CodeCommit 儲存庫CLOSED中，將 ID 42為 的提取請求狀態變更為 狀態MyDemoRepo。

```
aws codecommit update-pull-request-status \
    --pull-request-id 42 \
    --pull-request-status CLOSED
```

輸出：

```
{
    "pullRequest": {
        "approvalRules": [
            {
                "approvalRuleContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
                "approvalRuleId": "dd8b17fe-EXAMPLE",
                "approvalRuleName": "2-approvers-needed-for-this-change",
                "creationDate": 1571356106.936,
                "lastModifiedDate": 571356106.936,
                "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
                "ruleContentSha256": "4711b576EXAMPLE"
            }
        ],
        "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
        "clientRequestToken": "",
        "creationDate": 1508530823.165,
        "description": "Updated the pull request to remove unused global variable.",
        "lastActivityDate": 1508372423.12,
        "pullRequestId": "47",
        "pullRequestStatus": "CLOSED",
        "pullRequestTargets": [
            {
                "destinationCommit": "9f31c968EXAMPLE",
                "destinationReference": "refs/heads/main",
```

```
                "mergeMetadata": {
                    "isMerged": false,
                },
                "repositoryName": "MyDemoRepo",
                "sourceCommit": "99132ab0EXAMPLE",
                "sourceReference": "refs/heads/variables-branch"
            }
        ],
        "title": "Consolidation of global variables"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePullRequestStatus。

update-pull-request-title

以下程式碼範例顯示如何使用 update-pull-request-title。

AWS CLI

變更提取請求的標題

此範例示範如何變更 ID 為 的提取請求標題47。

```
aws codecommit update-pull-request-title \
    --pull-request-id 47 \
    --title "Consolidation of global variables - updated review"
```

輸出：

```
{
    "pullRequest": {
        "approvalRules": [
            {
                "approvalRuleContent": "{\"Version\": \"2018-11-08\",
\"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type
\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
                "approvalRuleId": "dd8b17fe-EXAMPLE",
                "approvalRuleName": "2-approver-rule-for-main",
                "creationDate": 1571356106.936,
                "lastModifiedDate": 571356106.936,
                "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
```

```
                "originApprovalRuleTemplate": {
                    "approvalRuleTemplateId": "dd8b26gr-EXAMPLE",
                    "approvalRuleTemplateName": "2-approver-rule-for-main"
                },
                "ruleContentSha256": "4711b576EXAMPLE"
            }
        ],
        "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
        "clientRequestToken": "",
        "creationDate": 1508530823.12,
        "description": "Review the latest changes and updates to the global
 variables. I have updated this request with some changes, including removing some
 unused variables.",
        "lastActivityDate": 1508372657.188,
        "pullRequestId": "47",
        "pullRequestStatus": "OPEN",
        "pullRequestTargets": [
            {
                "destinationCommit": "9f31c968EXAMPLE",
                "destinationReference": "refs/heads/main",
                "mergeMetadata": {
                    "isMerged": false,
                },
                "repositoryName": "MyDemoRepo",
                "sourceCommit": "99132ab0EXAMPLE",
                "sourceReference": "refs/heads/variables-branch"
            }
        ],
        "title": "Consolidation of global variables - updated review"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePullRequestTitle。

update-repository-description

以下程式碼範例顯示如何使用 update-repository-description。

AWS CLI

變更儲存庫的描述

此範例會變更 AWS CodeCommit 儲存庫的描述。只有在發生錯誤時，此命令才會產生輸出。

命令：

```
aws codecommit update-repository-description --repository-name MyDemoRepo --
repository-description "This description was changed"
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRepositoryDescription。

update-repository-name

以下程式碼範例顯示如何使用 update-repository-name。

AWS CLI

變更儲存庫的名稱

此範例會變更 AWS CodeCommit 儲存庫的名稱。只有在發生錯誤時，此命令才會產生輸出。變更 AWS CodeCommit 儲存庫的名稱會變更使用者連線到儲存庫所需的 SSH 和 HTTPS URLs。使用者將無法連接到此儲存庫，直到他們更新連線設定為止。此外，由於儲存庫的 ARN 將會變更，因此變更儲存庫名稱會使依賴此儲存庫 ARN 的任何 IAM 使用者政策失效。

命令：

```
aws codecommit update-repository-name --old-name MyDemoRepo --new-
name MyRenamedDemoRepo
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRepositoryName。

使用 的 CodeDeploy 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 CodeDeploy 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`add-tags-to-on-premises-instances`

以下程式碼範例顯示如何使用 `add-tags-to-on-premises-instances`。

AWS CLI

將標籤新增至內部部署執行個體

下列`add-tags-to-on-premises-instances`範例會將 in AWS CodeDeploy 相同的現場部署執行個體標籤與兩個現場部署執行個體建立關聯。它不會向 AWS CodeDeploy 註冊現場部署執行個體。

```
aws deploy add-tags-to-on-premises-instances \
    --instance-names AssetTag12010298EX AssetTag23121309EX \
    --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddTagsToOnPremisesInstances。

`batch-get-application-revisions`

以下程式碼範例顯示如何使用 `batch-get-application-revisions`。

AWS CLI

擷取應用程式修訂的相關資訊

下列`batch-get-application-revisions`範例會擷取 GitHub 儲存庫中存放之指定修訂版的相關資訊。

```
aws deploy batch-get-application-revisions \
```

```
    --application-name my-codedeploy-application \
    --revisions "[{\"gitHubLocation\": {\"commitId\":
 \"fa85936EXAMPLEa31736c051f10d77297EXAMPLE\",\"repository\": \"my-github-token/my-
repository\"},\"revisionType\": \"GitHub\"}]"
```

輸出：

```
{
    "revisions": [
        {
            "genericRevisionInfo": {
                "description": "Application revision registered by Deployment ID: d-
A1B2C3111",
                "lastUsedTime": 1556912355.884,
                "registerTime": 1556912355.884,
                "firstUsedTime": 1556912355.884,
                "deploymentGroups": []
            },
            "revisionLocation": {
                "revisionType": "GitHub",
                "gitHubLocation": {
                    "commitId": "fa85936EXAMPLEa31736c051f10d77297EXAMPLE",
                    "repository": "my-github-token/my-repository"
                }
            }
        }
    ],
    "applicationName": "my-codedeploy-application",
    "errorMessage": ""
}
```

如需詳細資訊，請參閱 AWS CodeDeploy API 參考中的 BatchGetApplicationRevisions。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetApplicationRevisions。

batch-get-applications

以下程式碼範例顯示如何使用 batch-get-applications。

AWS CLI

取得多個應用程式的相關資訊

下列batch-get-applications範例顯示與使用者帳戶相關聯的多個應用程式的相關資訊 AWS 。

```
aws deploy batch-get-applications --application-names WordPress_App MyOther_App
```

輸出：

```
{
    "applicationsInfo": [
        {
            "applicationName": "WordPress_App",
            "applicationId": "d9dd6993-f171-44fa-a811-211e4EXAMPLE",
            "createTime": 1407878168.078,
            "linkedToGitHub": false
        },
        {
            "applicationName": "MyOther_App",
            "applicationId": "8ca57519-31da-42b2-9194-8bb16EXAMPLE",
            "createTime": 1407453571.63,
            "linkedToGitHub": false
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetApplications。

batch-get-deployment-groups

以下程式碼範例顯示如何使用 batch-get-deployment-groups。

AWS CLI

擷取一或多個部署群組的相關資訊

下列batch-get-deployment-groups範例會擷取與指定 CodeDeploy 應用程式相關聯之兩個部署群組的相關資訊。

```
aws deploy batch-get-deployment-groups \
    --application-name my-codedeploy-application \
    --deployment-group-names "[\"my-deployment-group-1\",\"my-deployment-group-2\"]"
```

輸出：

```
{
    "deploymentGroupsInfo": [
        {
            "deploymentStyle": {
                "deploymentOption": "WITHOUT_TRAFFIC_CONTROL",
                "deploymentType": "IN_PLACE"
            },
            "autoRollbackConfiguration": {
                "enabled": false
            },
            "onPremisesTagSet": {
                "onPremisesTagSetList": []
            },
            "serviceRoleArn": "arn:aws:iam::123456789012:role/
CodeDeployServiceRole",
            "lastAttemptedDeployment": {
                "endTime": 1556912366.415,
                "status": "Failed",
                "createTime": 1556912355.884,
                "deploymentId": "d-A1B2C3111"
            },
            "autoScalingGroups": [],
            "deploymentGroupName": "my-deployment-group-1",
            "ec2TagSet": {
                "ec2TagSetList": [
                    [
                        {
                            "Type": "KEY_AND_VALUE",
                            "Value": "my-EC2-instance",
                            "Key": "Name"
                        }
                    ]
                ]
            },
            "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-11111example",
            "triggerConfigurations": [],
            "applicationName": "my-codedeploy-application",
            "computePlatform": "Server",
            "deploymentConfigName": "CodeDeployDefault.AllAtOnce"
        },
        {
            "deploymentStyle": {
```

```
                "deploymentOption": "WITHOUT_TRAFFIC_CONTROL",
                "deploymentType": "IN_PLACE"
            },
            "autoRollbackConfiguration": {
                "enabled": false
            },
            "onPremisesTagSet": {
                "onPremisesTagSetList": []
            },
            "serviceRoleArn": "arn:aws:iam::123456789012:role/
CodeDeployServiceRole",
            "autoScalingGroups": [],
            "deploymentGroupName": "my-deployment-group-2",
            "ec2TagSet": {
                "ec2TagSetList": [
                    [
                        {
                            "Type": "KEY_AND_VALUE",
                            "Value": "my-EC2-instance",
                            "Key": "Name"
                            }
                    ]
                ]
            },
            "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-22222example",
            "triggerConfigurations": [],
            "applicationName": "my-codedeploy-application",
            "computePlatform": "Server",
            "deploymentConfigName": "CodeDeployDefault.AllAtOnce"
        }
    ],
    "errorMessage": ""
}
```

如需詳細資訊，請參閱 CodeDeploy API 參考中的 BatchGetDeploymentGroups。 AWS CodeDeploy

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetDeploymentGroups。

batch-get-deployment-targets

以下程式碼範例顯示如何使用 batch-get-deployment-targets。

AWS CLI

擷取與部署相關聯的目標

下列batch-get-deployment-targets範例會傳回與指定部署相關聯的其中一個目標的相關資訊。

```
aws deploy batch-get-deployment-targets \
    --deployment-id "d-1A2B3C4D5" \
    --target-ids "i-01a2b3c4d5e6f1111"
```

輸出：

```
{
    "deploymentTargets": [
        {
            "deploymentTargetType": "InstanceTarget",
            "instanceTarget": {
                "lifecycleEvents": [
                    {
                        "startTime": 1556918592.162,
                        "lifecycleEventName": "ApplicationStop",
                        "status": "Succeeded",
                        "endTime": 1556918592.247,
                        "diagnostics": {
                            "scriptName": "",
                            "errorCode": "Success",
                            "logTail": "",
                            "message": "Succeeded"
                        }
                    },
                    {
                        "startTime": 1556918593.193,
                        "lifecycleEventName": "DownloadBundle",
                        "status": "Succeeded",
                        "endTime": 1556918593.981,
                        "diagnostics": {
                            "scriptName": "",
                            "errorCode": "Success",
                            "logTail": "",
                            "message": "Succeeded"
                        }
                    },
```

```
                    {
                        "startTime": 1556918594.805,
                        "lifecycleEventName": "BeforeInstall",
                        "status": "Succeeded",
                        "endTime": 1556918681.807,
                        "diagnostics": {
                            "scriptName": "",
                            "errorCode": "Success",
                            "logTail": "",
                            "message": "Succeeded"
                        }
                    }
                ],
                "targetArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-01a2b3c4d5e6f1111",
                "deploymentId": "d-1A2B3C4D5",
                "lastUpdatedAt": 1556918687.504,
                "targetId": "i-01a2b3c4d5e6f1111",
                "status": "Succeeded"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 CodeDeploy API 參考中的 BatchGetDeploymentTargets。 AWS CodeDeploy

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetDeploymentTargets。

batch-get-deployments

以下程式碼範例顯示如何使用 batch-get-deployments。

AWS CLI

取得多個部署的相關資訊

下列batch-get-deployments範例顯示與使用者帳戶相關聯的多個部署的相關資訊 AWS 。

```
aws deploy batch-get-deployments --deployment-ids d-A1B2C3111 d-A1B2C3222
```

輸出：

```
{
    "deploymentsInfo": [
        {
            "applicationName": "WordPress_App",
            "status": "Failed",
            "deploymentOverview": {
                "Failed": 0,
                "InProgress": 0,
                "Skipped": 0,
                "Succeeded": 1,
                "Pending": 0
            },
            "deploymentConfigName": "CodeDeployDefault.OneAtATime",
            "creator": "user",
            "deploymentGroupName": "WordPress_DG",
            "revision": {
                "revisionType": "S3",
                "s3Location": {
                "bundleType": "zip",
                "version": "uTecLusEXAMPLEFXtfUcyfV8bEXAMPLE",
                "bucket": "amzn-s3-demo-bucket",
                "key": "WordPressApp.zip"
                }
            },
            "deploymentId": "d-A1B2C3111",
            "createTime": 1408480721.9,
            "completeTime": 1408480741.822
        },
        {
            "applicationName": "MyOther_App",
            "status": "Failed",
            "deploymentOverview": {
                "Failed": 1,
                "InProgress": 0,
                "Skipped": 0,
                "Succeeded": 0,
                "Pending": 0
            },
            "deploymentConfigName": "CodeDeployDefault.OneAtATime",
            "creator": "user",
            "errorInformation": {
                "message": "Deployment failed: Constraint default violated: No hosts
 succeeded.",
```

```
                "code": "HEALTH_CONSTRAINTS"
            },
            "deploymentGroupName": "MyOther_DG",
            "revision": {
                "revisionType": "S3",
                "s3Location": {
                "bundleType": "zip",
                "eTag": "\"dd56cfdEXAMPLE8e768f9d77fEXAMPLE\"",
                "bucket": "amzn-s3-demo-bucket",
                "key": "MyOtherApp.zip"
                }
            },
            "deploymentId": "d-A1B2C3222",
            "createTime": 1409764576.589,
            "completeTime": 1409764596.101
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetDeployments。

batch-get-on-premises-instances

以下程式碼範例顯示如何使用 batch-get-on-premises-instances。

AWS CLI

取得一或多個內部部署執行個體的相關資訊

下列batch-get-on-premises-instances範例取得兩個現場部署執行個體的相關資訊。

```
aws deploy batch-get-on-premises-instances --instance-
names AssetTag12010298EX AssetTag23121309EX
```

輸出：

```
{
    "instanceInfos": [
        {
            "iamUserArn": "arn:aws:iam::123456789012:user/AWS/CodeDeploy/
AssetTag12010298EX",
            "tags": [
                {
```

```
                    "Value": "CodeDeployDemo-OnPrem",
                    "Key": "Name"
                }
            ],
            "instanceName": "AssetTag12010298EX",
            "registerTime": 1425579465.228,
            "instanceArn": "arn:aws:codedeploy:us-west-2:123456789012:instance/
AssetTag12010298EX_4IwLNI2Alh"
        },
        {
            "iamUserArn": "arn:aws:iam::123456789012:user/AWS/CodeDeploy/
AssetTag23121309EX",
            "tags": [
                {
                    "Value": "CodeDeployDemo-OnPrem",
                    "Key": "Name"
                }
            ],
            "instanceName": "AssetTag23121309EX",
            "registerTime": 1425595585.988,
            "instanceArn": "arn:aws:codedeploy:us-west-2:80398EXAMPLE:instance/
AssetTag23121309EX_PomUy64Was"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetOnPremisesInstances。

continue-deployment

以下程式碼範例顯示如何使用 continue-deployment。

AWS CLI

開始重新路由流量，無需等待指定的等待時間。

下列continue-deployment範例會開始重新路由來自原始環境中執行個體的流量，這些執行個體已準備好開始將流量轉移到替代環境中的執行個體。

```
aws deploy continue-deployment \
    --deployment-id "d-A1B2C3111" \
    --deployment-wait-type "READY_WAIT"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 CodeDeploy API 參考中的 ContinueDeploymentAWS CodeDeploy。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ContinueDeployment。

`create-application`

以下程式碼範例顯示如何使用 `create-application`。

AWS CLI

建立應用程式

下列`create-application`範例會建立應用程式，並將其與使用者帳戶建立關聯 AWS 。

```
aws deploy create-application --application-name MyOther_App
```

輸出：

```
{
    "applicationId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateApplication。

`create-deployment-config`

以下程式碼範例顯示如何使用 `create-deployment-config`。

AWS CLI

建立自訂部署組態

下列`create-deployment-config`範例會建立自訂部署組態，並將其與使用者帳戶建立關聯 AWS 。

```
aws deploy create-deployment-config \
    --deployment-config-name ThreeQuartersHealthy \
    --minimum-healthy-hosts type=FLEET_PERCENT,value=75
```

輸出：

```
{
    "deploymentConfigId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDeploymentConfig。

create-deployment-group

以下程式碼範例顯示如何使用 create-deployment-group。

AWS CLI

建立部署群組

下列create-deployment-group範例會建立部署群組，並將其與指定的應用程式和使用者帳戶建立關聯 AWS 。

```
aws deploy create-deployment-group \
    --application-name WordPress_App \
    --auto-scaling-groups CodeDeployDemo-ASG \
    --deployment-config-name CodeDeployDefault.OneAtATime \
    --deployment-group-name WordPress_DG \
    --ec2-tag-filters Key=Name,Value=CodeDeployDemo,Type=KEY_AND_VALUE \
    --service-role-arn arn:aws:iam::123456789012:role/CodeDeployDemoRole
```

輸出：

```
{
    "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDeploymentGroup。

create-deployment

以下程式碼範例顯示如何使用 create-deployment。

AWS CLI

範例 1：使用 EC2/內部部署運算平台建立 CodeDeploy 部署

下列create-deployment範例會建立部署，並將其與使用者帳戶建立關聯 AWS 。

```
aws deploy create-deployment \
    --application-name WordPress_App \
    --deployment-config-name CodeDeployDefault.OneAtATime \
    --deployment-group-name WordPress_DG \
    --description "My demo deployment" \
    --s3-location bucket=amzn-s3-demo-
bucket,bundleType=zip,eTag=dd56cfdEXAMPLE8e768f9d77fEXAMPLE,key=WordPressApp.zip
```

輸出：

```
{
    "deploymentId": "d-A1B2C3111"
}
```

範例 2：使用 Amazon ECS 運算平台建立 CodeDeploy 部署

下列create-deployment範例使用下列兩個檔案來部署 Amazon ECS 服務。

create-deployment.json 檔案的內容：

```
{
    "applicationName": "ecs-deployment",
    "deploymentGroupName": "ecs-deployment-dg",
    "revision": {
        "revisionType": "S3",
        "s3Location": {
            "bucket": "ecs-deployment-bucket",
            "key": "appspec.yaml",
            "bundleType": "YAML"
        }
    }
}
```

該檔案接著appspec.yaml會從名為 的 S3 儲存貯體擷取下列檔案ecs-deployment-bucket。

```
version: 0.0
Resources:
  - TargetService:
      Type: AWS::ECS::Service
      Properties:
```

```
        TaskDefinition: "arn:aws:ecs:region:123456789012:task-definition/ecs-task-
def:2"
        LoadBalancerInfo:
          ContainerName: "sample-app"
          ContainerPort: 80
        PlatformVersion: "LATEST"
```

命令：

```
aws deploy create-deployment \
    --cli-input-json file://create-deployment.json \
    --region us-east-1
```

輸出：

```
{
    "deploymentId": "d-1234ABCDE"
}
```

如需詳細資訊，請參閱 CodeDeploy API 參考中的 CreateDeploymentAWS CodeDeploy。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDeployment。

delete-application

以下程式碼範例顯示如何使用 delete-application。

AWS CLI

如欲刪除應用程式

下列delete-application範例會刪除與使用者帳戶相關聯的指定應用程式 AWS 。

```
aws deploy delete-application --application-name WordPress_App
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteApplication。

delete-deployment-config

以下程式碼範例顯示如何使用 delete-deployment-config。

AWS CLI

刪除部署組態

下列delete-deployment-config範例會刪除與使用者帳戶相關聯的自訂部署組態 AWS 。

```
aws deploy delete-deployment-config --deployment-config-name ThreeQuartersHealthy
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDeploymentConfig。

delete-deployment-group

以下程式碼範例顯示如何使用 delete-deployment-group。

AWS CLI

刪除部署群組

下列delete-deployment-group範例會刪除與指定應用程式相關聯的部署群組。

```
aws deploy delete-deployment-group \
    --application-name WordPress_App \
    --deployment-group-name WordPress_DG
```

輸出：

```
{
    "hooksNotCleanedUp": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDeploymentGroup。

delete-git-hub-account-token

以下程式碼範例顯示如何使用 delete-git-hub-account-token。

AWS CLI

刪除 GitHub 帳戶連線

下列delete-git-hub-account-token範例會刪除指定 GitHub 帳戶的連線。

```
aws deploy delete-git-hub-account-token --token-name my-github-account
```

輸出：

```
{
    "tokenName": "my-github-account"
}
```

如需詳細資訊，請參閱 AWS CodeDeploy API 參考中的 DeleteGitHubAccountToken。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteGitHubAccountToken。

deregister-on-premises-instance

以下程式碼範例顯示如何使用 deregister-on-premises-instance。

AWS CLI

取消註冊現場部署執行個體

下列deregister-on-premises-instance範例會使用 AWS CodeDeploy 取消註冊現場部署執行個體，但不會刪除與執行個體相關聯的 IAM 使用者，也不會取消在 AWS CodeDeploy 中與執行個體的現場部署執行個體標籤的關聯。它也不會從執行個體解除安裝 AWS CodeDeploy 代理程式，也不會從執行個體移除內部部署組態檔案。

```
aws deploy deregister-on-premises-instance --instance-name AssetTag12010298EX
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterOnPremisesInstance。

deregister

以下程式碼範例顯示如何使用 deregister。

AWS CLI

取消註冊現場部署執行個體

下列deregister範例會使用 AWS CodeDeploy 取消註冊現場部署執行個體。它不會刪除與執行個體相關聯的 IAM 使用者。它會取消 in AWS CodeDeploy 內部部署標籤與執行個體的關聯。它不會從執行個體解除安裝 AWS CodeDeploy Agent，也不會從執行個體中移除內部部署組態檔案。

```
aws deploy deregister \
    --instance-name AssetTag12010298EX \
    --no-delete-iam-user \
    --region us-west-2
```

輸出：

```
Retrieving on-premises instance information... DONE
IamUserArn: arn:aws:iam::80398EXAMPLE:user/AWS/CodeDeploy/AssetTag12010298EX
Tags: Key=Name,Value=CodeDeployDemo-OnPrem
Removing tags from the on-premises instance... DONE
Deregistering the on-premises instance... DONE
Run the following command on the on-premises instance to uninstall the codedeploy-
agent:
aws deploy uninstall
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的取消註冊。

get-application-revision

以下程式碼範例顯示如何使用 get-application-revision。

AWS CLI

取得應用程式修訂的相關資訊

下列get-application-revision範例顯示與指定應用程式相關聯之應用程式修訂版的相關資訊。

```
aws deploy get-application-revision \
    --application-name WordPress_App \
    --s3-location bucket=amzn-s3-demo-
bucket,bundleType=zip,eTag=dd56cfdEXAMPLE8e768f9d77fEXAMPLE,key=WordPressApp.zip
```

輸出：

```
{
```

```
    "applicationName": "WordPress_App",
    "revisionInfo": {
        "description": "Application revision registered by Deployment ID: d-
A1B2C3111",
        "registerTime": 1411076520.009,
        "deploymentGroups": "WordPress_DG",
        "lastUsedTime": 1411076520.009,
        "firstUsedTime": 1411076520.009
    },
    "revision": {
        "revisionType": "S3",
        "s3Location": {
            "bundleType": "zip",
            "eTag": "dd56cfdEXAMPLE8e768f9d77fEXAMPLE",
            "bucket": "amzn-s3-demo-bucket",
            "key": "WordPressApp.zip"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetApplicationRevision。

get-application

以下程式碼範例顯示如何使用 get-application。

AWS CLI

取得應用程式的相關資訊

下列get-application範例顯示與使用者帳戶相關聯之應用程式的相關資訊 AWS 。

```
aws deploy get-application --application-name WordPress_App
```

輸出：

```
{
    "application": {
        "applicationName": "WordPress_App",
        "applicationId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "createTime": 1407878168.078,
        "linkedToGitHub": false
```

```
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetApplication。

get-deployment-config

以下程式碼範例顯示如何使用 get-deployment-config。

AWS CLI

取得部署組態的相關資訊

下列get-deployment-config範例顯示與使用者帳戶相關聯之部署組態的相關資訊 AWS 。

```
aws deploy get-deployment-config --deployment-config-name ThreeQuartersHealthy
```

輸出：

```
{
    "deploymentConfigInfo": {
        "deploymentConfigId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "minimumHealthyHosts": {
            "type": "FLEET_PERCENT",
            "value": 75
        },
        "createTime": 1411081164.379,
        "deploymentConfigName": "ThreeQuartersHealthy"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDeploymentConfig。

get-deployment-group

以下程式碼範例顯示如何使用 get-deployment-group。

AWS CLI

檢視部署群組的相關資訊

下列get-deployment-group範例顯示與指定應用程式相關聯之部署群組的相關資訊。

```
aws deploy get-deployment-group \
    --application-name WordPress_App \
    --deployment-group-name WordPress_DG
```

輸出：

```
{
    "deploymentGroupInfo": {
        "applicationName": "WordPress_App",
        "autoScalingGroups": [
            "CodeDeployDemo-ASG"
        ],
        "deploymentConfigName": "CodeDeployDefault.OneAtATime",
        "ec2TagFilters": [
            {
                "Type": "KEY_AND_VALUE",
                "Value": "CodeDeployDemo",
                "Key": "Name"
            }
        ],
        "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "serviceRoleArn": "arn:aws:iam::123456789012:role/CodeDeployDemoRole",
        "deploymentGroupName": "WordPress_DG"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDeploymentGroup。

get-deployment-instance

以下程式碼範例顯示如何使用 get-deployment-instance。

AWS CLI

取得部署執行個體的相關資訊

下列get-deployment-instance範例顯示與指定部署相關聯的部署執行個體相關資訊。

```
aws deploy get-deployment-instance --deployment-id d-QA4G4F9EX --instance-
id i-902e9fEX
```

輸出：

```
{
    "instanceSummary": {
        "instanceId": "arn:aws:ec2:us-east-1:80398EXAMPLE:instance/i-902e9fEX",
        "lifecycleEvents": [
            {
                "status": "Succeeded",
                "endTime": 1408480726.569,
                "startTime": 1408480726.437,
                "lifecycleEventName": "ApplicationStop"
            },
            {
                "status": "Succeeded",
                "endTime": 1408480728.016,
                "startTime": 1408480727.665,
                "lifecycleEventName": "DownloadBundle"
            },
            {
                "status": "Succeeded",
                "endTime": 1408480729.744,
                "startTime": 1408480729.125,
                "lifecycleEventName": "BeforeInstall"
            },
            {
                "status": "Succeeded",
                "endTime": 1408480730.979,
                "startTime": 1408480730.844,
                "lifecycleEventName": "Install"
            },
            {
                "status": "Failed",
                "endTime": 1408480732.603,
                "startTime": 1408480732.1,
                "lifecycleEventName": "AfterInstall"
            },
            {
                "status": "Skipped",
                "endTime": 1408480732.606,
                "lifecycleEventName": "ApplicationStart"
            },
            {
                "status": "Skipped",
                "endTime": 1408480732.606,
```

```
                "lifecycleEventName": "ValidateService"
            }
        ],
        "deploymentId": "d-QA4G4F9EX",
        "lastUpdatedAt": 1408480733.152,
        "status": "Failed"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDeploymentInstance。

get-deployment-target

以下程式碼範例顯示如何使用 get-deployment-target。

AWS CLI

傳回部署目標的相關資訊

下列get-deployment-target範例會傳回與指定部署相關聯之部署目標的相關資訊。

```
aws deploy get-deployment-target \
    --deployment-id "d-A1B2C3111" \
    --target-id "i-a1b2c3d4e5f611111"
```

輸出：

```
{
    "deploymentTarget": {
        "deploymentTargetType": "InstanceTarget",
        "instanceTarget": {
            "lastUpdatedAt": 1556918687.504,
            "targetId": "i-a1b2c3d4e5f611111",
            "targetArn": "arn:aws:ec2:us-west-2:123456789012:instance/i-
a1b2c3d4e5f611111",
            "status": "Succeeded",
            "lifecycleEvents": [
                {
                    "status": "Succeeded",
                    "diagnostics": {
                        "errorCode": "Success",
                        "message": "Succeeded",
                        "logTail": "",
```

```
                "scriptName": ""
            },
            "lifecycleEventName": "ApplicationStop",
            "startTime": 1556918592.162,
            "endTime": 1556918592.247
        },
        {
            "status": "Succeeded",
            "diagnostics": {
                "errorCode": "Success",
                "message": "Succeeded",
                "logTail": "",
                "scriptName": ""
            },
            "lifecycleEventName": "DownloadBundle",
            "startTime": 1556918593.193,
            "endTime": 1556918593.981
        },
        {
            "status": "Succeeded",
            "diagnostics": {
                "errorCode": "Success",
                "message": "Succeeded",
                "logTail": "",
                "scriptName": ""
            },
            "lifecycleEventName": "BeforeInstall",
            "startTime": 1556918594.805,
            "endTime": 1556918681.807
        },
        {
            "status": "Succeeded",
            "diagnostics": {
                "errorCode": "Success",
                "message": "Succeeded",
                "logTail": "",
                "scriptName": ""
            },
            "lifecycleEventName": "Install",
            "startTime": 1556918682.696,
            "endTime": 1556918683.005
        },
        {
            "status": "Succeeded",
```

```
                    "diagnostics": {
                        "errorCode": "Success",
                        "message": "Succeeded",
                        "logTail": "",
                        "scriptName": ""
                    },
                    "lifecycleEventName": "AfterInstall",
                    "startTime": 1556918684.135,
                    "endTime": 1556918684.216
                },
                {
                    "status": "Succeeded",
                    "diagnostics": {
                        "errorCode": "Success",
                        "message": "Succeeded",
                        "logTail": "",
                        "scriptName": ""
                    },
                    "lifecycleEventName": "ApplicationStart",
                    "startTime": 1556918685.211,
                    "endTime": 1556918685.295
                },
                {
                    "status": "Succeeded",
                    "diagnostics": {
                        "errorCode": "Success",
                        "message": "Succeeded",
                        "logTail": "",
                        "scriptName": ""
                    },
                    "lifecycleEventName": "ValidateService",
                    "startTime": 1556918686.65,
                    "endTime": 1556918686.747
                }
            ],
            "deploymentId": "d-A1B2C3111"
        }
    }
}
```

如需詳細資訊，請參閱 CodeDeploy API 參考中的 GetDeploymentTarget。 AWS CodeDeploy

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDeploymentTarget。

get-deployment

以下程式碼範例顯示如何使用 get-deployment。

AWS CLI

取得部署的相關資訊

下列get-deployment範例顯示與使用者帳戶相關聯的部署相關資訊 AWS 。

```
aws deploy get-deployment --deployment-id d-A1B2C3123
```

輸出：

```
{
    "deploymentInfo": {
        "applicationName": "WordPress_App",
        "status": "Succeeded",
        "deploymentOverview": {
            "Failed": 0,
            "InProgress": 0,
            "Skipped": 0,
            "Succeeded": 1,
            "Pending": 0
        },
        "deploymentConfigName": "CodeDeployDefault.OneAtATime",
        "creator": "user",
        "description": "My WordPress app deployment",
        "revision": {
            "revisionType": "S3",
            "s3Location":  {
            "bundleType": "zip",
            "eTag": "\"dd56cfdEXAMPLE8e768f9d77fEXAMPLE\"",
            "bucket": "amzn-s3-demo-bucket",
            "key": "WordPressApp.zip"
            }
        },
        "deploymentId": "d-A1B2C3123",
        "deploymentGroupName": "WordPress_DG",
        "createTime": 1409764576.589,
        "completeTime": 1409764596.101,
        "ignoreApplicationStopFailures": false
    }
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDeployment。

get-on-premises-instance

以下程式碼範例顯示如何使用 get-on-premises-instance。

AWS CLI

取得內部部署執行個體的相關資訊

下列get-on-premises-instance範例會擷取指定內部部署執行個體的相關資訊。

```
aws deploy get-on-premises-instance --instance-name AssetTag12010298EX
```

輸出：

```
{
    "instanceInfo": {
    "iamUserArn": "arn:aws:iam::123456789012:user/AWS/CodeDeploy/
AssetTag12010298EX",
        "tags": [
        {
            "Value": "CodeDeployDemo-OnPrem",
            "Key": "Name"
        }
        ],
        "instanceName": "AssetTag12010298EX",
        "registerTime": 1425579465.228,
        "instanceArn": "arn:aws:codedeploy:us-east-1:123456789012:instance/
AssetTag12010298EX_4IwLNI2Alh"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetOnPremisesInstance。

install

以下程式碼範例顯示如何使用 install。

AWS CLI

安裝現場部署執行個體

下列install範例會將現場部署組態檔案從執行個體上的指定位置複製到 AWS CodeDeploy 代理程式預期找到的執行個體上的位置。它也會在執行個體上安裝 AWS CodeDeploy 代理程式。它不會建立任何 IAM 使用者，也不會向 AWS CodeDeploy 註冊現場部署執行個體，也不會為執行個體關聯 AWS CodeDeploy 中的任何現場部署執行個體標籤。

```
aws deploy install \
    --override-config \
    --config-file C:\temp\codedeploy.onpremises.yml \
    --region us-west-2 \
    --agent-installer s3://aws-codedeploy-us-west-2/latest/codedeploy-agent.msi
```

輸出：

```
Creating the on-premises instance configuration file... DONE
Installing the AWS CodeDeploy Agent... DONE
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的安裝。

list-application-revisions

以下程式碼範例顯示如何使用 list-application-revisions。

AWS CLI

取得應用程式修訂的相關資訊

下列list-application-revisions範例顯示與指定應用程式相關聯的所有應用程式修訂的相關資訊。

```
aws deploy list-application-revisions \
    --application-name WordPress_App \
    --s-3-bucket amzn-s3-demo-bucket \
    --deployed exclude \
    --s-3-key-prefix WordPress_ \
    --sort-by lastUsedTime \
    --sort-order descending
```

輸出：

```
{
    "revisions": [
        {
            "revisionType": "S3",
            "s3Location": {
                "version": "uTecLusvCB_JqHFXtfUcyfV8bEXAMPLE",
                "bucket": "amzn-s3-demo-bucket",
                "key": "WordPress_App.zip",
                "bundleType": "zip"
            }
        },
        {
            "revisionType": "S3",
            "s3Location": {
                "version": "tMk.UxgDpMEVb7V187ZM6wVAWEXAMPLE",
                "bucket": "amzn-s3-demo-bucket",
                "key": "WordPress_App_2-0.zip",
                "bundleType": "zip"
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListApplicationRevisions。

list-applications

以下程式碼範例顯示如何使用 list-applications。

AWS CLI

取得應用程式的相關資訊

下列list-applications範例顯示與使用者帳戶相關聯的所有應用程式的相關資訊 AWS 。

```
aws deploy list-applications
```

輸出：

```
{
```

```
    "applications": [
        "WordPress_App",
        "MyOther_App"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListApplications。

`list-deployment-configs`

以下程式碼範例顯示如何使用 `list-deployment-configs`。

AWS CLI

取得部署組態的相關資訊

下列`list-deployment-configs`範例顯示與使用者帳戶相關聯的所有部署組態的相關資訊 AWS 。

```
aws deploy list-deployment-configs
```

輸出：

```
{
    "deploymentConfigsList": [
        "ThreeQuartersHealthy",
        "CodeDeployDefault.AllAtOnce",
        "CodeDeployDefault.HalfAtATime",
        "CodeDeployDefault.OneAtATime"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDeploymentConfigs。

`list-deployment-groups`

以下程式碼範例顯示如何使用 `list-deployment-groups`。

AWS CLI

取得部署群組的相關資訊

下列list-deployment-groups範例顯示與指定應用程式相關聯的所有部署群組的相關資訊。

```
aws deploy list-deployment-groups --application-name WordPress_App
```

輸出：

```
{
    "applicationName": "WordPress_App",
    "deploymentGroups": [
        "WordPress_DG",
        "WordPress_Beta_DG"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDeploymentGroups。

list-deployment-instances

以下程式碼範例顯示如何使用 list-deployment-instances。

AWS CLI

取得部署執行個體的相關資訊

下列list-deployment-instances範例顯示與指定部署相關聯的所有部署執行個體的相關資訊。

```
aws deploy list-deployment-instances \
    --deployment-id d-A1B2C3111 \
    --instance-status-filter Succeeded
```

輸出：

```
{
    "instancesList": [
        "i-EXAMPLE11",
        "i-EXAMPLE22"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDeploymentInstances。

`list-deployment-targets`

以下程式碼範例顯示如何使用 `list-deployment-targets`。

AWS CLI

擷取與部署相關聯的目標 IDs 清單

下列`list-deployment-targets`範例會擷取與狀態為 "Failed" 或 "InProgress" 之部署相關聯的目標 IDs 清單。InProgress."

```
aws deploy list-deployment-targets \
    --deployment-id "d-A1B2C3111" \
    --target-filters "{\"TargetStatus\":[\"Failed\",\"InProgress\"]}"
```

輸出：

```
{
    "targetIds": [
        "i-0f1558aaf90e5f1f9"
    ]
}
```

如需詳細資訊，請參閱 CodeDeploy API 參考中的 ListDeploymentTargets。 AWS CodeDeploy

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDeploymentTargets。

`list-deployments`

以下程式碼範例顯示如何使用 `list-deployments`。

AWS CLI

取得部署的相關資訊

下列`list-deployments`範例顯示與指定應用程式和部署群組相關聯的所有部署的相關資訊。

```
aws deploy list-deployments \
    --application-name WordPress_App \
    --create-time-range start=2014-08-19T00:00:00,end=2014-08-20T00:00:00 \
    --deployment-group-name WordPress_DG \
    --include-only-statuses Failed
```

輸出：

```
{
    "deployments": [
        "d-EXAMPLE11",
        "d-EXAMPLE22",
        "d-EXAMPLE33"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDeployments。

list-git-hub-account-token-names

以下程式碼範例顯示如何使用 list-git-hub-account-token-names。

AWS CLI

列出與 GitHub 帳戶之預存連線的名稱

下列list-git-hub-account-token-names範例列出目前 AWS 使用者與 GitHub 帳戶之間儲存連線的名稱。

```
aws deploy list-git-hub-account-token-names
```

輸出：

```
{
    "tokenNameList": [
        "my-first-token",
        "my-second-token",
        "my-third-token"
    ]
}
```

如需詳細資訊，請參閱 AWS CodeDeploy API 參考中的 ListGitHubAccountTokenNames。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListGitHubAccountTokenNames。

list-on-premises-instances

以下程式碼範例顯示如何使用 list-on-premises-instances。

AWS CLI

取得一或多個內部部署執行個體的相關資訊

下列`list-on-premises-instances`範例會擷取在 AWS CodeDeploy 中註冊之執行個體的可用現場部署執行個體名稱清單，以及具有在 AWS CodeDeploy 中與執行個體相關聯的指定現場部署執行個體標籤。

```
aws deploy list-on-premises-instances \
    --registration-status Registered \
    --tag-filters Key=Name,Value=CodeDeployDemo-OnPrem,Type=KEY_AND_VALUE
```

輸出：

```
{
    "instanceNames": [
        "AssetTag12010298EX"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListOnPremisesInstances。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出資源的標籤（應用程式）

下列`list-tags-for-resource`範例列出套用至 CodeDeploy 中名為 testApp 之應用程式的標籤。

```
aws deploy list-tags-for-resource \
    --resource-arn arn:aws:codedeploy:us-west-2:111122223333:application:testApp
```

輸出：

```
{
    "Tags": [
```

```
        {
            "Key": "Type",
            "Value": "testType"
        },
        {
            "Key": "Name",
            "Value": "testName"
        }
    ]
}
```

如需詳細資訊，請參閱《CodeDeploy 使用者指南》中的在 CodeDeploy 中標記部署群組的執行個體。 AWS CodeDeploy

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

push

以下程式碼範例顯示如何使用 push。

AWS CLI

綁定和部署相容於 AWS CodeDeploy 的應用程式修訂版至 Amazon S3

下列push範例會封裝應用程式修訂版並將其部署至 Amazon S3，然後將應用程式修訂版與指定的應用程式建立關聯。

```
aws deploy push \
    --application-name WordPress_App \
    --description "This is my deployment" \
    --ignore-hidden-files \
    --s3-location s3://amzn-s3-demo-bucket/WordPressApp.zip \
    --source /tmp/MyLocalDeploymentFolder/
```

輸出說明如何使用 create-deployment命令來建立使用上傳應用程式修訂版的部署。

```
To deploy with this revision, run:
aws deploy create-deployment --application-name WordPress_App --
deployment-config-name <deployment-config-name> --deployment-group-
name <deployment-group-name> --s3-location bucket=amzn-s3-demo-
bucket,key=WordPressApp.zip,bundleType=zip,eTag="cecc9b8EXAMPLE50a6e71fdb88EXAMPLE",version=
```

- 如需 API 詳細資訊，請參閱推送AWS CLI 命令參考。

register-application-revision

以下程式碼範例顯示如何使用 register-application-revision。

AWS CLI

註冊已上傳應用程式修訂版的相關資訊

下列register-application-revision範例會註冊儲存在 Amazon S3 with AWS CodeDeploy 中已上傳應用程式修訂版的相關資訊。

```
aws deploy register-application-revision \
    --application-name WordPress_App \
    --description "Revised WordPress application" \
    --s3-location bucket=amzn-s3-demo-
bucket,key=RevisedWordPressApp.zip,bundleType=zip,eTag=cecc9b8a08eac650a6e71fdb88EXAMPLE
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterApplicationRevision。

register-on-premises-instance

以下程式碼範例顯示如何使用 register-on-premises-instance。

AWS CLI

註冊現場部署執行個體

下列register-on-premises-instance範例向 AWS CodeDeploy 註冊現場部署執行個體。它不會建立指定的 IAM 使用者，也不會在 AWS CodeDeploy 中建立任何現場部署執行個體標籤與已註冊執行個體的關聯。

```
aws deploy register-on-premises-instance \
    --instance-name AssetTag12010298EX \
    --iam-user-arn arn:aws:iam::80398EXAMPLE:user/CodeDeployDemoUser-OnPrem
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterOnPremisesInstance。

register

以下程式碼範例顯示如何使用 register。

AWS CLI

註冊現場部署執行個體

下列register範例會向 AWS CodeDeploy 註冊現場部署執行個體、將 AWS CodeDeploy現場部署執行個體標籤與已註冊的執行個體建立關聯，以及建立可複製到執行個體的現場部署組態檔案。它不會建立 IAM 使用者，也不會在執行個體上安裝 AWS CodeDeploy Agent。

```
aws deploy register \
    --instance-name AssetTag12010298EX \
    --iam-user-arn arn:aws:iam::80398EXAMPLE:user/CodeDeployUser-OnPrem \
    --tags Key=Name,Value=CodeDeployDemo-OnPrem \
    --region us-west-2
```

輸出：

```
Registering the on-premises instance... DONE
Adding tags to the on-premises instance... DONE
Copy the on-premises configuration file named codedeploy.onpremises.yml to the on-
premises instance, and run the following command on the on-premises instance to
 install and configure the AWS CodeDeploy Agent:
aws deploy install --config-file codedeploy.onpremises.yml
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的註冊。

remove-tags-from-on-premises-instances

以下程式碼範例顯示如何使用 remove-tags-from-on-premises-instances。

AWS CLI

從一或多個內部部署執行個體移除標籤

下列remove-tags-from-on-premises-instances範例會取消 AWS CodeDeploy 中指定的內部部署標籤與內部部署執行個體的關聯。它不會在 AWS CodeDeploy 中取消註冊現場部署執行個體，也不會從執行個體解除安裝 AWS CodeDeploy 代理程式，也不會從執行個體中移除現場部署組態檔案，也不會刪除與執行個體相關聯的 IAM 使用者。

```
aws deploy remove-tags-from-on-premises-instances \
    --instance-names AssetTag12010298EX AssetTag23121309EX \
    --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveTagsFromOnPremisesInstances。

stop-deployment

以下程式碼範例顯示如何使用 stop-deployment。

AWS CLI

嘗試停止部署

下列stop-deployment範例會嘗試停止與使用者帳戶相關聯的進行中部署 AWS 。

aws 部署 stop-deployment --deployment-id d-A1B2C3111

輸出：

```
{
    "status": "Succeeded",
    "statusMessage": "No more commands will be scheduled for execution in the
 deployment instances"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopDeployment。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

標記 resoure （ 應用程式 ）

下列tag-resource範例會將兩個具有索引鍵名稱和類型的標籤，以及 testName 和 testType 值新增至 CodeDeploy 中名為 testApp 的應用程式： CodeDeploy.:

```
aws deploy tag-resource \
    --resource-arn  arn:aws:codedeploy:us-west-2:111122223333:application:testApp \
    --tags Key=Name,Value=testName Key=Type,Value=testType
```

如果成功，此命令不會產生輸出。

如需詳細資訊，請參閱《CodeDeploy 使用者指南》中的在 CodeDeploy 中標記部署群組的執行個體。 AWS CodeDeploy

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

`uninstall`

以下程式碼範例顯示如何使用 `uninstall`。

AWS CLI

解除安裝現場部署執行個體

下列`uninstall`範例會從現場部署執行個體解除安裝 AWS CodeDeploy Agent，並從執行個體中移除現場部署組態檔案。它不會取消註冊執行個體 in AWS CodeDeploy，也不會取消任何內部部署執行個體標籤 in AWS CodeDeploy 與執行個體的關聯，也不會刪除與執行個體相關聯的 IAM 使用者。

```
aws deploy uninstall
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的解除安裝。

`untag-resource`

以下程式碼範例顯示如何使用 `untag-resource`。

AWS CLI

從資源移除標籤（應用程式）

下列`untag-resource`範例會從 CodeDeploy 中名為 testApp 的應用程式中移除兩個具有索引鍵名稱和類型的標籤。

```
aws deploy untag-resource \
    --resource-arn  arn:aws:codedeploy:us-west-2:111122223333:application:testApp \
    --tag-keys Name Type
```

如果成功，此命令不會產生輸出。

如需詳細資訊，請參閱 CodeDeploy 使用者指南中的在 CodeDeploy 中標記部署群組的執行個體。AWS CodeDeploy

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-application

以下程式碼範例顯示如何使用 update-application。

AWS CLI

變更應用程式的詳細資訊

下列update-application範例會變更與使用者帳戶相關聯的應用程式名稱 AWS 。

```
aws deploy update-application \
    --application-name WordPress_App \
    --new-application-name My_WordPress_App
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateApplication。

update-deployment-group

以下程式碼範例顯示如何使用 update-deployment-group。

AWS CLI

變更部署群組的相關資訊

下列update-deployment-group範例會變更與指定應用程式相關聯的部署群組設定。

```
aws deploy update-deployment-group \
    --application-name WordPress_App \
    --auto-scaling-groups My_CodeDeployDemo_ASG \
```

```
    --current-deployment-group-name WordPress_DG \
    --deployment-config-name CodeDeployDefault.AllAtOnce \
    --ec2-tag-filters Key=Name,Type=KEY_AND_VALUE,Value=My_CodeDeployDemo \
    --new-deployment-group-name My_WordPress_DepGroup \
    --service-role-arn arn:aws:iam::80398EXAMPLE:role/CodeDeployDemo-2
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDeploymentGroup。

使用 的 CodeGuru Reviewer 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 CodeGuru Reviewer 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-repository`

以下程式碼範例顯示如何使用 `associate-repository`。

AWS CLI

範例 1：建立 Bitbucket 儲存庫關聯

下列`associate-repository`範例會使用現有的 Bitbucket 儲存庫建立儲存庫關聯。

```
aws codeguru-reviewer associate-repository \
    --repository 'Bitbucket={Owner=sample-owner, Name=mySampleRepo,
 ConnectionArn=arn:aws:codestar-connections:us-west-2:123456789012:connection/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 }'
```

輸出：

```
{
    "RepositoryAssociation": {
        "ProviderType": "Bitbucket",
        "Name": "mySampleRepo",
        "LastUpdatedTimeStamp": 1596216896.979,
        "AssociationId": "association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "CreatedTimeStamp": 1596216896.979,
        "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "State": "Associating",
        "StateReason": "Pending Repository Association",
        "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "Owner": "sample-owner"
    }
}
```

如需詳細資訊，請參閱《Amazon CodeGuru Reviewer 使用者指南》中的在 Amazon CodeGuru Reviewer 中建立 Bitbucket 儲存庫關聯。 CodeGuru

範例 2：建立 GitHub Enterprise 儲存庫關聯

下列`associate-repository`範例使用現有的 GitHub Enterprise 儲存庫建立儲存庫關聯。

```
aws codeguru-reviewer associate-repository \
    --repository 'GitHubEnterpriseServer={Owner=sample-owner, Name=mySampleRepo,
 ConnectionArn=arn:aws:codestar-connections:us-west-2:123456789012:connection/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 }'
```

輸出：

```
{
    "RepositoryAssociation": {
        "ProviderType": "GitHubEnterpriseServer",
        "Name": "mySampleRepo",
        "LastUpdatedTimeStamp": 1596216896.979,
        "AssociationId": "association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "CreatedTimeStamp": 1596216896.979,
        "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "State": "Associating",
        "StateReason": "Pending Repository Association",
```

```
        "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "Owner": "sample-owner"
    }
}
```

如需詳細資訊，請參閱《Amazon Codeguru Reviewer 使用者指南》中的在 Amazon CodeGuru Reviewer 中建立 GitHub Enterprise Server 儲存庫關聯。

範例 3：建立 AWS CodeCommit 儲存庫關聯

下列associate-repository範例會使用現有的 AWS CodeCommit 儲存庫建立儲存庫關聯。

```
aws codeguru-reviewer associate-repository \
    --repository CodeCommit={Name=mySampleRepo}
```

輸出：

```
{
    "RepositoryAssociation": {
        "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Name": "My-ecs-beta-repo",
        "LastUpdatedTimeStamp": 1595634764.029,
        "ProviderType": "CodeCommit",
        "CreatedTimeStamp": 1595634764.029,
        "Owner": "544120495673",
        "State": "Associating",
        "StateReason": "Pending Repository Association",
        "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:544120495673:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
}
```

如需詳細資訊，請參閱《Amazon AWS CodeCommit CodeGuru Reviewer 使用者指南》中的在 Amazon CodeGuru Reviewer 中建立 CodeCommit 儲存庫關聯。 CodeGuru

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateRepository。

create-code-review

以下程式碼範例顯示如何使用 create-code-review。

AWS CLI

建立程式碼檢閱。

以下內容會在名為 的 AWS CodeCommit 儲存庫mainline分支中create-code-review建立程式碼檢閱my-repository-name。

```
aws codeguru-reviewer create-code-review \
    --name my-code-review \
    --repository-association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --type '{"RepositoryAnalysis": {"RepositoryHead": {"BranchName": "mainline"}}}'
```

輸出：

```
{
    "CodeReview": {
        "Name": "my-code-review",
        "CodeReviewArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222:code-
review:RepositoryAnalysis-my-code-review",
        "RepositoryName": "my-repository-name",
        "Owner": "123456789012",
        "ProviderType": "CodeCommit",
        "State": "Pending",
        "StateReason": "CodeGuru Reviewer has received the request, and a code
 review is scheduled.",
        "CreatedTimeStamp": 1618873489.195,
        "LastUpdatedTimeStamp": 1618873489.195,
        "Type": "RepositoryAnalysis",
        "SourceCodeType": {
            "RepositoryHead": {
                "BranchName": "mainline"
            }
        },
        "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
}
```

如需詳細資訊，請參閱《Amazon CodeGuru Reviewer 使用者指南》中的在 Amazon CodeGuru Reviewer 中建立程式碼檢閱 CodeGuru。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCodeReview。

describe-code-review

以下程式碼範例顯示如何使用 describe-code-review。

AWS CLI

列出程式碼檢閱的詳細資訊。

以下describe-code-review列出名為 "my-repo-name" 之 AWS CodeCommit 儲存庫的「主要」分支中程式碼檢閱的相關資訊。

```
aws codeguru-reviewer put-recommendation-feedback \
    --code-review-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefgh12345678 \
    --recommendation-
id 3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb \
    --reactions ThumbsUp
```

輸出

```
{
        "CodeReview": {
            "Name": "My-ecs-beta-repo-master-xs6di4kfd4j269dz",
            "CodeReviewArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222:code-
review:RepositoryAnalysis-my-repo-name",
            "RepositoryName": "My-ecs-beta-repo",
            "Owner": "123456789012",
            "ProviderType": "CodeCommit",
            "State": "Pending",
            "StateReason": "CodeGuru Reviewer is reviewing the source code.",
            "CreatedTimeStamp": 1618874226.226,
            "LastUpdatedTimeStamp": 1618874233.689,
            "Type": "RepositoryAnalysis",
            "SourceCodeType": {
                "RepositoryHead": {
                    "BranchName": "mainline"
                }
            },
```

```
            "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        }
    }
```

如需詳細資訊，請參閱《Amazon CodeGuru Reviewer 使用者指南》中的檢視程式碼檢閱詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCodeReview。

describe-recommendation-feedback

以下程式碼範例顯示如何使用 describe-recommendation-feedback。

AWS CLI

檢視有關建議意見回饋的資訊

以下describe-recommendation-feedback顯示有關建議意見回饋的資訊。此建議有一個ThumbsUp反應。

```
aws codeguru-reviewer describe-recommendation-feedback \
    --code-review-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefgh12345678 \
    --recommendation-
id 3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb
```

輸出：

```
{
    "RecommendationFeedback": {
        "CodeReviewArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefgh12345678",
        "RecommendationId":
 "3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb",
        "Reactions": [
            "ThumbsUp"
        ],
        "UserId": "aws-user-id",
        "CreatedTimeStamp": 1618877070.313,
        "LastUpdatedTimeStamp": 1618877948.881
```

```
    }
}
```

如需詳細資訊，請參閱《Amazon CodeGuru Reviewer 使用者指南》中的檢視建議並提供意見回饋和步驟 4：提供意見回饋。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRecommendationFeedback。

describe-repository-association

以下程式碼範例顯示如何使用 describe-repository-association。

AWS CLI

範例 1：傳回 GitHub 儲存庫關聯的相關資訊

下列describe-repository-association範例會傳回使用 GitHub Enterprise 儲存庫且處於 Associated 狀態之儲存庫關聯的相關資訊。

```
aws codeguru-reviewer describe-repository-association \
    --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "RepositoryAssociation": {
        "AssociationId": "b822717e-0711-4e8a-bada-0e738289c75e",
        "Name": "mySampleRepo",
        "LastUpdatedTimeStamp": 1588102637.649,
        "ProviderType": "GitHub",
        "CreatedTimeStamp": 1588102615.636,
        "Owner": "sample-owner",
        "State": "Associated",
        "StateReason": "Pull Request Notification configuration successful",
        "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
}
```

如需詳細資訊，請參閱《Amazon CodeGuru Reviewer 使用者指南》中的在 Amazon CodeGuru Reviewer 中建立 GitHub Enterprise Server 儲存庫關聯。 CodeGuru

範例 2：傳回失敗儲存庫關聯的相關資訊

下列describe-repository-association範例會傳回使用 GitHub Enterprise 儲存庫且處於 Failed 狀態之儲存庫關聯的相關資訊。

```
aws codeguru-reviewer describe-repository-association \
    --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "RepositoryAssociation": {
        "ProviderType": "GitHubEnterpriseServer",
        "Name": "mySampleRepo",
        "LastUpdatedTimeStamp": 1596217036.892,
        "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "CreatedTimeStamp": 1596216896.979,
        "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "State": "Failed",
        "StateReason": "Failed, Please retry.",
        "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
        "Owner": "sample-owner"
    }
}
```

如需詳細資訊，請參閱《Amazon CodeGuru Reviewer 使用者指南》中的在 Amazon CodeGuru Reviewer 中建立 GitHub Enterprise Server 儲存庫關聯。 CodeGuru

範例 3：傳回取消關聯儲存庫關聯的相關資訊

下列describe-repository-association範例會傳回使用 GitHub Enterprise 儲存庫且處於 Disassociating 狀態之儲存庫關聯的相關資訊。

```
aws codeguru-reviewer describe-repository-association \
    --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "RepositoryAssociation": {
        "ProviderType": "GitHubEnterpriseServer",
        "Name": "mySampleRepo",
        "LastUpdatedTimeStamp": 1596217036.892,
        "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "CreatedTimeStamp": 1596216896.979,
        "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "State": "Disassociating",
        "StateReason": "Source code access removal in progress",
        "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
        "Owner": "sample-owner"
    }
}
```

如需詳細資訊，請參閱《Amazon CodeGuru Reviewer 使用者指南》中的在 Amazon CodeGuru Reviewer 中建立 GitHub Enterprise Server 儲存庫關聯。 CodeGuru

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRepositoryAssociation。

`disassociate-repository`

以下程式碼範例顯示如何使用 `disassociate-repository`。

AWS CLI

取消與儲存庫關聯的關聯

下列會`disassociate-repository`取消與使用 AWS CodeCommit 儲存庫的儲存庫關聯。

```
aws codeguru-reviewer disassociate-repository \
    --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "RepositoryAssociation": {
        "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```
        "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Name": "my-repository",
        "Owner": "123456789012",
        "ProviderType": "CodeCommit",
        "State": "Disassociating",
        "LastUpdatedTimeStamp": 1618939174.759,
        "CreatedTimeStamp": 1595636947.096
    },
    "Tags": {
        "Status": "Secret",
        "Team": "Saanvi"
    }
}
```

如需詳細資訊，請參閱《Amazon CodeGuru Reviewer 使用者指南》中的取消 CodeGuru Reviewer 中的儲存庫關聯。 CodeGuru

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateRepository。

`list-code-reviews`

以下程式碼範例顯示如何使用 `list-code-reviews`。

AWS CLI

列出過去 90 天內在您的 AWS 帳戶中建立的程式碼檢閱。

下列`list-code-reviews`範例列出在過去 90 天內使用提取請求建立的程式碼檢閱。

```
aws codeguru-reviewer list-code-reviews \
    --type PullRequest
```

輸出：

```
{
    "CodeReviewSummaries": [
        {
            "LastUpdatedTimeStamp": 1588897288.054,
            "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "ProviderType": "GitHub",
            "PullRequestId": "5",
            "MetricsSummary": {
```

```
                "MeteredLinesOfCodeCount": 24,
                "FindingsCount": 1
            },
            "CreatedTimeStamp": 1588897068.512,
            "State": "Completed",
            "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Owner": "sample-owner",
            "RepositoryName": "sample-repository-name",
            "Type": "PullRequest"
        },
        {
            "LastUpdatedTimeStamp": 1588869793.263,
            "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "ProviderType": "GitHub",
            "PullRequestId": "4",
            "MetricsSummary": {
                "MeteredLinesOfCodeCount": 29,
                "FindingsCount": 0
            },
            "CreatedTimeStamp": 1588869575.949,
            "State": "Completed",
            "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "Owner": "sample-owner",
            "RepositoryName": "sample-repository-name",
            "Type": "PullRequest"
        },
        {
            "LastUpdatedTimeStamp": 1588870511.211,
            "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "ProviderType": "GitHub",
            "PullRequestId": "4",
            "MetricsSummary": {
                "MeteredLinesOfCodeCount": 2,
                "FindingsCount": 0
            },
            "CreatedTimeStamp": 1588870292.425,
            "State": "Completed",
            "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "Owner": "sample-owner",
            "RepositoryName": "sample-repository-name",
            "Type": "PullRequest"
```

```
        },
        {
            "LastUpdatedTimeStamp": 1588118522.452,
            "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
            "ProviderType": "GitHub",
            "PullRequestId": "3",
            "MetricsSummary": {
                "MeteredLinesOfCodeCount": 29,
                "FindingsCount": 0
            },
            "CreatedTimeStamp": 1588118301.131,
            "State": "Completed",
            "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
            "Owner": "sample-owner",
            "RepositoryName": "sample-repository-name",
            "Type": "PullRequest"
        },
        {
            "LastUpdatedTimeStamp": 1588112205.207,
            "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE55555",
            "ProviderType": "GitHub",
            "PullRequestId": "2",
            "MetricsSummary": {
                "MeteredLinesOfCodeCount": 25,
                "FindingsCount": 0
            },
            "CreatedTimeStamp": 1588111987.443,
            "State": "Completed",
            "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE55555",
            "Owner": "sample-owner",
            "RepositoryName": "sample-repository-name",
            "Type": "PullRequest"
        },
        {
            "LastUpdatedTimeStamp": 1588104489.981,
            "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE66666",
            "ProviderType": "GitHub",
            "PullRequestId": "1",
            "MetricsSummary": {
                "MeteredLinesOfCodeCount": 25,
                "FindingsCount": 0
            },
```

```
            "CreatedTimeStamp": 1588104270.223,
            "State": "Completed",
            "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE66666",
            "Owner": "sample-owner",
            "RepositoryName": "sample-repository-name",
            "Type": "PullRequest"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CodeGuru Reviewer 使用者指南》中的檢視所有程式碼檢閱。CodeGuru

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCodeReviews。

`list-recommendation-feedback`

以下程式碼範例顯示如何使用 `list-recommendation-feedback`。

AWS CLI

列出有關關聯儲存庫建議的客戶建議意見回饋

以下`list-recommendation-feedback`列出客戶對程式碼檢閱的所有建議的意見回饋。此程式碼檢閱有一個來自客戶的意見回饋：「ThumbsUp」。

```
aws codeguru-reviewer list-recommendation-feedback \
    --code-review-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefgh12345678
```

輸出：

```
{
    "RecommendationFeedbackSummaries": [
        {
            "RecommendationId":
 "3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb",
            "Reactions": [
                "ThumbsUp"
            ],
            "UserId": "aws-user-id"
```

```
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CodeGuru Reviewer 使用者指南》中的步驟 4：提供意見回饋。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRecommendationFeedback。

`list-recommendations`

以下程式碼範例顯示如何使用 `list-recommendations`。

AWS CLI

列出已完成程式碼檢閱的建議

下列`list-recommendations`範例列出已完成程式碼檢閱的建議。此程式碼檢閱有一個建議。

```
aws codeguru-reviewer list-recommendations \
    --code-review-arn arn:aws:codeguru-reviewer:us-west-2:544120495673:code-review:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "RecommendationSummaries": [
        {
            "Description": "\n\n**Problem**  \n You are using a `ConcurrentHashMap`, but your usage of `containsKey()` and `get()` may not be thread-safe at lines: **63 and 64**. In between the check and the `get()` another thread can remove the key and the `get()` will return `null`. The remove that can remove the key is at line: **59**.\n\n**Fix**  \n Consider calling `get()`, checking instead of your current check if the returned object is `null`, and then using that object only, without calling `get()` again.\n\n**More info**  \n [View an example on GitHub](https://github.com/apache/hadoop/blob/f16cf877e565084c66bc63605659b157c4394dc8/hadoop-tools/hadoop-aws/src/main/java/org/apache/hadoop/fs/s3a/s3guard/S3Guard.java#L302-L304) (external link).",
            "RecommendationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "StartLine": 63,
            "EndLine": 64,
            "FilePath": "src/main/java/com/company/sample/application/CreateOrderThread.java"
        }
```

```
    ]
}
```

如需詳細資訊，請參閱《Amazon CodeGuru Reviewer 使用者指南》中的步驟 4：提供意見回饋。CodeGuru

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRecommendations。

list-repository-associations

以下程式碼範例顯示如何使用 list-repository-associations。

AWS CLI

列出您 AWS 帳戶中的儲存庫關聯

下列list-repository-associations範例會傳回您帳戶中儲存庫關聯摘要物件的清單。您可以依 ProviderType、Name、 State和 篩選傳回的清單Owner。

```
aws codeguru-reviewer list-repository-associations
```

輸出：

```
{
    "RepositoryAssociationSummaries": [
        {
            "LastUpdatedTimeStamp": 1595886609.616,
            "Name": "test",
            "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Owner": "sample-owner",
            "State": "Associated",
            "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "ProviderType": "Bitbucket"
        },
        {
            "LastUpdatedTimeStamp": 1595636969.035,
            "Name": "CodeDeploy-CodePipeline-ECS-Tutorial",
            "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "Owner": "123456789012",
            "State": "Associated",
            "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
```

```
            "ProviderType": "CodeCommit"
        },
        {
            "LastUpdatedTimeStamp": 1595634785.983,
            "Name": "My-ecs-beta-repo",
            "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "Owner": "123456789012",
            "State": "Associated",
            "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "ProviderType": "CodeCommit"
        },
        {
            "LastUpdatedTimeStamp": 1590712811.77,
            "Name": "MyTestCodeCommit",
            "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
            "Owner": "123456789012",
            "State": "Associated",
            "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
            "ProviderType": "CodeCommit"
        },
        {
            "LastUpdatedTimeStamp": 1588102637.649,
            "Name": "aws-codeguru-profiler-sample-application",
            "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE55555",
            "Owner": "sample-owner",
            "State": "Associated",
            "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE55555",
            "ProviderType": "GitHub"
        },
        {
            "LastUpdatedTimeStamp": 1588028233.995,
            "Name": "codeguru-profiler-demo-app",
            "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE66666",
            "Owner": "sample-owner",
            "State": "Associated",
            "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE66666",
            "ProviderType": "GitHub"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon CodeGuru Reviewer 使用者指南》中的在 CodeGuru Reviewer 中檢視所有儲存庫關聯。 CodeGuru

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRepositoryAssociations。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出關聯儲存庫上的標籤

下列`list-tags-for-resource`列出相關聯儲存庫上的標籤。此關聯的儲存庫有兩個標籤。

```
aws codeguru-reviewer list-tags-for-resource \
    --resource-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "Tags": {
        "Status": "Secret",
        "Team": "Saanvi"
    }
}
```

如需詳細資訊，請參閱《Amazon CodeGuru Reviewer 使用者指南》中的檢視 CodeGuru Reviewer 關聯儲存庫 (AWS CLI) 的標籤。 CodeGuru

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`put-recommendation-feedback`

以下程式碼範例顯示如何使用 `put-recommendation-feedback`。

AWS CLI

將建議新增至程式碼檢閱

以下put-recommendation-feedback提供程式碼檢閱ThumbsUp的建議。

```
aws codeguru-reviewer put-recommendation-feedback \
    --code-review-arn \arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefgh12345678 \
    --recommendation-
id 3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb \
    --reactions ThumbsUp
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CodeGuru Reviewer 使用者指南》中的步驟 4：提供意見回饋。CodeGuru

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutRecommendationFeedback。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤新增至相關聯的儲存庫

下列會將兩個標籤tag-resource新增至相關聯的儲存庫

```
aws codeguru-reviewer tag-resource \
    --resource-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --tags Status=Secret,Team=Saanvi
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CodeGuru Reviewer 使用者指南》中的將標籤新增至 CodeGuru Reviewer 關聯儲存庫 (AWS CLI) 和新增或更新 CodeGuru Reviewer 關聯儲存庫 (CLI) 的標籤。CodeGuru AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

取消標記相關聯的儲存庫

以下內容會從相關聯的儲存庫untag-resource移除兩個具有金鑰「秘密」和「團隊」的標籤。

```
aws codeguru-reviewer untag-resource \
    --resource-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --tag-keys Status Team
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CodeGuru Reviewer 使用者指南》中的從 CodeGuru Reviewer 關聯儲存庫 (AWS CLI) 移除標籤。 CodeGuru

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

使用 的 CodePipeline 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 CodePipeline 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`acknowledge-job`

以下程式碼範例顯示如何使用 acknowledge-job。

AWS CLI

擷取指定任務的相關資訊

此範例會傳回指定任務的相關資訊，包括該任務存在時的狀態。這僅用於任務工作者和自訂動作。若要判斷 nonce 的值和任務 ID，請使用 aws codepipeline poll-for-jobs。

命令：

```
aws codepipeline acknowledge-job --job-id f4f4ff82-2d11-EXAMPLE --nonce 3
```

輸出：

```
{
  "status": "InProgress"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcknowledgeJob。

create-custom-action-type

以下程式碼範例顯示如何使用 create-custom-action-type。

AWS CLI

建立自訂動作

此範例使用已建立的 JSON 檔案 （此處名為 MyCustomAction.json) 為 AWS CodePipeline 建立自訂動作，其中包含自訂動作的結構。 MyCustomAction.json) 如需建立自訂動作需求的詳細資訊，包括 檔案的結構，請參閱 AWS CodePipeline 使用者指南。

```
aws codepipeline create-custom-action-type --cli-input-json file://
MyCustomAction.json
```

JSON 檔案 的內容MyCustomAction.json：

```
{
    "category": "Build",
    "provider": "MyJenkinsProviderName",
    "version": "1",
    "settings": {
        "entityUrlTemplate": "https://192.0.2.4/job/{Config:ProjectName}/",
        "executionUrlTemplate": "https://192.0.2.4/job/{Config:ProjectName}/
lastSuccessfulBuild/{ExternalExecutionId}/"
```

```
    },
    "configurationProperties": [
        {
            "name": "MyJenkinsExampleBuildProject",
            "required": true,
            "key": true,
            "secret": false,
            "queryable": false,
            "description": "The name of the build project must be provided when this
 action is added to the pipeline.",
            "type": "String"
        }
    ],
    "inputArtifactDetails": {
        "maximumCount": 1,
        "minimumCount": 0
    },
    "outputArtifactDetails": {
        "maximumCount": 1,
        "minimumCount": 0
    }
}
```

此命令會傳回自訂動作的結構。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCustomActionType。

create-pipeline

以下程式碼範例顯示如何使用 create-pipeline。

AWS CLI

建立管道

此範例使用已建立的 JSON 檔案 （此處名為 MySecondPipeline.json) AWS CodePipeline 中建立管道，其中包含管道的結構。如需建立管道需求的詳細資訊，包括 檔案的結構，請參閱 AWS CodePipeline 使用者指南。

命令：

```
aws codepipeline create-pipeline --cli-input-json file://MySecondPipeline.json
```

JSON 檔案範例內容：

```
{
 "pipeline": {
  "roleArn": "arn:aws:iam::111111111111:role/AWS-CodePipeline-Service",
  "stages": [
    {
      "name": "Source",
      "actions": [
        {
          "inputArtifacts": [],
          "name": "Source",
          "actionTypeId": {
            "category": "Source",
            "owner": "AWS",
            "version": "1",
            "provider": "S3"
          },
          "outputArtifacts": [
            {
              "name": "MyApp"
            }
          ],
          "configuration": {
            "S3Bucket": "awscodepipeline-demo-bucket",
            "S3ObjectKey": "aws-codepipeline-s3-aws-codedeploy_linux.zip"
          },
          "runOrder": 1
        }
      ]
    },
    {
      "name": "Beta",
      "actions": [
        {
          "inputArtifacts": [
            {
              "name": "MyApp"
            }
          ],
          "name": "CodePipelineDemoFleet",
          "actionTypeId": {
            "category": "Deploy",
            "owner": "AWS",
```

```
            "version": "1",
            "provider": "CodeDeploy"
          },
          "outputArtifacts": [],
          "configuration": {
            "ApplicationName": "CodePipelineDemoApplication",
            "DeploymentGroupName": "CodePipelineDemoFleet"
          },
          "runOrder": 1
        }
      ]
    }
  ],
  "artifactStore": {
    "type": "S3",
    "location": "codepipeline-us-east-1-11EXAMPLE11"
  },
  "name": "MySecondPipeline",
  "version": 1
 }
}
```

輸出：

```
This command returns the structure of the pipeline.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePipeline。

`delete-custom-action-type`

以下程式碼範例顯示如何使用 `delete-custom-action-type`。

AWS CLI

刪除自訂動作

此範例使用已建立的 JSON 檔案 （此處名為 DeleteMyCustomAction.json) 刪除 AWS CodePipeline 中的自訂動作，其中包含要刪除的動作類型、提供者名稱和版本編號。使用 list-action-types 命令來檢視類別、版本和提供者的正確值。

命令：

```
aws codepipeline delete-custom-action-type --cli-input-json file://
DeleteMyCustomAction.json
```

JSON 檔案範例內容：

```
{
  "category": "Build",
  "version": "1",
  "provider": "MyJenkinsProviderName"
}
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCustomActionType。

delete-pipeline

以下程式碼範例顯示如何使用 delete-pipeline。

AWS CLI

刪除管道

此範例會從CodePipeline 刪除名為 MySecondPipeline 的管道。 AWS CodePipeline 使用 list-pipelines 命令來檢視與 AWS 您的帳戶相關聯的管道清單。

命令：

```
aws codepipeline delete-pipeline --name MySecondPipeline
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePipeline。

delete-webhook

以下程式碼範例顯示如何使用 delete-webhook。

AWS CLI

刪除 Webhook

下列delete-webhook範例會刪除 GitHub 第 1 版來源動作的 Webhook。您必須先使用 deregister-webhook-with-third-party命令取消註冊 Webhook，然後再刪除它。

```
aws codepipeline delete-webhook \
    --name my-webhook
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CodePipeline 使用者指南中的刪除 GitHub 來源的 Webhook。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteWebhook。

deregister-webhook-with-third-party

以下程式碼範例顯示如何使用 deregister-webhook-with-third-party。

AWS CLI

取消註冊 Webhook

下列deregister-webhook-with-third-party範例會刪除 GitHub 第 1 版來源動作的 Webhook。您必須先取消註冊 Webhook，然後才能將其刪除。

```
aws codepipeline deregister-webhook-with-third-party \
    --webhook-name my-webhook
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CodePipeline 使用者指南中的刪除 GitHub 來源的 Webhook。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterWebhookWithThirdParty。

disable-stage-transition

以下程式碼範例顯示如何使用 disable-stage-transition。

AWS CLI

停用轉換至管道中的階段

此範例會停用轉換為 CodePipeline 中 MyFirstPipeline AWS CodePipeline 管道的 Beta 階段。

命令：

```
aws codepipeline disable-stage-transition --pipeline-name MyFirstPipeline --stage-name Beta  --transition-type Inbound
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableStageTransition。

enable-stage-transition

以下程式碼範例顯示如何使用 enable-stage-transition。

AWS CLI

啟用轉換至管道中的階段

此範例可讓 轉換至 CodePipeline 中 MyFirstPipeline AWS CodePipeline 管道的 Beta 階段。

命令：

```
aws codepipeline enable-stage-transition --pipeline-name MyFirstPipeline --stage-name Beta  --transition-type Inbound
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableStageTransition。

get-job-details

以下程式碼範例顯示如何使用 get-job-details。

AWS CLI

取得任務的詳細資訊

此範例會傳回其 ID 由 f4f4ff82-2d11-EXAMPLE 表示之任務的詳細資訊。此命令僅用於自訂動作。呼叫此命令時，AWS CodePipeline 會傳回用於存放管道成品之 Amazon S3 儲存貯體的臨時登入資料，如果自訂動作需要的話。如果定義了任何 動作，此命令也會傳回為動作定義的任何秘密值。

命令：

```
aws codepipeline get-job-details --job-id f4f4ff82-2d11-EXAMPLE
```

輸出：

```
{
 "jobDetails": {
  "accountId": "111111111111",
  "data": {
    "actionConfiguration": {
      "__type": "ActionConfiguration",
      "configuration": {
        "ProjectName": "MyJenkinsExampleTestProject"
      }
    },
    "actionTypeId": {
      "__type": "ActionTypeId",
      "category": "Test",
      "owner": "Custom",
      "provider": "MyJenkinsProviderName",
      "version": "1"
    },
    "artifactCredentials": {
      "__type": "AWSSessionCredentials",
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "secretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
      "sessionToken":
 "fICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwd
+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/
f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/
MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpEIbb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQ
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs
    },
    "inputArtifacts": [
```

```
      {
        "__type": "Artifact",
        "location": {
          "s3Location": {
            "bucketName": "codepipeline-us-east-1-11EXAMPLE11",
            "objectKey": "MySecondPipeline/MyAppBuild/EXAMPLE"
          },
          "type": "S3"
        },
        "name": "MyAppBuild"
      }
    ],
    "outputArtifacts": [],
    "pipelineContext": {
      "__type": "PipelineContext",
      "action": {
        "name": "MyJenkinsTest-Action"
      },
      "pipelineName": "MySecondPipeline",
      "stage": {
        "name": "Testing"
      }
    }
  },
  "id": "f4f4ff82-2d11-EXAMPLE"
 }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetJobDetails。

get-pipeline-state

以下程式碼範例顯示如何使用 get-pipeline-state。

AWS CLI

取得管道狀態的相關資訊

此範例會傳回名為 MyFirstPipeline 之管道的最新狀態。

命令：

```
aws codepipeline get-pipeline-state --name MyFirstPipeline
```

輸出：

```
{
 "created": 1446137312.204,
 "pipelineName": "MyFirstPipeline",
 "pipelineVersion": 1,
 "stageStates": [
  {
    "actionStates": [
      {
        "actionName": "Source",
        "entityUrl": "https://console.aws.amazon.com/s3/home?#",
        "latestExecution": {
          "lastStatusChange": 1446137358.328,
          "status": "Succeeded"
        }
      }
    ],
    "stageName": "Source"
  },
  {
    "actionStates": [
      {
        "actionName": "CodePipelineDemoFleet",
        "entityUrl": "https://console.aws.amazon.com/codedeploy/home?#/applications/
CodePipelineDemoApplication/deployment-groups/CodePipelineDemoFleet",
        "latestExecution": {
          "externalExecutionId": "d-EXAMPLE",
          "externalExecutionUrl": "https://console.aws.amazon.com/codedeploy/home?#/
deployments/d-EXAMPLE",
          "lastStatusChange": 1446137493.131,
          "status": "Succeeded",
          "summary": "Deployment Succeeded"
        }
      }
    ],
    "inboundTransitionState": {
      "enabled": true
    },
    "stageName": "Beta"
  }
 ],
 "updated": 1446137312.204
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPipelineState。

get-pipeline

以下程式碼範例顯示如何使用 get-pipeline。

AWS CLI

檢視管道的結構

此範例會傳回名為 MyFirstPipeline 之管道的結構。

命令：

```
aws codepipeline get-pipeline --name MyFirstPipeline
```

輸出：

```
{
  "pipeline": {
      "roleArn": "arn:aws:iam::111111111111:role/AWS-CodePipeline-Service",
      "stages": [
          {
              "name": "Source",
              "actions": [
                  {
                      "inputArtifacts": [],
                      "name": "Source",
                      "actionTypeId": {
                          "category": "Source",
                          "owner": "AWS",
                          "version": "1",
                          "provider": "S3"
                      },
                      "outputArtifacts": [
                          {
                              "name": "MyApp"
                          }
                      ],
                      "configuration": {
```

```
                        "S3Bucket": "awscodepipeline-demo-bucket",
                        "S3ObjectKey": "aws-codepipeline-s3-aws-
codedeploy_linux.zip"
                    },
                    "runOrder": 1
                }
            ]
        },
        {
            "name": "Beta",
            "actions": [
                {
                    "inputArtifacts": [
                        {
                            "name": "MyApp"
                        }
                    ],
                    "name": "CodePipelineDemoFleet",
                    "actionTypeId": {
                        "category": "Deploy",
                        "owner": "AWS",
                        "version": "1",
                        "provider": "CodeDeploy"
                    },
                    "outputArtifacts": [],
                    "configuration": {
                        "ApplicationName": "CodePipelineDemoApplication",
                        "DeploymentGroupName": "CodePipelineDemoFleet"
                    },
                    "runOrder": 1
                }
            ]
        }
    ],
    "artifactStore": {
        "type": "S3",
        "location": "codepipeline-us-east-1-11EXAMPLE11"
    },
    "name": "MyFirstPipeline",
    "version": 1
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPipeline。

list-action-executions

以下程式碼範例顯示如何使用 list-action-executions。

AWS CLI

列出動作執行

下列list-action-executions範例檢視管道的動作執行詳細資訊，例如動作執行 ID、輸入成品、輸出成品、執行結果和狀態。

```
aws codepipeline list-action-executions \
    --pipeline-name myPipeline
```

輸出：

```
{
    "actionExecutionDetails": [
        {
            "pipelineExecutionId": "EXAMPLE0-adfc-488e-bf4c-1111111720d3",
            "actionExecutionId": "EXAMPLE4-2ee8-4853-bd6a-111111158148",
            "pipelineVersion": 12,
            "stageName": "Deploy",
            "actionName": "Deploy",
            "startTime": 1598572628.6,
            "lastUpdateTime": 1598572661.255,
            "status": "Succeeded",
            "input": {
                "actionTypeId": {
                    "category": "Deploy",
                    "owner": "AWS",
                    "provider": "CodeDeploy",
                    "version": "1"
                },
                "configuration": {
                    "ApplicationName": "my-application",
                    "DeploymentGroupName": "my-deployment-group"
                },
                "resolvedConfiguration": {
                    "ApplicationName": "my-application",
                    "DeploymentGroupName": "my-deployment-group"
                },
                "region": "us-east-1",
```

```
                "inputArtifacts": [
                    {
                        "name": "SourceArtifact",
                        "s3location": {
                            "bucket": "artifact-bucket",
                            "key": "myPipeline/SourceArti/key"
                        }
                    }
                ],
                "namespace": "DeployVariables"
            },
            "output": {
                "outputArtifacts": [],
                "executionResult": {
                    "externalExecutionId": "d-EXAMPLEE5",
                    "externalExecutionSummary": "Deployment Succeeded",
                    "externalExecutionUrl": "https://myaddress.com"
                },
                "outputVariables": {}
            }
        },
        {
            "pipelineExecutionId": "EXAMPLE0-adfc-488e-bf4c-1111111720d3",
            "actionExecutionId": "EXAMPLE5-abb4-4192-9031-11111113a7b0",
            "pipelineVersion": 12,
            "stageName": "Source",
            "actionName": "Source",
            "startTime": 1598572624.387,
            "lastUpdateTime": 1598572628.16,
            "status": "Succeeded",
            "input": {
                "actionTypeId": {
                    "category": "Source",
                    "owner": "AWS",
                    "provider": "CodeCommit",
                    "version": "1"
                },
                "configuration": {
                    "BranchName": "production",
                    "PollForSourceChanges": "false",
                    "RepositoryName": "my-repo"
                },
                "resolvedConfiguration": {
                    "BranchName": "production",
```

```
                    "PollForSourceChanges": "false",
                    "RepositoryName": "my-repo"
                },
                "region": "us-east-1",
                "inputArtifacts": [],
                "namespace": "SourceVariables"
            },
            "output": {
                "outputArtifacts": [
                    {
                        "name": "SourceArtifact",
                        "s3location": {
                            "bucket": "amzn-s3-demo-bucket",
                            "key": "myPipeline/SourceArti/key"
                        }
                    }
                ],
                "executionResult": {
                    "externalExecutionId":
 "1111111ad99dcd35914c00b7fbea13995EXAMPLE",
                    "externalExecutionSummary": "Edited template.yml",
                    "externalExecutionUrl": "https://myaddress.com"
                },
                "outputVariables": {
                    "AuthorDate": "2020-05-08T17:45:43Z",
                    "BranchName": "production",
                    "CommitId": "EXAMPLEad99dcd35914c00b7fbea139951111111",
                    "CommitMessage": "Edited template.yml",
                    "CommitterDate": "2020-05-08T17:45:43Z",
                    "RepositoryName": "my-repo"
                }
            }
        },
. . . .
```

如需詳細資訊，請參閱 AWS CodePipeline 使用者指南中的檢視動作執行 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListActionExecutions。

list-action-types

以下程式碼範例顯示如何使用 `list-action-types`。

AWS CLI

檢視可用的動作類型

list-action-types 命令會傳回您 AWS 帳戶可用的所有動作結構。此範例使用 --action-owner-filter 選項來僅傳回自訂動作。

命令：

```
aws codepipeline list-action-types --action-owner-filter Custom
```

輸出：

```
{
  "actionTypes": [
      {
          "inputArtifactDetails": {
              "maximumCount": 5,
              "minimumCount": 0
          },
          "actionConfigurationProperties": [
              {
                  "secret": false,
                  "required": true,
                  "name": "MyJenkinsExampleBuildProject",
                  "key": true,
                  "queryable": true
              }
          ],
          "outputArtifactDetails": {
              "maximumCount": 5,
              "minimumCount": 0
          },
          "id": {
              "category": "Build",
              "owner": "Custom",
              "version": "1",
              "provider": "MyJenkinsProviderName"
          },
          "settings": {
              "entityUrlTemplate": "http://192.0.2.4/job/{Config:ProjectName}",
              "executionUrlTemplate": "http://192.0.2.4/job/{Config:ProjectName}/
{ExternalExecutionId}"
```

```
            }
        },
        {
            "inputArtifactDetails": {
                "maximumCount": 5,
                "minimumCount": 0
            },
            "actionConfigurationProperties": [
                {
                    "secret": false,
                    "required": true,
                    "name": "MyJenkinsExampleTestProject",
                    "key": true,
                    "queryable": true
                }
            ],
            "outputArtifactDetails": {
                "maximumCount": 5,
                "minimumCount": 0
            },
            "id": {
                "category": "Test",
                "owner": "Custom",
                "version": "1",
                "provider": "MyJenkinsProviderName"
            },
            "settings": {
                "entityUrlTemplate": "http://192.0.2.4/job/{Config:ProjectName}",
                "executionUrlTemplate": "http://192.0.2.4/job/{Config:ProjectName}/
{ExternalExecutionId}"
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListActionTypes。

list-pipeline-executions

以下程式碼範例顯示如何使用 list-pipeline-executions。

AWS CLI

檢視管道執行歷史記錄

下列list-pipeline-executions範例顯示您 AWS 帳戶中管道的管道執行歷史記錄。

```
aws codepipeline list-pipeline-executions \
    --pipeline-name MyPipeline
```

輸出：

```
{
    "pipelineExecutionSummaries": [
        {
            "lastUpdateTime": 1496380678.648,
            "pipelineExecutionId": "7cf7f7cb-3137-539g-j458-d7eu3EXAMPLE",
            "startTime": 1496380258.243,
            "status": "Succeeded"
        },
        {
            "lastUpdateTime": 1496591045.634,
            "pipelineExecutionId": "3137f7cb-8d494hj4-039j-d84l-d7eu3EXAMPLE",
            "startTime": 1496590401.222,
            "status": "Succeeded"
        },
        {
            "lastUpdateTime": 1496946071.6456,
            "pipelineExecutionId": "4992f7jf-7cf7-913k-k334-d7eu3EXAMPLE",
            "startTime": 1496945471.5645,
            "status": "Succeeded"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodePipeline 使用者指南中的檢視執行歷史記錄。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPipelineExecutions。

list-pipelines

以下程式碼範例顯示如何使用 list-pipelines。

AWS CLI

檢視管道清單

此範例列出與使用者帳戶相關聯的所有 AWS CodePipeline 管道 AWS 。

命令：

```
aws codepipeline list-pipelines
```

輸出：

```
{
  "pipelines": [
      {
          "updated": 1439504274.641,
          "version": 1,
          "name": "MyFirstPipeline",
          "created": 1439504274.641
      },
      {
          "updated": 1436461837.992,
          "version": 2,
          "name": "MySecondPipeline",
          "created": 1436460801.381
      }
      ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPipelines。

list-tags-for-resource

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出標籤

下列`list-tags-for-resource`範例會擷取連接至指定管道資源的所有標籤清單。

```
aws codepipeline list-tags-for-resource \
```

```
    --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline
```

輸出：

```
{
    "tags": {
        "Project": "ProjectA",
        "IscontainerBased": "true"
    }
}
```

如需詳細資訊，請參閱 AWS CodePipeline 使用者指南中的檢視管道的標籤 (CLI)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`list-webhooks`

以下程式碼範例顯示如何使用 `list-webhooks`。

AWS CLI

列出 Webhook

下列`list-webhooks`範例會擷取連接至指定管道資源的所有標籤清單。

```
aws codepipeline list-webhooks \
    --endpoint-url "https://codepipeline.eu-central-1.amazonaws.com" \
    --region "eu-central-1"
```

輸出：

```
{
    "webhooks": [
        {
            "url": "https://webhooks.domain.com/
trigger111111111EXAMPLE1111111111111111": {
                "authenticationConfiguration": {
                    "SecretToken": "Secret"
                },
                "name": "my-webhook",
                "authentication": "GITHUB_HMAC",
                "targetPipeline": "my-Pipeline",
```

```
                "targetAction": "Source",
                "filters": [
                    {
                        "jsonPath": "$.ref",
                        "matchEquals": "refs/heads/{Branch}"
                    }
                ]
            },
            "arn": "arn:aws:codepipeline:eu-central-1:123456789012:webhook:my-
webhook"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodePipeline 使用者指南中的列出您帳戶中的 Webhook。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListWebhooks。

poll-for-jobs

以下程式碼範例顯示如何使用 poll-for-jobs。

AWS CLI

檢視任何可用的任務

此範例會傳回任何任務的相關資訊，讓任務工作者採取行動。此範例使用預先定義的 JSON 檔案 (MyActionTypeInfo.json)，提供任務工作者處理任務之動作類型的相關資訊。此命令僅用於自訂動作。呼叫此命令時， AWS CodePipeline 會傳回用於存放管道成品之 Amazon S3 儲存貯體的臨時登入資料。如果定義了任何 動作，此命令也會傳回為動作定義的任何秘密值。

命令：

```
aws codepipeline poll-for-jobs --cli-input-json file://MyActionTypeInfo.json
```

JSON 檔案範例內容：

```
{
  "actionTypeId": {
    "category": "Test",
    "owner": "Custom",
    "provider": "MyJenkinsProviderName",
```

```
    "version": "1"
  },
  "maxBatchSize": 5,
  "queryParam": {
      "ProjectName": "MyJenkinsTestProject"
  }
}
```

輸出：

```
{
 "jobs": [
  {
    "accountId": "111111111111",
    "data": {
      "actionConfiguration": {
        "__type": "ActionConfiguration",
        "configuration": {
          "ProjectName": "MyJenkinsExampleTestProject"
        }
      },
      "actionTypeId": {
        "__type": "ActionTypeId",
        "category": "Test",
        "owner": "Custom",
        "provider": "MyJenkinsProviderName",
        "version": "1"
      },
      "artifactCredentials": {
        "__type": "AWSSessionCredentials",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "secretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "sessionToken":
 "fICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwd
+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/
f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/
MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpEIbb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQ
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs
      },
      "inputArtifacts": [
        {
          "__type": "Artifact",
          "location": {
```

```
            "s3Location": {
              "bucketName": "codepipeline-us-east-1-11EXAMPLE11",
              "objectKey": "MySecondPipeline/MyAppBuild/EXAMPLE"
            },
            "type": "S3"
          },
          "name": "MyAppBuild"
        }
      ],
      "outputArtifacts": [],
      "pipelineContext": {
        "__type": "PipelineContext",
        "action": {
          "name": "MyJenkinsTest-Action"
        },
        "pipelineName": "MySecondPipeline",
        "stage": {
          "name": "Testing"
        }
      }
    },
    "id": "ef66c259-64f9-EXAMPLE",
    "nonce": "3"
  }
 ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PollForJobs。

put-webhook

以下程式碼範例顯示如何使用 put-webhook。

AWS CLI

建立 Webhook

下列put-webhook範例會為 GitHub 第 1 版來源動作建立 Webhook。建立 Webhook 之後，您必須使用 register-webhook-with-third-party命令進行註冊。

```
aws codepipeline put-webhook \
    --cli-input-json file://webhook_json.json \
```

```
    --region "eu-central-1"
```

webhook_json.json 的內容：

```
{
    "webhook": {
        "name": "my-webhook",
        "targetPipeline": "pipeline_name",
        "targetAction": "source_action_name",
        "filters": [
            {
                "jsonPath": "$.ref",
                "matchEquals": "refs/heads/{Branch}"
            }
        ],
        "authentication": "GITHUB_HMAC",
        "authenticationConfiguration": {
            "SecretToken": "secret"
        }
    }
}
```

輸出：

```
{
    "webhook": {
        "url": "https://webhooks.domain.com/
trigger111111111EXAMPLE11111111111111111",
        "definition": {
            "authenticationConfiguration": {
                "SecretToken": "secret"
            },
            "name": "my-webhook",
            "authentication": "GITHUB_HMAC",
            "targetPipeline": "pipeline_name",
            "targetAction": "Source",
            "filters": [
                {
                    "jsonPath": "$.ref",
                    "matchEquals": "refs/heads/{Branch}"
                }
            ]
        },
```

```
        "arn": "arn:aws:codepipeline:eu-central-1:123456789012:webhook:my-webhook"
    },
    "tags": [
        {
            "key": "Project",
            "value": "ProjectA"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS CodePipeline 使用者指南中的為 GitHub 來源建立 Webhook。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutWebhook。

retry-stage-execution

以下程式碼範例顯示如何使用 retry-stage-execution。

AWS CLI

重試失敗的動作

下列retry-stage-execution範例會重試動作失敗的階段。

```
aws codepipeline retry-stage-execution \
    --pipeline-name MyPipeline \
    --stage-name Deploy \
    --pipeline-execution-id b59babff-5f34-EXAMPLE \
    --retry-mode FAILED_ACTIONS
```

輸出：

```
{
    "pipelineExecutionId": "b59babff-5f34-EXAMPLE"
}
```

如需詳細資訊，請參閱 AWS CodePipeline 使用者指南中的重試失敗動作 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RetryStageExecution。

start-pipeline-execution

以下程式碼範例顯示如何使用 start-pipeline-execution。

AWS CLI

透過管道執行最新的修訂版

此範例會透過名為 "MyFirstPipeline" 的管道，執行管道來源階段中存在的最新修訂版。

命令：

```
aws codepipeline start-pipeline-execution --name MyFirstPipeline
```

輸出：

```
{
  "pipelineExecutionId": "3137f7cb-7cf7-EXAMPLE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartPipelineExecution。

stop-pipeline-execution

以下程式碼範例顯示如何使用 stop-pipeline-execution。

AWS CLI

停止管道執行

下列stop-pipeline-execution範例預設為等待進行中的動作完成，然後停止管道。如果執行已處於 Stopping (停止中) 狀態，您就無法選擇停止並等待。您可以選擇停止並捨棄已處於 Stopping (停止中) 狀態的執行。

```
aws codepipeline stop-pipeline-execution \
    --pipeline-name MyFirstPipeline \
    --pipeline-execution-id d-EXAMPLE \
    --reason "Stopping pipeline after the build action is done"
```

此命令不會傳回輸出。

如需詳細資訊，請參閱 AWS CodePipeline 使用者指南中的停止管道執行 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopPipelineExecution。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

標記資源

下列tag-resource範例會將一組提供的標籤與管道建立關聯。使用此命令來新增或編輯標籤。

```
aws codepipeline tag-resource \
    --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline \
    --tags key=Project,value=ProjectA key=IscontainerBased,value=true
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CodePipeline 使用者指南中的將標籤新增至管道 (CLI)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從連線資源移除 AWS 標籤

下列untag-resource範例會從指定的資源移除標籤。

```
aws codepipeline untag-resource \
    --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline \
    --tag-keys Project IscontainerBased
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CodePipeline 使用者指南中的從管道 (CLI) 移除標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-pipeline

以下程式碼範例顯示如何使用 update-pipeline。

AWS CLI

更新管道的結構

此範例使用 update-pipeline 命令搭配 --cli-input-json 引數。此範例使用預先定義的 JSON 檔案 (MyFirstPipeline.json) 來更新管道結構。 AWS CodePipeline 會辨識 JSON 檔案中包含的管道名稱，然後套用管道結構中修改欄位的任何變更來更新管道。

建立預先定義的 JSON 檔案時，請使用下列準則：

如果您使用的是使用 get-pipeline 命令擷取的管道結構，您必須從 JSON 檔案的管道結構中移除中繼資料區段 (「中繼資料」：{} 行和其中的「建立」、「pipelineARN」和「更新」欄位）。管道名稱無法變更。

命令：

```
aws codepipeline update-pipeline --cli-input-json file://MyFirstPipeline.json
```

JSON 檔案內容範例：

```
{
 "pipeline": {
  "roleArn": "arn:aws:iam::111111111111:role/AWS-CodePipeline-Service",
  "stages": [
    {
      "name": "Source",
      "actions": [
        {
          "inputArtifacts": [],
          "name": "Source",
          "actionTypeId": {
            "category": "Source",
            "owner": "AWS",
            "version": "1",
            "provider": "S3"
          },
          "outputArtifacts": [
            {
              "name": "MyApp"
            }
          ],
          "configuration": {
```

```
            "S3Bucket": "awscodepipeline-demo-bucket2",
            "S3ObjectKey": "aws-codepipeline-s3-aws-codedeploy_linux.zip"
          },
          "runOrder": 1
        }
      ]
    },
    {
      "name": "Beta",
      "actions": [
        {
          "inputArtifacts": [
            {
              "name": "MyApp"
            }
          ],
          "name": "CodePipelineDemoFleet",
          "actionTypeId": {
            "category": "Deploy",
            "owner": "AWS",
            "version": "1",
            "provider": "CodeDeploy"
          },
          "outputArtifacts": [],
          "configuration": {
            "ApplicationName": "CodePipelineDemoApplication",
            "DeploymentGroupName": "CodePipelineDemoFleet"
          },
          "runOrder": 1
        }
      ]
    }
  ],
  "artifactStore": {
    "type": "S3",
    "location": "codepipeline-us-east-1-11EXAMPLE11"
  },
  "name": "MyFirstPipeline",
  "version": 1
 }
}
```

輸出：

```
{
 "pipeline": {
  "artifactStore": {
    "location": "codepipeline-us-east-1-11EXAMPLE11",
    "type": "S3"
  },
  "name": "MyFirstPipeline",
  "roleArn": "arn:aws:iam::111111111111:role/AWS-CodePipeline-Service",
  "stages": [
    {
      "actions": [
        {
          "actionTypeId": {
            "__type": "ActionTypeId",
            "category": "Source",
            "owner": "AWS",
            "provider": "S3",
            "version": "1"
          },
          "configuration": {
            "S3Bucket": "awscodepipeline-demo-bucket2",
            "S3ObjectKey": "aws-codepipeline-s3-aws-codedeploy_linux.zip"
          },
          "inputArtifacts": [],
          "name": "Source",
          "outputArtifacts": [
            {
              "name": "MyApp"
            }
          ],
          "runOrder": 1
        }
      ],
      "name": "Source"
    },
    {
      "actions": [
        {
          "actionTypeId": {
            "__type": "ActionTypeId",
            "category": "Deploy",
            "owner": "AWS",
            "provider": "CodeDeploy",
```

```
            "version": "1"
          },
          "configuration": {
            "ApplicationName": "CodePipelineDemoApplication",
            "DeploymentGroupName": "CodePipelineDemoFleet"
          },
          "inputArtifacts": [
            {
              "name": "MyApp"
            }
          ],
          "name": "CodePipelineDemoFleet",
          "outputArtifacts": [],
          "runOrder": 1
        }
      ],
      "name": "Beta"
    }
  ],
  "version": 3
 }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePipeline。

AWS CodeStar 使用 的通知範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 AWS CodeStar 通知來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-notification-rule`

以下程式碼範例顯示如何使用 `create-notification-rule`。

AWS CLI

建立通知規則

下列`create-notification-rule`範例使用名為 的 JSON 檔案`rule.json`，`MyNotificationRule`為指定 AWS 帳戶中名為 的儲存庫建立名為 `MyDemoRepo` 的通知規則。建立分支和標籤時，具有`FULL`詳細資訊類型的通知會傳送至指定的目標 Amazon SNS 主題。

```
aws codestar-notifications create-notification-rule \
    --cli-input-json file://rule.json
```

`rule.json` 的內容：

```
{
    "Name": "MyNotificationRule",
    "EventTypeIds": [
        "codecommit-repository-branches-and-tags-created"
    ],
    "Resource": "arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo",
    "Targets": [
        {
            "TargetType": "SNS",
            "TargetAddress": "arn:aws:sns:us-
east-1:123456789012:MyNotificationTopic"
        }
    ],
    "Status": "ENABLED",
    "DetailType": "FULL"
}
```

輸出：

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
```

```
}
```

如需詳細資訊，請參閱《 AWS 開發人員工具主控台使用者指南》中的建立通知規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateNotificationRule。

delete-notification-rule

以下程式碼範例顯示如何使用 delete-notification-rule。

AWS CLI

刪除通知規則

下列delete-notification-rule範例會刪除指定的通知規則。

```
aws codestar-notifications delete-notification-rule \
    --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE
```

輸出：

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

如需詳細資訊，請參閱《 AWS 開發人員工具主控台使用者指南》中的刪除通知規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNotificationRule。

delete-target

以下程式碼範例顯示如何使用 delete-target。

AWS CLI

刪除通知規則目標

下列delete-target範例會從設定為使用它做為目標的所有通知規則中移除指定的目標，然後刪除目標。

```
aws codestar-notifications  delete-target \
    --target-address arn:aws:sns:us-east-1:123456789012:MyNotificationTopic \
    --force-unsubscribe-all
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 AWS 開發人員工具主控台使用者指南》中的刪除通知規則目標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTarget。

describe-notification-rule

以下程式碼範例顯示如何使用 describe-notification-rule。

AWS CLI

擷取通知規則的詳細資訊

下列describe-notification-rule範例會擷取指定通知規則的詳細資訊。

```
aws codestar-notifications describe-notification-rule \
    --arn arn:aws:codestar-notifications:us-west-2:123456789012:notificationrule/
dc82df7a-EXAMPLE
```

輸出：

```
{
    "LastModifiedTimestamp": 1569199844.857,
    "EventTypes": [
        {
            "ServiceName": "CodeCommit",
            "EventTypeName": "Branches and tags: Created",
            "ResourceType": "Repository",
            "EventTypeId": "codecommit-repository-branches-and-tags-created"
        }
    ],
    "Status": "ENABLED",
    "DetailType": "FULL",
    "Resource": "arn:aws:codecommit:us-west-2:123456789012:MyDemoRepo",
    "Arn": "arn:aws:codestar-notifications:us-west-w:123456789012:notificationrule/
dc82df7a-EXAMPLE",
    "Targets": [
```

```
        {
            "TargetStatus": "ACTIVE",
            "TargetAddress": "arn:aws:sns:us-
west-2:123456789012:MyNotificationTopic",
            "TargetType": "SNS"
        }
    ],
    "Name": "MyNotificationRule",
    "CreatedTimestamp": 1569199844.857,
    "CreatedBy": "arn:aws:iam::123456789012:user/Mary_Major"
}
```

如需詳細資訊，請參閱《 AWS 開發人員工具主控台使用者指南》中的檢視通知規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeNotificationRule。

`list-event-types`

以下程式碼範例顯示如何使用 `list-event-types`。

AWS CLI

取得通知規則的事件類型清單

下列`list-event-types`範例會擷取 CodeDeploy 應用程式所有可用通知事件類型的篩選清單。如果您不使用篩選條件，命令會傳回所有資源類型的所有通知事件類型。

```
aws codestar-notifications list-event-types \
    --filters Name=SERVICE_NAME,Value=CodeDeploy
```

輸出：

```
{
    "EventTypes": [
        {
            "EventTypeId": "codedeploy-application-deployment-succeeded",
            "ServiceName": "CodeDeploy",
            "EventTypeName": "Deployment: Succeeded",
            "ResourceType": "Application"
        },
        {
            "EventTypeId": "codedeploy-application-deployment-failed",
```

```
            "ServiceName": "CodeDeploy",
            "EventTypeName": "Deployment: Failed",
            "ResourceType": "Application"
        },
        {
            "EventTypeId": "codedeploy-application-deployment-started",
            "ServiceName": "CodeDeploy",
            "EventTypeName": "Deployment: Started",
            "ResourceType": "Application"
        }
    ]
}
```

如需詳細資訊，請參閱《 AWS 開發人員工具主控台使用者指南》中的建立通知規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEventTypes。

`list-notification-rules`

以下程式碼範例顯示如何使用 `list-notification-rules`。

AWS CLI

擷取通知規則清單

下列`list-notification-rules`範例會擷取指定區域中所有通知規則的清單 AWS 。

```
aws codestar-notifications list-notification-rules --region us-east-1
```

輸出：

```
{
    "NotificationRules": [
        {
            "Id": "dc82df7a-EXAMPLE",
            "Arn": "arn:aws:codestar-notifications:us-
east-1:123456789012:notificationrule/dc82df7a-EXAMPLE"
        },
        {
            "Id": "8d1f0983-EXAMPLE",
            "Arn": "arn:aws:codestar-notifications:us-
east-1:123456789012:notificationrule/8d1f0983-EXAMPLE"
        }
```

```
    ]
}
```

如需詳細資訊，請參閱《 AWS 開發人員工具主控台使用者指南》中的檢視通知規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListNotificationRules。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

取得連接至通知規則的標籤清單

下列list-tags-for-resource範例會擷取連接至指定通知規則的所有標籤清單。在此範例中，通知規則目前沒有與其相關聯的標籤。

```
aws codestar-notifications list-tags-for-resource \
    --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
fe1efd35-EXAMPLE
```

輸出：

```
{
    "Tags": {}
}
```

如需詳細資訊，請參閱《 AWS 開發人員工具主控台使用者指南》中的建立通知規則。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

list-targets

以下程式碼範例顯示如何使用 list-targets。

AWS CLI

擷取通知規則目標的清單

下列list-targets範例會擷取指定區域中所有通知規則目標的清單 AWS 。

```
aws codestar-notifications list-targets \
    --region us-east-1
```

輸出：

```
{
    "Targets": [
        {
            "TargetAddress": "arn:aws:sns:us-
east-1:123456789012:MySNSTopicForNotificationRules",
            "TargetType": "SNS",
            "TargetStatus": "ACTIVE"
        },
        {
            "TargetAddress": "arn:aws:sns:us-
east-1:123456789012:MySNSTopicForNotificationsAboutMyDemoRepo",
            "TargetType": "SNS",
            "TargetStatus": "ACTIVE"
        }
    ]
}
```

如需詳細資訊，請參閱《 AWS 開發人員工具主控台使用者指南》中的檢視通知規則目標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTargets。

subscribe

以下程式碼範例顯示如何使用 subscribe。

AWS CLI

將目標新增至通知規則

下列subscribe範例會將 Amazon SNS 主題新增為指定通知規則的目標。

```
aws codestar-notifications subscribe \
    --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE \
    --target TargetType=SNS,TargetAddress=arn:aws:sns:us-
east-1:123456789012:MyNotificationTopic
```

輸出：

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

如需詳細資訊，請參閱《 AWS 開發人員工具主控台使用者指南》中的新增或移除 Amazon SNS 主題做為通知規則的目標。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 Subscribe。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤新增至通知規則

下列tag-resource範例會將索引鍵名稱為 Team且值為 的標籤Li_Juan新增至指定的通知規則。

```
aws codestar-notifications tag-resource \
    --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
fe1efd35-EXAMPLE \
    --tags Team=Li_Juan
```

輸出：

```
{
    "Tags": {
        "Team": "Li_Juan"
    }
}
```

如需詳細資訊，請參閱《 AWS 開發人員工具主控台使用者指南》中的建立通知規則。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

unsubscribe

以下程式碼範例顯示如何使用 unsubscribe。

AWS CLI

從通知規則中移除目標

下列unsubscribe範例會從指定的通知規則中移除 Amazon SNS 主題做為目標。

```
aws codestar-notifications unsubscribe \
    --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE \
    --target TargetType=SNS,TargetAddress=arn:aws:sns:us-
east-1:123456789012:MyNotificationTopic
```

輸出：

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
    "TargetAddress": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopic"
}
```

如需詳細資訊，請參閱《 AWS 開發人員工具主控台使用者指南》中的新增或移除 Amazon SNS 主題做為通知規則的目標。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 Unsubscribe。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從通知規則移除標籤

下列untag-resource範例Team會從指定的通知規則中移除具有金鑰名稱的標籤。

```
aws codestar-notifications untag-resource \
    --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
fe1efd35-EXAMPLE \
    --tag-keys Team
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 AWS 開發人員工具主控台使用者指南》中的編輯通知規則。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-notification-rule

以下程式碼範例顯示如何使用 update-notification-rule。

AWS CLI

更新通知規則

下列update-notification-rule範例123456789012會使用名為 MyNotificationRule的 JSON 檔案，更新 AWS 帳戶中名為 的通知規則update.json。

```
aws codestar-notifications update-notification-rule \
    --cli-input-json file://update.json
```

update.json 的內容：

```
{
    "Name": "MyUpdatedNotificationRule",
    "EventTypeIds": [
        "codecommit-repository-branches-and-tags-created"
    ],
    "Resource": "arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo",
    "Targets": [
        {
            "TargetType": "SNS",
            "TargetAddress": "arn:aws:sns:us-
east-1:123456789012:MyNotificationTopic"
        }
    ],
    "Status": "ENABLED",
    "DetailType": "FULL"
}
```

輸出：

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
```

```
}
```

如需詳細資訊，請參閱《 AWS 開發人員工具主控台使用者指南》中的編輯通知規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateNotificationRule。

使用 的 CodeConnections 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 CodeConnections 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-connection`

以下程式碼範例顯示如何使用 `create-connection`。

AWS CLI

建立連線

下列`create-connection`範例示範如何建立與第三方儲存庫的連線。此範例會建立第三方供應商為 Bitbucket 的連線。

根據預設，透過 CLI AWS 或 AWS CloudFormation 建立的連線處於待定狀態。建立與 CLI 或 AWS CloudFormation 的連線後，請使用 主控台編輯連線，使其狀態為可用。

```
aws codestar-connections create-connection \
    --provider-type Bitbucket \
    --connection-name MyConnection
```

輸出：

```
{
    "ConnectionArn": "arn:aws:codestar-connections:us-
east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f"
}
```

如需詳細資訊，請參閱《 開發人員工具主控台使用者指南》中的建立連線。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateConnection。

create-host

以下程式碼範例顯示如何使用 create-host。

AWS CLI

建立主機

下列create-host範例示範如何建立主機，以代表安裝第三方供應商之基礎設施的端點。此範例會建立主機，其中第三方安裝的提供者是 GitHub Enterprise Server。

根據預設，透過 CLI AWS 建立的主機處於待定狀態。使用 CLI 建立主機之後，請使用 主控台或 CLI 來設定主機，使其狀態可用。

```
aws codestar-connections create-host \
    --name MyHost \
    --provider-type GitHubEnterpriseServer \
    --provider-endpoint "https://my-instance.dev"
```

輸出：

```
{
    "HostArn": "arn:aws:codestar-connections:us-east-1:123456789012:host/My-
Host-28aef605"
}
```

如需詳細資訊，請參閱《 開發人員工具主控台使用者指南》中的建立主機 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateHost。

delete-connection

以下程式碼範例顯示如何使用 delete-connection。

AWS CLI

刪除連線

下列delete-connection範例顯示如何刪除連線。

```
aws codestar-connections delete-connection \
    --connection-arn arn:aws:codestar-connections:us-west-2:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 開發人員工具主控台使用者指南》中的刪除連線 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteConnection。

delete-host

以下程式碼範例顯示如何使用 delete-host。

AWS CLI

刪除主機

下列delete-host範例示範如何刪除主機。您必須先刪除與主機相關聯的所有連線，才能刪除該主機。

```
aws codestar-connections delete-host \
    --host-arn "arn:aws:codestar-connections:us-east-1 :123456789012:host/My-Host-28aef605"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 開發人員工具主控台使用者指南》中的刪除主機 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteHost。

get-connection

以下程式碼範例顯示如何使用 get-connection。

AWS CLI

取得連線的相關資訊

下列get-connection範例顯示連線的詳細資訊。

```
aws codestar-connections get-connection \
    --connection-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

輸出：

```
{
    "Connection": {
        "ConnectionName": "MyConnection",
        "ConnectionArn": "arn:aws:codestar-connections:us-east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
        "ProviderType": "Bitbucket",
        "OwnerAccountId": "123456789012",
        "ConnectionStatus": "AVAILABLE"
    }
}
```

如需詳細資訊，請參閱《 開發人員工具主控台使用者指南》中的檢視連線詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetConnection。

get-host

以下程式碼範例顯示如何使用 get-host。

AWS CLI

取得主機的相關資訊

下列get-host範例顯示主機的詳細資訊：

```
aws codestar-connections get-host \
    --host-arn arn:aws:codestar-connections:us-east-1:123456789012:host/MyHost-28aef605
```

輸出：

```
{
    "Name": "MyHost",
    "Status": "AVAILABLE",
    "ProviderType": "GitHubEnterpriseServer",
    "ProviderEndpoint": "https://test-instance-1.dev/"
}
```

如需詳細資訊，請參閱《 開發人員工具主控台使用者指南》中的檢視主機詳細資訊 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetHost。

list-connections

以下程式碼範例顯示如何使用 `list-connections`。

AWS CLI

列出連線

下列`list-connections`範例會擷取您帳戶中所有 Bitbucket 供應商類型的連線清單：

```
aws codestar-connections list-connections \
--provider-type Bitbucket \
--max-results 5 \
--next-token: next-token
```

輸出：

```
{
    "Connections": [
        {
            "ConnectionName": "my-connection",
            "ProviderType": "Bitbucket",
            "Status": "PENDING",
            "ARN": "arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
            "OwnerAccountId": "123456789012"
        },
        {
            "ConnectionName": "my-other-connection",
            "ProviderType": "Bitbucket",
            "Status": "AVAILABLE",
```

```
            "ARN": "arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
            "OwnerAccountId": "123456789012"
        },
    ],
    "NextToken": "next-token"
}
```

如需詳細資訊，請參閱《 開發人員工具主控台使用者指南》中的列出連線 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListConnections。

list-hosts

以下程式碼範例顯示如何使用 `list-hosts`。

AWS CLI

列出主機

下列`list-hosts`範例會擷取您帳戶中所有主機的清單。

```
aws codestar-connections list-hosts
```

輸出：

```
{
    "Hosts": [
        {
            "Name": "My-Host",
            "HostArn": "arn:aws:codestar-connections:us-east-1:123456789012:host/My-
Host-28aef605",
            "ProviderType": "GitHubEnterpriseServer",
            "ProviderEndpoint": "https://my-instance.test.dev",
            "Status": "AVAILABLE"
        }
    ]
}
```

如需詳細資訊，請參閱《 開發人員工具主控台使用者指南》中的列出主機 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListHosts。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出標籤

下列list-tags-for-resource範例會擷取連接至指定連線資源的所有標籤清單。

```
aws codestar-connections list-tags-for-resource \
    --resource-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

輸出：

```
{
    "Tags": [
        {
            "Key": "Project",
            "Value": "ProjectA"
        },
        {
            "Key": "ReadOnly",
            "Value": "true"
        }
    ]
}
```

如需詳細資訊，請參閱《 開發人員工具主控台使用者指南》中的檢視連線資源的標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

標記資源

下列tag-resource範例會將一組提供的標籤與連線建立關聯。使用此命令來新增或編輯標籤。

```
aws codestar-connections tag-resource \
    --resource-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f \
    --tags Key=Project,Value=ProjectA Key=IscontainerBased,Value=true
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 開發人員工具主控台使用者指南》中的將標籤新增至連線資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從連線資源移除 AWS 標籤

以下內容會從指定的資源untag-resource移除標籤。

```
aws codestar-connections untag-resource \
    --resource-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f \
    --tag-keys Project ReadOnly
```

輸出：

```
{
    "Tags": []
}
```

如需詳細資訊，請參閱《 開發人員工具主控台使用者指南》中的從連線資源移除標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

使用 的 Amazon Cognito 身分範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon Cognito Identity 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-identity-pool`

以下程式碼範例顯示如何使用 `create-identity-pool`。

AWS CLI

若要使用 Cognito 身分集區提供者建立身分集區

此範例會建立名為 MyIdentityPool 的身分集區。它具有 Cognito 身分集區提供者。不允許使用未驗證的身分。

命令：

```
aws cognito-identity create-identity-pool --identity-pool-
name MyIdentityPool --no-allow-unauthenticated-identities --cognito-
identity-providers ProviderName="cognito-idp.us-west-2.amazonaws.com/us-
west-2_aaaaaaaaa",ClientId="3n4b5urk1ft4fl3mg5e62d9ado",ServerSideTokenCheck=false
```

輸出：

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
      {
          "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-west-2_111111111",
          "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
          "ServerSideTokenCheck": false
      }
  ]
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateIdentityPool。

delete-identities

以下程式碼範例顯示如何使用 delete-identities。

AWS CLI

若要刪除身分集區

此範例會刪除身分集區。

命令：

```
aws cognito-identity delete-identity-pool --identity-ids-to-delete "us-
west-2:11111111-1111-1111-1111-111111111111"
```

輸出：

```
{
  "UnprocessedIdentityIds": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteIdentities。

delete-identity-pool

以下程式碼範例顯示如何使用 delete-identity-pool。

AWS CLI

若要刪除身分集區

以下範例 delete-identity-pool 會刪除指定的身分集區。

命令：

```
aws cognito-identity delete-identity-pool \
```

```
    --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteIdentityPool。

describe-identity-pool

以下程式碼範例顯示如何使用 describe-identity-pool。

AWS CLI

描述身分集區

此範例說明身分集區。

命令：

```
aws cognito-identity describe-identity-pool --identity-pool-id "us-
west-2:11111111-1111-1111-1111-111111111111"
```

輸出：

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
      {
          "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-west-2_111111111",
          "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
          "ServerSideTokenCheck": false
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeIdentityPool。

get-identity-pool-roles

以下程式碼範例顯示如何使用 get-identity-pool-roles。

AWS CLI

取得身分集區角色

此範例會取得身分集區角色。

命令：

```
aws cognito-identity get-identity-pool-roles --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

輸出：

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "Roles": {
      "authenticated": "arn:aws:iam::111111111111:role/Cognito_MyIdentityPoolAuth_Role",
      "unauthenticated": "arn:aws:iam::111111111111:role/Cognito_MyIdentityPoolUnauth_Role"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIdentityPoolRoles。

list-identity-pools

以下程式碼範例顯示如何使用 list-identity-pools。

AWS CLI

若要列出身分集區

此範例會列出身分集區。最多列出 20 個身分識別。

命令：

```
aws cognito-identity list-identity-pools --max-results 20
```

輸出：

```
{
```

```
  "IdentityPools": [
      {
          "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
          "IdentityPoolName": "MyIdentityPool"
      },
      {
          "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
          "IdentityPoolName": "AnotherIdentityPool"
      },
      {
          "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
          "IdentityPoolName": "IdentityPoolRegionA"
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListIdentityPools。

set-identity-pool-roles

以下程式碼範例顯示如何使用 set-identity-pool-roles。

AWS CLI

設定身分集區角色

下列set-identity-pool-roles範例會設定身分集區角色。

```
aws cognito-identity set-identity-pool-roles \
    --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111" \
    --roles authenticated="arn:aws:iam::111111111111:role/
Cognito_MyIdentityPoolAuth_Role"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetIdentityPoolRoles。

update-identity-pool

以下程式碼範例顯示如何使用 update-identity-pool。

AWS CLI

更新身分集區

此範例會更新身分集區。它會將名稱設定為 MyIdentityPool。它將 Cognito 新增為身分提供者。它不允許未驗證的身分。

命令：

```
aws cognito-identity update-identity-pool --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111" --identity-pool-name "MyIdentityPool" --no-allow-unauthenticated-identities --cognito-identity-providers ProviderName="cognito-idp.us-west-2.amazonaws.com/us-west-2_111111111",ClientId="3n4b5urk1ft4fl3mg5e62d9ado",ServerSideTokenCheck=false
```

輸出：

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
      {
          "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-west-2_111111111",
          "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
          "ServerSideTokenCheck": false
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateIdentityPool。

使用 的 Amazon Cognito 身分提供者範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon Cognito Identity Provider 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`add-custom-attributes`

以下程式碼範例顯示如何使用 `add-custom-attributes`。

AWS CLI

新增自訂屬性

此範例會將自訂屬性 CustomAttr1 新增至使用者集區。它是字串類型，需要最少 1 個字元，最多 15 個字元。這不是必要的。

命令：

```
aws cognito-idp add-custom-attributes --user-pool-id us-west-2_aaaaaaaaa --custom-
attributes
 Name="CustomAttr1",AttributeDataType="String",DeveloperOnlyAttribute=false,Required=false,S
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddCustomAttributes。

`admin-add-user-to-group`

以下程式碼範例顯示如何使用 `admin-add-user-to-group`。

AWS CLI

將使用者新增至群組

此範例會將使用者 Jane 新增至群組 MyGroup。

命令：

```
aws cognito-idp admin-add-user-to-group --user-pool-id us-west-2_aaaaaaaaa --
username Jane --group-name MyGroup
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminAddUserToGroup。

`admin-confirm-sign-up`

以下程式碼範例顯示如何使用 `admin-confirm-sign-up`。

AWS CLI

確認使用者註冊

此範例會確認使用者 jane@example.com。

命令：

```
aws cognito-idp admin-confirm-sign-up --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminConfirmSignUp。

admin-create-user

以下程式碼範例顯示如何使用 admin-create-user。

AWS CLI

建立使用者

下列admin-create-user範例會建立具有指定設定電子郵件地址和電話號碼的使用者。

```
aws cognito-idp admin-create-user \
    --user-pool-id us-west-2_aaaaaaaaa \
    --username diego \
    --user-attributes Name=email,Value=diego@example.com
 Name=phone_number,Value="+15555551212" \
    --message-action SUPPRESS
```

輸出：

```
{
    "User": {
        "Username": "diego",
        "Attributes": [
            {
                "Name": "sub",
                "Value": "7325c1de-b05b-4f84-b321-9adc6e61f4a2"
            },
            {
```

```
                "Name": "phone_number",
                "Value": "+15555551212"
            },
            {
                "Name": "email",
                "Value": "diego@example.com"
            }
        ],
        "UserCreateDate": 1548099495.428,
        "UserLastModifiedDate": 1548099495.428,
        "Enabled": true,
        "UserStatus": "FORCE_CHANGE_PASSWORD"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminCreateUser。

admin-delete-user-attributes

以下程式碼範例顯示如何使用 admin-delete-user-attributes。

AWS CLI

刪除使用者屬性

此範例會刪除使用者 diego@example.com 的自訂屬性 CustomAttr1。

命令：

```
aws cognito-idp admin-delete-user-attributes --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com --user-attribute-names "custom:CustomAttr1"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminDeleteUserAttributes。

admin-delete-user

以下程式碼範例顯示如何使用 admin-delete-user。

AWS CLI

若要刪除使用者

此範例會刪除使用者。

命令：

```
aws cognito-idp admin-delete-user --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminDeleteUser。

admin-disable-provider-for-user

以下程式碼範例顯示如何使用 admin-disable-provider-for-user。

AWS CLI

從本機使用者設定檔取消聯合身分使用者的連結

下列admin-disable-provider-for-user範例會將 Google 使用者與其連結的本機設定檔中斷連線。

```
aws cognito-idp admin-disable-provider-for-user \
    --user-pool-id us-west-2_EXAMPLE \
    --
user ProviderAttributeName=Cognito_Subject,ProviderAttributeValue=0000000000000000,ProviderN
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的將聯合身分使用者連結至現有使用者設定檔。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminDisableProviderForUser。

admin-disable-user

以下程式碼範例顯示如何使用 admin-disable-user。

AWS CLI

防止使用者登入

下列admin-disable-user範例可防止使用者 登入diego@example.com。

```
aws cognito-idp admin-disable-user \
```

```
    --user-pool-id us-west-2_EXAMPLE \
    --username diego@example.com
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的管理使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminDisableUser。

admin-enable-user

以下程式碼範例顯示如何使用 admin-enable-user。

AWS CLI

啟用使用者登入

下列admin-enable-user範例會啟用使用者 diego@example.com 的登入。

```
aws cognito-idp admin-enable-user \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego@example.com
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的管理使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminEnableUser。

admin-forget-device

以下程式碼範例顯示如何使用 admin-forget-device。

AWS CLI

忘記裝置

此範例會忘記使用者名稱為 jane@example.com 的裝置

命令：

```
aws cognito-idp admin-forget-device --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com --device-key us-west-2_abcd_1234-5678
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminForgetDevice。

admin-get-device

以下程式碼範例顯示如何使用 admin-get-device。

AWS CLI

取得裝置

下列admin-get-device範例顯示使用者 的一個裝置diego。

```
aws cognito-idp admin-get-device \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego \
    --device-key us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "Device": {
        "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "DeviceAttributes": [
            {
                "Name": "device_status",
                "Value": "valid"
            },
            {
                "Name": "device_name",
                "Value": "MyDevice"
            },
            {
                "Name": "dev:device_arn",
                "Value": "arn:aws:cognito-idp:us-west-2:123456789012:owner/diego.us-
west-2_EXAMPLE/device/us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
            },
            {
                "Name": "dev:device_owner",
                "Value": "diego.us-west-2_EXAMPLE"
            },
            {
                "Name": "last_ip_used",
                "Value": "192.0.2.1"
            },
            {
```

```
                "Name": "dev:device_remembered_status",
                "Value": "remembered"
            },
            {
                "Name": "dev:device_sdk",
                "Value": "aws-sdk"
            }
        ],
        "DeviceCreateDate": 1715100742.022,
        "DeviceLastModifiedDate": 1723233651.167,
        "DeviceLastAuthenticatedDate": 1715100742.0
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的在使用者集區中使用使用者裝置。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminGetDevice。

admin-get-user

以下程式碼範例顯示如何使用 admin-get-user。

AWS CLI

若要取得使用者

此範例取得使用者名稱 jane@example.com 的相關資訊。

命令：

```
aws cognito-idp admin-get-user --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com
```

輸出：

```
{
  "Username": "4320de44-2322-4620-999b-5e2e1c8df013",
  "Enabled": true,
  "UserStatus": "FORCE_CHANGE_PASSWORD",
  "UserCreateDate": 1548108509.537,
  "UserAttributes": [
      {
          "Name": "sub",
```

```
            "Value": "4320de44-2322-4620-999b-5e2e1c8df013"
        },
        {
            "Name": "email_verified",
            "Value": "true"
        },
        {
            "Name": "phone_number_verified",
            "Value": "true"
        },
        {
            "Name": "phone_number",
            "Value": "+01115551212"
        },
        {
            "Name": "email",
            "Value": "jane@example.com"
        }
    ],
    "UserLastModifiedDate": 1548108509.537
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AdminGetUser。

admin-initiate-auth

以下程式碼範例顯示如何使用 admin-initiate-auth。

AWS CLI

以管理員身分登入使用者

下列admin-initiate-auth範例登入使用者 diego@example.com。此範例也包含威脅防護的中繼資料，以及 Lambda 觸發的 ClientMetadata。使用者已設定為 TOTP MFA，並收到挑戰，要求他們先從驗證器應用程式提供程式碼，才能完成身分驗證。

```
aws cognito-idp admin-initiate-auth \
    --user-pool-id us-west-2_EXAMPLE \
    --client-id 1example23456789 \
    --auth-flow ADMIN_USER_PASSWORD_AUTH \
    --auth-parameters USERNAME=diego@example.com,PASSWORD="My@Example
$Password3!",SECRET_HASH=ExampleEncodedClientIdSecretAndUsername= \
```

```
    --context-data="{\"EncodedData\":\"abc123example\",\"HttpHeaders\":
[{\"headerName\":\"UserAgent\",\"headerValue\":\"Mozilla/5.0 (Windows NT 6.1;
 Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0\"}],\"IpAddress\":\"192.0.2.1\",
\"ServerName\":\"example.com\",\"ServerPath\":\"/login\"}" \
    --client-metadata="{\"MyExampleKey\": \"MyExampleValue\"}"
```

輸出：

```
{
    "ChallengeName": "SOFTWARE_TOKEN_MFA",
    "Session": "AYABeExample...",
    "ChallengeParameters": {
        "FRIENDLY_DEVICE_NAME": "MyAuthenticatorApp",
        "USER_ID_FOR_SRP": "diego@example.com"
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的管理員身分驗證流程。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AdminInitiateAuth。

admin-link-provider-for-user

以下程式碼範例顯示如何使用 admin-link-provider-for-user。

AWS CLI

將本機使用者連結至聯合身分使用者

下列admin-link-provider-for-user範例會將本機使用者 diego 連結至將使用 Google 進行聯合登入的使用者。

```
aws cognito-idp admin-link-provider-for-user \
    --user-pool-id us-west-2_EXAMPLE \
    --destination-user ProviderName=Cognito,ProviderAttributeValue=diego \
    --source-
user ProviderAttributeName=Cognito_Subject,ProviderAttributeValue=0000000000000000,ProviderN
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的將聯合身分使用者連結至現有使用者設定檔。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminLinkProviderForUser。

admin-list-devices

以下程式碼範例顯示如何使用 admin-list-devices。

AWS CLI

列出使用者的裝置

下列admin-list-devices範例列出使用者 diego 的裝置。

```
aws cognito-idp admin-list-devices \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego \
    --limit 1
```

輸出：

```
{
    "Devices": [
        {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceAttributes": [
                {
                    "Name": "device_status",
                    "Value": "valid"
                },
                {
                    "Name": "device_name",
                    "Value": "MyDevice"
                },
                {
                    "Name": "dev:device_arn",
                    "Value": "arn:aws:cognito-idp:us-west-2:123456789012:owner/
diego.us-west-2_EXAMPLE/device/us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
                },
                {
                    "Name": "dev:device_owner",
                    "Value": "diego.us-west-2_EXAMPLE"
                },
                {
                    "Name": "last_ip_used",
                    "Value": "192.0.2.1"
                },
```

```
                {
                    "Name": "dev:device_remembered_status",
                    "Value": "remembered"
                },
                {
                    "Name": "dev:device_sdk",
                    "Value": "aws-sdk"
                }
            ],
            "DeviceCreateDate": 1715100742.022,
            "DeviceLastModifiedDate": 1723233651.167,
            "DeviceLastAuthenticatedDate": 1715100742.0
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的在使用者集區中使用使用者裝置。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminListDevices。

admin-list-groups-for-user

以下程式碼範例顯示如何使用 admin-list-groups-for-user。

AWS CLI

列出使用者的群組

此範例列出使用者名稱 jane@example.com 的群組。

命令：

```
aws cognito-idp admin-list-groups-for-user --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com
```

輸出：

```
{
  "Groups": [
      {
          "Description": "Sample group",
          "Precedence": 1,
```

```
            "LastModifiedDate": 1548097827.125,
            "RoleArn": "arn:aws:iam::111111111111:role/SampleRole",
            "GroupName": "SampleGroup",
            "UserPoolId": "us-west-2_aaaaaaaaa",
            "CreationDate": 1548097827.125
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminListGroupsForUser。

admin-list-user-auth-events

以下程式碼範例顯示如何使用 admin-list-user-auth-events。

AWS CLI

列出使用者的授權事件

下列admin-list-user-auth-events範例列出使用者 diego 的最新使用者活動日誌事件。

```
aws cognito-idp admin-list-user-auth-events \
    --user-pool-id us-west-2_ywDJHlIfU \
    --username brcotter+050123 \
    --max-results 1
```

輸出：

```
{
    "AuthEvents": [
        {
            "EventId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "EventType": "SignIn",
            "CreationDate": 1726694203.495,
            "EventResponse": "InProgress",
            "EventRisk": {
                "RiskDecision": "AccountTakeover",
                "RiskLevel": "Medium",
                "CompromisedCredentialsDetected": false
            },
            "ChallengeResponses": [
                {
```

```
                    "ChallengeName": "Password",
                    "ChallengeResponse": "Success"
                }
            ],
            "EventContextData": {
                "IpAddress": "192.0.2.1",
                "City": "Seattle",
                "Country": "United States"
            }
        }
    ],
    "NextToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222#2024-09-18T21:16:43.495Z"
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的檢視和匯出使用者事件歷史記錄。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminListUserAuthEvents。

admin-remove-user-from-group

以下程式碼範例顯示如何使用 admin-remove-user-from-group。

AWS CLI

從群組中移除使用者

此範例會從 SampleGroup 中移除 jane@example.com。

命令：

```
aws cognito-idp admin-remove-user-from-group --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com --group-name SampleGroup
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminRemoveUserFromGroup。

admin-reset-user-password

以下程式碼範例顯示如何使用 admin-reset-user-password。

AWS CLI

重設使用者密碼

此範例會重設 diego@example.com 的密碼。

命令：

```
aws cognito-idp admin-reset-user-password --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminResetUserPassword。

admin-respond-to-auth-challenge

以下程式碼範例顯示如何使用 admin-respond-to-auth-challenge。

AWS CLI

回應身分驗證挑戰

根據您的身分驗證流程、使用者集區組態和使用者設定，有許多方法可以回應不同的身分驗證挑戰。下列admin-respond-to-auth-challenge範例提供 diego@example.com 的 TOTP MFA 代碼並完成登入。此使用者集區已開啟裝置記住，因此身分驗證結果也會傳回新的裝置金鑰。

```
aws cognito-idp admin-respond-to-auth-challenge \
    --user-pool-id us-west-2_EXAMPLE \
    --client-id 1example23456789 \
    --challenge-name SOFTWARE_TOKEN_MFA \
    --challenge-responses USERNAME=diego@example.com,SOFTWARE_TOKEN_MFA_CODE=000000
 \
    --session AYABeExample...
```

輸出：

```
{
    "ChallengeParameters": {},
    "AuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "eyJra123abcEXAMPLE",
        "IdToken": "eyJra789ghiEXAMPLE",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-ExAmPlE1"
```

```
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的管理員身分驗證流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminRespondToAuthChallenge。

admin-set-user-mfa-preference

以下程式碼範例顯示如何使用 admin-set-user-mfa-preference。

AWS CLI

設定使用者 MFA 偏好設定

此範例會設定使用者名稱 diego@example.com 的 SMS MFA 偏好設定。

命令：

```
aws cognito-idp admin-set-user-mfa-preference --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com --sms-mfa-settings Enabled=false,PreferredMfa=false
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminSetUserMfaPreference。

admin-set-user-password

以下程式碼範例顯示如何使用 admin-set-user-password。

AWS CLI

將使用者密碼設定為管理員

下列admin-set-user-password範例會永久設定 diego@example.com 的密碼。

```
aws cognito-idp admin-set-user-password \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego@example.com \
    --password MyExamplePassword1! \
    --permanent
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的密碼、密碼復原和密碼政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminSetUserPassword。

admin-set-user-settings

以下程式碼範例顯示如何使用 admin-set-user-settings。

AWS CLI

設定使用者設定

此範例會將使用者名稱 diego@example.com 的 MFA 交付偏好設定設定為 EMAIL。

命令：

```
aws cognito-idp admin-set-user-settings --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com --mfa-options DeliveryMedium=EMAIL
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminSetUserSettings。

admin-update-auth-event-feedback

以下程式碼範例顯示如何使用 admin-update-auth-event-feedback。

AWS CLI

提供授權事件的意見回饋

此範例會將 event-id 識別的授權事件的意見回饋值設定為 Valid。

命令：

```
aws cognito-idp admin-update-auth-event-feedback --user-pool-id us-west-2_aaaaaaaaa
 --username diego@example.com --event-id c2c2cf89-c0d3-482d-aba6-99d78a5b0bfe --
feedback-value Valid
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminUpdateAuthEventFeedback。

admin-update-device-status

以下程式碼範例顯示如何使用 admin-update-device-status。

AWS CLI

更新裝置狀態

此範例會將裝置金鑰識別的裝置記住狀態設定為 not_remembered。

命令：

```
aws cognito-idp admin-update-device-status --user-pool-id us-west-2_aaaaaaaaa
 --username diego@example.com --device-key xxxx  --device-remembered-
status not_remembered
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminUpdateDeviceStatus。

admin-update-user-attributes

以下程式碼範例顯示如何使用 admin-update-user-attributes。

AWS CLI

更新使用者屬性

此範例會更新使用者 diego@example.com 的自訂使用者屬性 CustomAttr1。

命令：

```
aws cognito-idp admin-update-user-attributes --user-pool-id us-
west-2_aaaaaaaaa --username diego@example.com --user-attributes
 Name="custom:CustomAttr1",Value="Purple"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminUpdateUserAttributes。

admin-user-global-sign-out

以下程式碼範例顯示如何使用 admin-user-global-sign-out。

AWS CLI

以管理員身分登出使用者

下列admin-user-global-sign-out範例會將使用者登出 diego@example.com。

```
aws cognito-idp admin-user-global-sign-out \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego@example.com
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的使用使用者集區進行身分驗證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdminUserGlobalSignOut。

associate-software-token

以下程式碼範例顯示如何使用 associate-software-token。

AWS CLI

為 MFA 驗證器應用程式產生私密金鑰

下列associate-software-token範例會為已登入並收到存取權杖的使用者產生 TOTP 私有金鑰。產生的私有金鑰可以手動輸入到驗證器應用程式中，或者應用程式可以將它轉譯為使用者可以掃描的 QR 碼。

```
aws cognito-idp associate-software-token \
    --access-token eyJra456defEXAMPLE
```

輸出：

```
{
    "SecretCode": "QWERTYUIOP123456EXAMPLE"
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的 TOTP 軟體字符 MFA。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateSoftwareToken。

change-password

以下程式碼範例顯示如何使用 change-password。

AWS CLI

變更密碼

此範例會變更密碼。

命令：

```
aws cognito-idp change-password --previous-password OldPassword --proposed-
password NewPassword --access-token ACCESS_TOKEN
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ChangePassword。

confirm-device

以下程式碼範例顯示如何使用 confirm-device。

AWS CLI

確認使用者裝置

下列confirm-device範例會為目前使用者新增記住的裝置。

```
aws cognito-idp confirm-device \
   --access-token eyJra456defEXAMPLE \
   --device-key us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
   --device-secret-verifier-
config PasswordVerifier=TXlWZXJpZmllclN0cmluZw,Salt=TXlTUlBTYWx0
```

輸出：

```
{
     "UserConfirmationNecessary": false
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的在使用者集區中使用使用者裝置。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ConfirmDevice。

confirm-forgot-password

以下程式碼範例顯示如何使用 confirm-forgot-password。

AWS CLI

確認忘記的密碼

此範例會確認使用者名稱 diego@example.com 的忘記密碼。

命令：

```
aws cognito-idp confirm-forgot-password --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username=diego@example.com --password PASSWORD --confirmation-code CONF_CODE
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ConfirmForgotPassword。

`confirm-sign-up`

以下程式碼範例顯示如何使用 `confirm-sign-up`。

AWS CLI

若要確認註冊

此範例會確認註冊使用者名稱 diego@example.com。

命令：

```
aws cognito-idp confirm-sign-up --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username=diego@example.com --confirmation-code CONF_CODE
```

- 如需 API 詳細資訊，請參閱《AWS CLI API 參考》中的 ConfirmSignUp。

`create-group`

以下程式碼範例顯示如何使用 `create-group`。

AWS CLI

建立群組

此範例會建立具有描述的群組。

命令：

```
aws cognito-idp create-group --user-pool-id us-west-2_aaaaaaaaa --group-
name MyNewGroup --description "New group."
```

輸出：

```
{
  "Group": {
      "GroupName": "MyNewGroup",
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "Description": "New group.",
      "LastModifiedDate": 1548270073.795,
      "CreationDate": 1548270073.795
  }
}
```

建立具有角色和優先順序的群組

此範例會建立具有描述的群組。它還包含角色和優先順序。

命令：

```
aws cognito-idp create-group --user-pool-id us-west-2_aaaaaaaaa --group-
name MyNewGroupWithRole --description "New group with a role." --role-
arn arn:aws:iam::111111111111:role/MyNewGroupRole --precedence 2
```

輸出：

```
{
  "Group": {
      "GroupName": "MyNewGroupWithRole",
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "Description": "New group with a role.",
      "RoleArn": "arn:aws:iam::111111111111:role/MyNewGroupRole",
      "Precedence": 2,
      "LastModifiedDate": 1548270211.761,
      "CreationDate": 1548270211.761
  }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateGroup。

create-identity-provider

以下程式碼範例顯示如何使用 create-identity-provider。

AWS CLI

範例 1：使用中繼資料 URL 建立使用者集區 SAML 身分提供者 (IdP)

下列create-identity-provider範例會建立新的 SAML IdP，其中包含來自公有 URL、屬性映射和兩個識別符的中繼資料。

```
aws cognito-idp create-identity-provider \
    --user-pool-id us-west-2_EXAMPLE \
    --provider-name MySAML \
    --provider-type SAML \
    --provider-
details IDPInit=true,IDPSignout=true,EncryptedResponses=true,MetadataURL=https://
auth.example.com/sso/saml/metadata,RequestSigningAlgorithm=rsa-sha256 \
    --attribute-mapping email=emailaddress,phone_number=phone,custom:111=department
 \
    --idp-identifiers CorpSAML WestSAML
```

輸出：

```
{
    "IdentityProvider": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ProviderName": "MySAML",
        "ProviderType": "SAML",
        "ProviderDetails": {
            "ActiveEncryptionCertificate": "MIICvTCCAaEXAMPLE",
            "EncryptedResponses": "true",
            "IDPInit": "true",
            "IDPSignout": "true",
            "MetadataURL": "https://auth.example.com/sso/saml/metadata",
            "RequestSigningAlgorithm": "rsa-sha256",
            "SLORedirectBindingURI": "https://auth.example.com/slo/saml",
            "SSORedirectBindingURI": "https://auth.example.com/sso/saml"
        },
        "AttributeMapping": {
            "custom:111": "department",
            "emailaddress": "email",
            "phone": "phone_number"
        },
        "IdpIdentifiers": [
            "CorpSAML",
            "WestSAML"
```

```
        ],
        "LastModifiedDate": 1726853833.977,
        "CreationDate": 1726853833.977
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的透過第三方新增使用者集區登入。

範例 2：使用中繼資料檔案建立使用者集區 SAML 身分提供者 (IdP)

下列create-identity-provider範例會建立新的 SAML IdP，其中包含來自檔案、屬性映射和兩個識別符的中繼資料。檔案語法在 --provider-details 參數中的作業系統之間可能不同。為此操作建立 JSON 輸入檔案最簡單：

```
aws cognito-idp create-identity-provider \
    --cli-input-json file://.\SAML-identity-provider.json
```

SAML-identity-provider.json 的內容：

```
{
    "AttributeMapping": {
        "email" : "idp_email",
        "email_verified" : "idp_email_verified"
    },
    "IdpIdentifiers": [ "platform" ],
    "ProviderDetails": {
        "MetadataFile": "<md:EntityDescriptor xmlns:md=
\"urn:oasis:names:tc:SAML:2.0:metadata\" entityID=\"http://www.example.com/
sso\"><md:IDPSSODescriptor WantAuthnRequestsSigned=\"false\"
 protocolSupportEnumeration=\"urn:oasis:names:tc:SAML:2.0:protocol
\"><md:KeyDescriptor use=\"signing\"><ds:KeyInfo xmlns:ds=\"http://
www.w3.org/2000/09/xmldsig#
\"><ds:X509Data><ds:X509Certificate>[IDP_CERTIFICATE_DATA]</ds:X509Certificate></
ds:X509Data></ds:KeyInfo></md:KeyDescriptor><md:SingleLogoutService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST\" Location=
\"https://www.example.com/slo/saml\"/><md:SingleLogoutService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect\" Location=\"https://www.example.com/slo/saml\"/
><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</
md:NameIDFormat><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress</md:NameIDFormat><md:SingleSignOnService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST\" Location=
```

```
\"https://www.example.com/sso/saml\"/><md:SingleSignOnService Binding=
\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect\" Location=\"https://
www.example.com/sso/saml\"/></md:IDPSSODescriptor></md:EntityDescriptor>",
        "IDPSignout" : "true",
        "RequestSigningAlgorithm" : "rsa-sha256",
        "EncryptedResponses" : "true",
        "IDPInit" : "true"
    },
    "ProviderName": "MySAML2",
    "ProviderType": "SAML",
    "UserPoolId": "us-west-2_EXAMPLE"
}
```

輸出：

```
{
    "IdentityProvider": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ProviderName": "MySAML2",
        "ProviderType": "SAML",
        "ProviderDetails": {
            "ActiveEncryptionCertificate":
 "[USER_POOL_ENCRYPTION_CERTIFICATE_DATA]",
            "EncryptedResponses": "true",
            "IDPInit": "true",
            "IDPSignout": "true",
            "MetadataFile": "<md:EntityDescriptor xmlns:md=
\"urn:oasis:names:tc:SAML:2.0:metadata\" entityID=\"http://www.example.com/
sso\"><md:IDPSSODescriptor WantAuthnRequestsSigned=\"false\"
 protocolSupportEnumeration=\"urn:oasis:names:tc:SAML:2.0:protocol
\"><md:KeyDescriptor use=\"signing\"><ds:KeyInfo xmlns:ds=\"http://
www.w3.org/2000/09/xmldsig#
\"><ds:X509Data><ds:X509Certificate>[IDP_CERTIFICATE_DATA]</ds:X509Certificate></
ds:X509Data></ds:KeyInfo></md:KeyDescriptor><md:SingleLogoutService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST\" Location=
\"https://www.example.com/slo/saml\"/><md:SingleLogoutService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect\" Location=\"https://www.example.com/slo/saml\"/
><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</
md:NameIDFormat><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress</md:NameIDFormat><md:SingleSignOnService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST\" Location=
\"https://www.example.com/sso/saml\"/><md:SingleSignOnService Binding=
```

```
\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect\" Location=\"https://
www.example.com/sso/saml\"/></md:IDPSSODescriptor></md:EntityDescriptor>",
            "RequestSigningAlgorithm": "rsa-sha256",
            "SLORedirectBindingURI": "https://www.example.com/slo/saml",
            "SSORedirectBindingURI": "https://www.example.com/sso/saml"
        },
        "AttributeMapping": {
            "email": "idp_email",
            "email_verified": "idp_email_verified"
        },
        "IdpIdentifiers": [
            "platform"
        ],
        "LastModifiedDate": 1726855290.731,
        "CreationDate": 1726855290.731
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的透過第三方新增使用者集區登入。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateIdentityProvider。

create-resource-server

以下程式碼範例顯示如何使用 create-resource-server。

AWS CLI

建立使用者集區用戶端

下列create-resource-server範例會使用自訂範圍建立新的資源伺服器。

```
aws cognito-idp create-resource-server \
    --user-pool-id us-west-2_EXAMPLE \
    --identifier solar-system-data \
    --name "Solar system object tracker" \
    --scopes ScopeName=sunproximity.read,ScopeDescription="Distance in AU from Sol"
 ScopeName=asteroids.add,ScopeDescription="Enter a new asteroid"
```

輸出：

```
{
    "ResourceServer": {
```

```
        "UserPoolId": "us-west-2_EXAMPLE",
        "Identifier": "solar-system-data",
        "Name": "Solar system object tracker",
        "Scopes": [
            {
                "ScopeName": "sunproximity.read",
                "ScopeDescription": "Distance in AU from Sol"
            },
            {
                "ScopeName": "asteroids.add",
                "ScopeDescription": "Enter a new asteroid"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的範圍、M2M 和具有資源伺服器的 APIs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateResourceServer。

create-user-import-job

以下程式碼範例顯示如何使用 create-user-import-job。

AWS CLI

建立使用者匯入任務

此範例會建立名為 MyImportJob 的使用者匯入任務。

如需匯入使用者的詳細資訊，請參閱從 CSV 檔案將使用者匯入使用者集區。

命令：

```
aws cognito-idp create-user-import-job --user-pool-id us-west-2_aaaaaaaaa --
job-name MyImportJob --cloud-watch-logs-role-arn arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole
```

輸出：

```
{
```

```
  "UserImportJob": {
      "JobName": "MyImportJob",
      "JobId": "import-qQ0DCt2fRh",
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "PreSignedUrl": "PRE_SIGNED_URL",
      "CreationDate": 1548271795.471,
      "Status": "Created",
      "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole",
      "ImportedUsers": 0,
      "SkippedUsers": 0,
      "FailedUsers": 0
  }
}
```

使用預先簽章的 URL，以 curl 上傳 .csv 檔案：

命令：

```
curl -v -T "PATH_TO_CSV_FILE" -H "x-amz-server-side-encryption:aws:kms"
 "PRE_SIGNED_URL"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateUserImportJob。

create-user-pool-client

以下程式碼範例顯示如何使用 create-user-pool-client。

AWS CLI

建立使用者集區用戶端

下列create-user-pool-client範例會建立新的使用者集區用戶端，其中包含用戶端秘密、明確的讀取和寫入屬性、使用使用者名稱密碼和 SRP 流程登入、使用三個 IdPs 登入、存取 OAuth 範圍的子集、PinPoint 分析，以及延伸的身分驗證工作階段有效性。

```
aws cognito-idp create-user-pool-client \
    --user-pool-id us-west-2_EXAMPLE \
    --client-name MyTestClient \
    --generate-secret \
    --refresh-token-validity 10 \
    --access-token-validity 60 \
```

```
    --id-token-validity 60 \
    --token-validity-units AccessToken=minutes,IdToken=minutes,RefreshToken=days \
    --read-attributes email phone_number email_verified phone_number_verified \
    --write-attributes email phone_number \
    --explicit-auth-
flows ALLOW_USER_PASSWORD_AUTH ALLOW_USER_SRP_AUTH ALLOW_REFRESH_TOKEN_AUTH \
    --supported-identity-providers Google Facebook MyOIDC \
    --callback-urls https://www.amazon.com https://example.com http://
localhost:8001 myapp://example \
    --allowed-o-auth-flows code implicit \
    --allowed-o-auth-scopes openid profile aws.cognito.signin.user.admin solar-
system-data/asteroids.add \
    --allowed-o-auth-flows-user-pool-client \
    --analytics-configuration ApplicationArn=arn:aws:mobiletargeting:us-
west-2:767671399759:apps/thisisanexamplepinpointapplicationid,UserDataShared=TRUE \
    --prevent-user-existence-errors ENABLED \
    --enable-token-revocation \
    --enable-propagate-additional-user-context-data \
    --auth-session-validity 4
```

輸出：

```
{
    "UserPoolClient": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ClientName": "MyTestClient",
        "ClientId": "123abc456defEXAMPLE",
        "ClientSecret": "this1234is5678my91011example1213client1415secret",
        "LastModifiedDate": 1726788459.464,
        "CreationDate": 1726788459.464,
        "RefreshTokenValidity": 10,
        "AccessTokenValidity": 60,
        "IdTokenValidity": 60,
        "TokenValidityUnits": {
            "AccessToken": "minutes",
            "IdToken": "minutes",
            "RefreshToken": "days"
        },
        "ReadAttributes": [
            "email_verified",
            "phone_number_verified",
            "phone_number",
            "email"
```

```
        ],
        "WriteAttributes": [
            "phone_number",
            "email"
        ],
        "ExplicitAuthFlows": [
            "ALLOW_USER_PASSWORD_AUTH",
            "ALLOW_USER_SRP_AUTH",
            "ALLOW_REFRESH_TOKEN_AUTH"
        ],
        "SupportedIdentityProviders": [
            "Google",
            "MyOIDC",
            "Facebook"
        ],
        "CallbackURLs": [
            "https://example.com",
            "https://www.amazon.com",
            "myapp://example",
            "http://localhost:8001"
        ],
        "AllowedOAuthFlows": [
            "implicit",
            "code"
        ],
        "AllowedOAuthScopes": [
            "aws.cognito.signin.user.admin",
            "openid",
            "profile",
            "solar-system-data/asteroids.add"
        ],
        "AllowedOAuthFlowsUserPoolClient": true,
        "AnalyticsConfiguration": {
            "ApplicationArn": "arn:aws:mobiletargeting:us-west-2:123456789012:apps/
thisisanexamplepinpointapplicationid",
            "RoleArn": "arn:aws:iam::123456789012:role/aws-service-role/cognito-
idp.amazonaws.com/AWSServiceRoleForAmazonCognitoIdp",
            "UserDataShared": true
        },
        "PreventUserExistenceErrors": "ENABLED",
        "EnableTokenRevocation": true,
        "EnablePropagateAdditionalUserContextData": true,
        "AuthSessionValidity": 4
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的應用程式特定設定與應用程式用戶端。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateUserPoolClient。

create-user-pool-domain

以下程式碼範例顯示如何使用 create-user-pool-domain。

AWS CLI

範例 1：建立使用者集區網域

下列create-user-pool-domain範例會建立新的自訂網域。

```
aws cognito-idp create-user-pool-domain \
    --user-pool-id us-west-2_EXAMPLE \
    --domain auth.example.com \
    --custom-domain-config CertificateArn=arn:aws:acm:us-
east-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

輸出：

```
{
    "CloudFrontDomain": "example1domain.cloudfront.net"
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的設定使用者集區網域。

範例 2：建立使用者集區網域

下列create-user-pool-domain範例會使用服務擁有的字首建立新的網域。

```
aws cognito-idp create-user-pool-domain \
    --user-pool-id us-west-2_EXAMPLE2 \
    --domain mydomainprefix
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的設定使用者集區網域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateUserPoolDomain。

create-user-pool

以下程式碼範例顯示如何使用 create-user-pool。

AWS CLI

建立最低限度設定的使用者集區

此範例會使用預設值建立名為 MyUserPool 的使用者集區。沒有必要的屬性，也沒有應用程式用戶端。MFA 和進階安全性已停用。

命令：

```
aws cognito-idp create-user-pool --pool-name MyUserPool
```

輸出：

```
{
  "UserPool": {
      "SchemaAttributes": [
          {
              "Name": "sub",
              "StringAttributeConstraints": {
                  "MinLength": "1",
                  "MaxLength": "2048"
              },
              "DeveloperOnlyAttribute": false,
              "Required": true,
              "AttributeDataType": "String",
              "Mutable": false
          },
          {
              "Name": "name",
              "StringAttributeConstraints": {
                  "MinLength": "0",
                  "MaxLength": "2048"
              },
              "DeveloperOnlyAttribute": false,
              "Required": false,
              "AttributeDataType": "String",
              "Mutable": true
          },
          {
              "Name": "given_name",
```

```
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "family_name",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "middle_name",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "nickname",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "preferred_username",
```

```
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "profile",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "picture",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "website",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "email",
```

```
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "Name": "email_verified",
            "Mutable": true
        },
        {
            "Name": "gender",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "birthdate",
            "StringAttributeConstraints": {
                "MinLength": "10",
                "MaxLength": "10"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "zoneinfo",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
```

```
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "locale",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "phone_number",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "Name": "phone_number_verified",
            "Mutable": true
        },
        {
            "Name": "address",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
```

```
            },
            {
                "Name": "updated_at",
                "NumberAttributeConstraints": {
                    "MinValue": "0"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "Number",
                "Mutable": true
            }
        ],
        "MfaConfiguration": "OFF",
        "Name": "MyUserPool",
        "LastModifiedDate": 1547833345.777,
        "AdminCreateUserConfig": {
            "UnusedAccountValidityDays": 7,
            "AllowAdminCreateUserOnly": false
        },
        "EmailConfiguration": {},
        "Policies": {
            "PasswordPolicy": {
                "RequireLowercase": true,
                "RequireSymbols": true,
                "RequireNumbers": true,
                "MinimumLength": 8,
                "RequireUppercase": true
            }
        },
        "CreationDate": 1547833345.777,
        "EstimatedNumberOfUsers": 0,
        "Id": "us-west-2_aaaaaaaaa",
        "LambdaConfig": {}
    }
}
```

用兩個必要屬性建立新的使用者集區

此範例會建立使用者集區 MyUserPool。集區設定為接受電子郵件作為使用者名稱屬性。它也會使用 Amazon Simple Email Service，將電子郵件來源地址設定為經過驗證的地址。

命令：

```
aws cognito-idp create-user-pool --pool-name MyUserPool --username-
attributes "email" --email-configuration=SourceArn="arn:aws:ses:us-
east-1:111111111111:identity/
jane@example.com",ReplyToEmailAddress="jane@example.com"
```

輸出：

```
{
  "UserPool": {
      "SchemaAttributes": [
          {
              "Name": "sub",
              "StringAttributeConstraints": {
                  "MinLength": "1",
                  "MaxLength": "2048"
              },
              "DeveloperOnlyAttribute": false,
              "Required": true,
              "AttributeDataType": "String",
              "Mutable": false
          },
          {
              "Name": "name",
              "StringAttributeConstraints": {
                  "MinLength": "0",
                  "MaxLength": "2048"
              },
              "DeveloperOnlyAttribute": false,
              "Required": false,
              "AttributeDataType": "String",
              "Mutable": true
          },
          {
              "Name": "given_name",
              "StringAttributeConstraints": {
                  "MinLength": "0",
                  "MaxLength": "2048"
              },
              "DeveloperOnlyAttribute": false,
              "Required": false,
              "AttributeDataType": "String",
              "Mutable": true
          },
```

```
            {
                "Name": "family_name",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "middle_name",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "nickname",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "preferred_username",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
```

```
            {
                "Name": "profile",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "picture",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "website",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "email",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
```

```
            {
                "AttributeDataType": "Boolean",
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "Name": "email_verified",
                "Mutable": true
            },
            {
                "Name": "gender",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "birthdate",
                "StringAttributeConstraints": {
                    "MinLength": "10",
                    "MaxLength": "10"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "zoneinfo",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "locale",
                "StringAttributeConstraints": {
                    "MinLength": "0",
```

```
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "phone_number",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "Name": "phone_number_verified",
            "Mutable": true
        },
        {
            "Name": "address",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "updated_at",
            "NumberAttributeConstraints": {
                "MinValue": "0"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "Number",
```

```
                "Mutable": true
            }
        ],
        "MfaConfiguration": "OFF",
        "Name": "MyUserPool",
        "LastModifiedDate": 1547837788.189,
        "AdminCreateUserConfig": {
            "UnusedAccountValidityDays": 7,
            "AllowAdminCreateUserOnly": false
        },
        "EmailConfiguration": {
            "ReplyToEmailAddress": "jane@example.com",
            "SourceArn": "arn:aws:ses:us-east-1:111111111111:identity/
jane@example.com"
        },
        "Policies": {
            "PasswordPolicy": {
                "RequireLowercase": true,
                "RequireSymbols": true,
                "RequireNumbers": true,
                "MinimumLength": 8,
                "RequireUppercase": true
            }
        },
        "UsernameAttributes": [
            "email"
        ],
        "CreationDate": 1547837788.189,
        "EstimatedNumberOfUsers": 0,
        "Id": "us-west-2_aaaaaaaaa",
        "LambdaConfig": {}
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateUserPool。

delete-group

以下程式碼範例顯示如何使用 delete-group。

AWS CLI

刪除群組

此範例會刪除群組。

命令：

```
aws cognito-idp delete-group --user-pool-id us-west-2_aaaaaaaaa  --group-
name MyGroupName
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteGroup。

delete-identity-provider

以下程式碼範例顯示如何使用 delete-identity-provider。

AWS CLI

刪除身分提供者

此範例會刪除身分提供者。

命令：

```
aws cognito-idp delete-identity-provider --user-pool-id us-west-2_aaaaaaaaa --
provider-name Facebook
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteIdentityProvider。

delete-resource-server

以下程式碼範例顯示如何使用 delete-resource-server。

AWS CLI

刪除資源伺服器

此範例會刪除名為 weather.example.com 的資源伺服器。

命令：

```
aws cognito-idp delete-resource-server --user-pool-id us-west-2_aaaaaaaaa --
identifier weather.example.com
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteResourceServer。

delete-user-attributes

以下程式碼範例顯示如何使用 delete-user-attributes。

AWS CLI

刪除使用者屬性

下列delete-user-attributes範例會從目前登入的使用者中刪除自訂屬性 "custom：attribute"。

```
aws cognito-idp delete-user-attributes \
    --access-token ACCESS_TOKEN \
    --user-attribute-names "custom:department"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的使用使用者屬性。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteUserAttributes。

delete-user-pool-client

以下程式碼範例顯示如何使用 delete-user-pool-client。

AWS CLI

刪除使用者集區用戶端

此範例會刪除使用者集區用戶端。

命令：

```
aws cognito-idp delete-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --client-id 38fjsnc484p94kpqsnet7mpld0
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteUserPoolClient。

delete-user-pool-domain

以下程式碼範例顯示如何使用 delete-user-pool-domain。

AWS CLI

刪除使用者集區網域

下列delete-user-pool-domain範例會刪除名為 的使用者集區網域 my-domain

```
aws cognito-idp delete-user-pool-domain \
    --user-pool-id us-west-2_aaaaaaaaa \
    --domain my-domain
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteUserPoolDomain。

delete-user-pool

以下程式碼範例顯示如何使用 delete-user-pool。

AWS CLI

刪除使用者集區

此範例使用使用者集區 ID us-west-2_aaaaaaaa 刪除使用者集區。

命令：

```
aws cognito-idp delete-user-pool --user-pool-id us-west-2_aaaaaaaaa
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteUserPool。

delete-user

以下程式碼範例顯示如何使用 delete-user。

AWS CLI

若要刪除使用者

此範例會刪除使用者。

命令：

```
aws cognito-idp delete-user --access-token ACCESS_TOKEN
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteUser。

describe-identity-provider

以下程式碼範例顯示如何使用 describe-identity-provider。

AWS CLI

描述身分提供者

此範例說明名為 Facebook 的身分提供者。

命令：

```
aws cognito-idp describe-identity-provider --user-pool-id us-west-2_aaaaaaaaa --
provider-name Facebook
```

輸出：

```
{
  "IdentityProvider": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "ProviderName": "Facebook",
      "ProviderType": "Facebook",
      "ProviderDetails": {
          "attributes_url": "https://graph.facebook.com/me?fields=",
          "attributes_url_add_attributes": "true",
          "authorize_scopes": myscope",
          "authorize_url": "https://www.facebook.com/v2.9/dialog/oauth",
          "client_id": "11111",
          "client_secret": "11111",
          "token_request_method": "GET",
          "token_url": "https://graph.facebook.com/v2.9/oauth/access_token"
      },
      "AttributeMapping": {
          "username": "id"
      },
      "IdpIdentifiers": [],
      "LastModifiedDate": 1548105901.736,
      "CreationDate": 1548105901.736
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeIdentityProvider。

describe-resource-server

以下程式碼範例顯示如何使用 describe-resource-server。

AWS CLI

描述資源伺服器

此範例說明資源伺服器 weather.example.com。

命令：

```
aws cognito-idp describe-resource-server --user-pool-id us-west-2_aaaaaaaaa --
identifier weather.example.com
```

輸出：

```
{
  "ResourceServer": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "Identifier": "weather.example.com",
      "Name": "Weather",
      "Scopes": [
          {
              "ScopeName": "weather.update",
              "ScopeDescription": "Update weather forecast"
          },
          {
              "ScopeName": "weather.read",
              "ScopeDescription": "Read weather forecasts"
          },
          {
              "ScopeName": "weather.delete",
              "ScopeDescription": "Delete a weather forecast"
          }
      ]
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeResourceServer。

describe-risk-configuration

以下程式碼範例顯示如何使用 describe-risk-configuration。

AWS CLI

描述風險組態

此範例說明與集區 us-west-2_aaaaaaaaa 相關聯的風險組態。

命令：

```
aws cognito-idp describe-risk-configuration --user-pool-id us-west-2_aaaaaaaaa
```

輸出：

```
{
  "RiskConfiguration": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "CompromisedCredentialsRiskConfiguration": {
          "EventFilter": [
              "SIGN_IN",
              "SIGN_UP",
              "PASSWORD_CHANGE"
          ],
          "Actions": {
              "EventAction": "BLOCK"
          }
      },
      "AccountTakeoverRiskConfiguration": {
          "NotifyConfiguration": {
              "From": "diego@example.com",
              "ReplyTo": "diego@example.com",
              "SourceArn": "arn:aws:ses:us-east-1:111111111111:identity/
diego@example.com",
              "BlockEmail": {
                  "Subject": "Blocked sign-in attempt",
                  "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
 email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
 blocked an unrecognized sign-in to your account with this information:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
```

```
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
                    "TextBody": "We blocked an unrecognized sign-in to your account
 with this information:\nTime: {login-time}\nDevice: {device-name}\nLocation:
 {city}, {country}\nIf this sign-in was not by you, you should change your password
 and notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you,
 you can follow {one-click-link-valid} to let us know"
                },
                "NoActionEmail": {
                    "Subject": "New sign-in attempt",
                    "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
 email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
 observed an unrecognized sign-in to your account with this information:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
                    "TextBody": "We observed an unrecognized sign-in to your account
 with this information:\nTime: {login-time}\nDevice: {device-name}\nLocation:
 {city}, {country}\nIf this sign-in was not by you, you should change your password
 and notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you,
 you can follow {one-click-link-valid} to let us know"
                },
                "MfaEmail": {
                    "Subject": "New sign-in attempt",
                    "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML email
 context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We required
 you to use multi-factor authentication for the following sign-in attempt:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
                    "TextBody": "We required you to use multi-factor authentication
 for the following sign-in attempt:\nTime: {login-time}\nDevice: {device-
name}\nLocation: {city}, {country}\nIf this sign-in was not by you, you should
 change your password and notify us by clicking on {one-click-link-invalid}\nIf this
 sign-in was by you, you can follow {one-click-link-valid} to let us know"
                }
            },
            "Actions": {
                "LowAction": {
                    "Notify": true,
```

```
                "EventAction": "NO_ACTION"
            },
            "MediumAction": {
                "Notify": true,
                "EventAction": "MFA_IF_CONFIGURED"
            },
            "HighAction": {
                "Notify": true,
                "EventAction": "MFA_IF_CONFIGURED"
            }
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRiskConfiguration。

describe-user-import-job

以下程式碼範例顯示如何使用 describe-user-import-job。

AWS CLI

描述使用者匯入任務

此範例說明使用者輸入任務。

如需匯入使用者的詳細資訊，請參閱從 CSV 檔案將使用者匯入使用者集區。

命令：

```
aws cognito-idp describe-user-import-job --user-pool-id us-west-2_aaaaaaaaa --job-id import-TZqNQvDRnW
```

輸出：

```
{
  "UserImportJob": {
      "JobName": "import-Test1",
      "JobId": "import-TZqNQvDRnW",
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "PreSignedUrl": "PRE_SIGNED URL",
      "CreationDate": 1548271708.512,
```

```
      "Status": "Created",
      "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole",
      "ImportedUsers": 0,
      "SkippedUsers": 0,
      "FailedUsers": 0
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeUserImportJob。

describe-user-pool-client

以下程式碼範例顯示如何使用 describe-user-pool-client。

AWS CLI

描述使用者集區用戶端

此範例說明使用者集區用戶端。

命令：

```
aws cognito-idp describe-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --
client-id 38fjsnc484p94kpqsnet7mpld0
```

輸出：

```
{
  "UserPoolClient": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "ClientName": "MyApp",
      "ClientId": "38fjsnc484p94kpqsnet7mpld0",
      "ClientSecret": "CLIENT_SECRET",
      "LastModifiedDate": 1548108676.163,
      "CreationDate": 1548108676.163,
      "RefreshTokenValidity": 30,
      "ReadAttributes": [
          "address",
          "birthdate",
          "custom:CustomAttr1",
          "custom:CustomAttr2",
          "email",
```

```
            "email_verified",
            "family_name",
            "gender",
            "given_name",
            "locale",
            "middle_name",
            "name",
            "nickname",
            "phone_number",
            "phone_number_verified",
            "picture",
            "preferred_username",
            "profile",
            "updated_at",
            "website",
            "zoneinfo"
        ],
        "WriteAttributes": [
            "address",
            "birthdate",
            "custom:CustomAttr1",
            "custom:CustomAttr2",
            "email",
            "family_name",
            "gender",
            "given_name",
            "locale",
            "middle_name",
            "name",
            "nickname",
            "phone_number",
            "picture",
            "preferred_username",
            "profile",
            "updated_at",
            "website",
            "zoneinfo"
        ],
        "ExplicitAuthFlows": [
            "ADMIN_NO_SRP_AUTH",
            "USER_PASSWORD_AUTH"
        ],
        "AllowedOAuthFlowsUserPoolClient": false
    }
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeUserPoolClient。

describe-user-pool-domain

以下程式碼範例顯示如何使用 describe-user-pool-domain。

AWS CLI

描述使用者集區用戶端

此範例描述名為 my-domain 的使用者集區網域。

命令：

```
aws cognito-idp describe-user-pool-domain --domain my-domain
```

輸出：

```
{
  "DomainDescription": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "AWSAccountId": "111111111111",
      "Domain": "my-domain",
      "S3Bucket": "aws-cognito-prod-pdx-assets",
      "CloudFrontDistribution": "aaaaaaaaaaaaa.cloudfront.net",
      "Version": "20190128175402",
      "Status": "ACTIVE",
      "CustomDomainConfig": {}
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeUserPoolDomain。

describe-user-pool

以下程式碼範例顯示如何使用 describe-user-pool。

AWS CLI

描述使用者集區

下列範例說明使用者集區 ID us-west-2_EXAMPLE 的使用者集區。

```
aws cognito-idp describe-user-pool \
    --user-pool-id us-west-2_EXAMPLE
```

輸出：

```
{
    "UserPool": {
        "Id": "us-west-2_EXAMPLE",
        "Name": "MyUserPool",
        "Policies": {
            "PasswordPolicy": {
                "MinimumLength": 8,
                "RequireUppercase": true,
                "RequireLowercase": true,
                "RequireNumbers": true,
                "RequireSymbols": true,
                "TemporaryPasswordValidityDays": 1
            }
        },
        "DeletionProtection": "ACTIVE",
        "LambdaConfig": {
            "PreSignUp": "arn:aws:lambda:us-
west-2:123456789012:function:MyPreSignUpFunction",
            "CustomMessage": "arn:aws:lambda:us-
west-2:123456789012:function:MyCustomMessageFunction",
            "PostConfirmation": "arn:aws:lambda:us-
west-2:123456789012:function:MyPostConfirmationFunction",
            "PreAuthentication": "arn:aws:lambda:us-
west-2:123456789012:function:MyPreAuthenticationFunction",
            "PostAuthentication": "arn:aws:lambda:us-
west-2:123456789012:function:MyPostAuthenticationFunction",
            "DefineAuthChallenge": "arn:aws:lambda:us-
west-2:123456789012:function:MyDefineAuthChallengeFunction",
            "CreateAuthChallenge": "arn:aws:lambda:us-
west-2:123456789012:function:MyCreateAuthChallengeFunction",
            "VerifyAuthChallengeResponse": "arn:aws:lambda:us-
west-2:123456789012:function:MyVerifyAuthChallengeFunction",
            "PreTokenGeneration": "arn:aws:lambda:us-
west-2:123456789012:function:MyPreTokenGenerationFunction",
            "UserMigration": "arn:aws:lambda:us-
west-2:123456789012:function:MyMigrateUserFunction",
```

```
            "PreTokenGenerationConfig": {
                "LambdaVersion": "V2_0",
                "LambdaArn": "arn:aws:lambda:us-
west-2:123456789012:function:MyPreTokenGenerationFunction"
            },
            "CustomSMSSender": {
                "LambdaVersion": "V1_0",
                "LambdaArn": "arn:aws:lambda:us-
west-2:123456789012:function:MyCustomSMSSenderFunction"
            },
            "CustomEmailSender": {
                "LambdaVersion": "V1_0",
                "LambdaArn": "arn:aws:lambda:us-
west-2:123456789012:function:MyCustomEmailSenderFunction"
            },
            "KMSKeyID": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222"
        },
        "LastModifiedDate": 1726784814.598,
        "CreationDate": 1602103465.273,
        "SchemaAttributes": [
            {
                "Name": "sub",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": false,
                "Required": true,
                "StringAttributeConstraints": {
                    "MinLength": "1",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "name",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
```

```
                "Name": "given_name",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "family_name",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "middle_name",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "nickname",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
```

```
            "Name": "preferred_username",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            }
        },
        {
            "Name": "profile",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            }
        },
        {
            "Name": "picture",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            }
        },
        {
            "Name": "website",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            }
        },
        {
```

```
                "Name": "email",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": true,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "email_verified",
                "AttributeDataType": "Boolean",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false
            },
            {
                "Name": "gender",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "birthdate",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "10",
                    "MaxLength": "10"
                }
            },
            {
                "Name": "zoneinfo",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
```

```
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "locale",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "phone_number",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "phone_number_verified",
                "AttributeDataType": "Boolean",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false
            },
            {
                "Name": "address",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
```

```
            }
        },
        {
            "Name": "updated_at",
            "AttributeDataType": "Number",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "NumberAttributeConstraints": {
                "MinValue": "0"
            }
        },
        {
            "Name": "identities",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {}
        },
        {
            "Name": "custom:111",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "1",
                "MaxLength": "256"
            }
        },
        {
            "Name": "dev:custom:222",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": true,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "1",
                "MaxLength": "421"
            }
        },
        {
            "Name": "custom:accesstoken",
```

```
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "custom:idtoken",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MaxLength": "2048"
                }
            }
        ],
        "AutoVerifiedAttributes": [
            "email"
        ],
        "SmsVerificationMessage": "Your verification code is {####}. ",
        "EmailVerificationMessage": "Your verification code is {####}. ",
        "EmailVerificationSubject": "Your verification code",
        "VerificationMessageTemplate": {
            "SmsMessage": "Your verification code is {####}. ",
            "EmailMessage": "Your verification code is {####}. ",
            "EmailSubject": "Your verification code",
            "EmailMessageByLink": "Please click the link below to verify your email
address. <b>{##Verify Your Email##}</b>\n this is from us-west-2_ywDJHlIfU",
            "EmailSubjectByLink": "Your verification link",
            "DefaultEmailOption": "CONFIRM_WITH_LINK"
        },
        "SmsAuthenticationMessage": "Your verification code is {####}. ",
        "UserAttributeUpdateSettings": {
            "AttributesRequireVerificationBeforeUpdate": []
        },
        "MfaConfiguration": "OPTIONAL",
        "DeviceConfiguration": {
            "ChallengeRequiredOnNewDevice": true,
            "DeviceOnlyRememberedOnUserPrompt": false
        },
        "EstimatedNumberOfUsers": 166,
```

```
        "EmailConfiguration": {
            "SourceArn": "arn:aws:ses:us-west-2:123456789012:identity/
admin@example.com",
            "EmailSendingAccount": "DEVELOPER"
        },
        "SmsConfiguration": {
            "SnsCallerArn": "arn:aws:iam::123456789012:role/service-role/userpool-
SMS-Role",
            "ExternalId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "SnsRegion": "us-west-2"
        },
        "UserPoolTags": {},
        "Domain": "myCustomDomain",
        "CustomDomain": "auth.example.com",
        "AdminCreateUserConfig": {
            "AllowAdminCreateUserOnly": false,
            "UnusedAccountValidityDays": 1,
            "InviteMessageTemplate": {
                "SMSMessage": "Your username is {username} and temporary password is
 {####}. ",
                "EmailMessage": "Your username is {username} and temporary password
 is {####}. ",
                "EmailSubject": "Your temporary password"
            }
        },
        "UserPoolAddOns": {
            "AdvancedSecurityMode": "ENFORCED",
            "AdvancedSecurityAdditionalFlows": {}
        },
        "Arn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_EXAMPLE",
        "AccountRecoverySetting": {
            "RecoveryMechanisms": [
                {
                    "Priority": 1,
                    "Name": "verified_email"
                }
            ]
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的Amazon Cognito 使用者集區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeUserPool。

forget-device

以下程式碼範例顯示如何使用 forget-device。

AWS CLI

忘記裝置

此範例會忘記裝置。

命令：

```
aws cognito-idp forget-device --device-key us-west-2_abcd_1234-5678
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ForgetDevice。

forgot-password

以下程式碼範例顯示如何使用 forgot-password。

AWS CLI

強制變更密碼

下列forgot-password範例會傳送訊息至 jane@example.com 以變更其密碼。

```
aws cognito-idp forgot-password --client-id 38fjsnc484p94kpqsnet7mpld0 --
username jane@example.com
```

輸出：

```
{
    "CodeDeliveryDetails": {
        "Destination": "j***@e***.com",
        "DeliveryMedium": "EMAIL",
        "AttributeName": "email"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ForgotPassword。

`get-csv-header`

以下程式碼範例顯示如何使用 `get-csv-header`。

AWS CLI

建立 csv 標頭

此範例會建立 csv 標頭。

如需匯入使用者的詳細資訊，請參閱從 CSV 檔案將使用者匯入使用者集區。

命令：

```
aws cognito-idp get-csv-header --user-pool-id us-west-2_aaaaaaaaa
```

輸出：

```
{
  "UserPoolId": "us-west-2_aaaaaaaaa",
  "CSVHeader": [
      "name",
      "given_name",
      "family_name",
      "middle_name",
      "nickname",
      "preferred_username",
      "profile",
      "picture",
      "website",
      "email",
      "email_verified",
      "gender",
      "birthdate",
      "zoneinfo",
      "locale",
      "phone_number",
      "phone_number_verified",
      "address",
      "updated_at",
```

```
      "cognito:mfa_enabled",
      "cognito:username"
  ]
}
```

... 從 CSV 檔案將使用者匯入使用者集區：https：//https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-pools-using-import-tool.html

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCsvHeader。

get-device

以下程式碼範例顯示如何使用 get-device。

AWS CLI

取得裝置

下列get-device範例顯示一個裝置供目前登入的使用者使用。

```
aws cognito-idp get-device \
    --access-token eyJra456defEXAMPLE \
    --device-key us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "Device": {
        "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "DeviceAttributes": [
            {
                "Name": "device_status",
                "Value": "valid"
            },
            {
                "Name": "device_name",
                "Value": "MyDevice"
            },
            {
                "Name": "dev:device_arn",
                "Value": "arn:aws:cognito-idp:us-west-2:123456789012:owner/diego.us-west-2_EXAMPLE/device/us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

```
            },
            {
                "Name": "dev:device_owner",
                "Value": "diego.us-west-2_EXAMPLE"
            },
            {
                "Name": "last_ip_used",
                "Value": "192.0.2.1"
            },
            {
                "Name": "dev:device_remembered_status",
                "Value": "remembered"
            },
            {
                "Name": "dev:device_sdk",
                "Value": "aws-sdk"
            }
        ],
        "DeviceCreateDate": 1715100742.022,
        "DeviceLastModifiedDate": 1723233651.167,
        "DeviceLastAuthenticatedDate": 1715100742.0
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的在使用者集區中使用使用者裝置。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDevice。

get-group

以下程式碼範例顯示如何使用 get-group。

AWS CLI

取得群組的相關資訊

下列get-group範例列出名為 之使用者群組的屬性MyGroup。此群組具有與其相關聯的優先順序和 IAM 角色。

```
aws cognito-idp get-group \
    --user-pool-id us-west-2_EXAMPLE \
    --group-name MyGroup
```

輸出：

```
{
    "Group": {
        "GroupName": "MyGroup",
        "UserPoolId": "us-west-2_EXAMPLE",
        "RoleArn": "arn:aws:iam::123456789012:role/example-cognito-role",
        "Precedence": 7,
        "LastModifiedDate": 1697211218.305,
        "CreationDate": 1611685503.954
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的將群組新增至使用者集區。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetGroup。

get-identity-provider-by-identifier

以下程式碼範例顯示如何使用 get-identity-provider-by-identifier。

AWS CLI

從 IdP 識別符取得身分提供者的組態

下列get-identity-provider-by-identifier範例會傳回識別碼為 的身分提供者組態mysso。

```
aws cognito-idp get-identity-provider-by-identifier \
    --user-pool-id us-west-2_EXAMPLE \
    --idp-identifier mysso
```

輸出：

```
{
    "IdentityProvider": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ProviderName": "MYSAML",
        "ProviderType": "SAML",
        "ProviderDetails": {
            "ActiveEncryptionCertificate": "[Certificate contents]",
```

```
            "IDPSignout": "false",
            "MetadataURL": "https://auth.example.com/saml/metadata/",
            "SLORedirectBindingURI": "https://auth.example.com/saml/logout/",
            "SSORedirectBindingURI": "https://auth.example.com/saml/assertion/"
        },
        "AttributeMapping": {
            "email": "email"
        },
        "IdpIdentifiers": [
            "mysso",
            "mysamlsso"
        ],
        "LastModifiedDate": 1705616729.188,
        "CreationDate": 1643734622.919
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的第三方 IdP 登入。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIdentityProviderByIdentifier。

get-log-delivery-configuration

以下程式碼範例顯示如何使用 get-log-delivery-configuration。

AWS CLI

顯示日誌交付組態

下列get-log-delivery-configuration範例顯示請求的使用者集區的日誌匯出設定。

```
aws cognito-idp get-log-delivery-configuration \
    --user-pool-id us-west-2_EXAMPLE
```

輸出：

```
{
    "LogDeliveryConfiguration": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "LogConfigurations": [
            {
                "LogLevel": "INFO",
```

```
                "EventSource": "userAuthEvents",
                "FirehoseConfiguration": {
                    "StreamArn": "arn:aws:firehose:us-
west-2:123456789012:deliverystream/my-test-deliverystream"
                }
            },
            {
                "LogLevel": "ERROR",
                "EventSource": "userNotification",
                "CloudWatchLogsConfiguration": {
                    "LogGroupArn": "arn:aws:logs:us-west-2:123456789012:log-
group:my-message-delivery-logs"
                }
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的匯出使用者集區日誌。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLogDeliveryConfiguration。

get-signing-certificate

以下程式碼範例顯示如何使用 get-signing-certificate。

AWS CLI

顯示 SAML 簽署憑證

下列get-signing-certificate範例顯示請求使用者集區的 SAML 2.0 簽署憑證。

```
aws cognito-idp get-signing-certificate \
    --user-pool-id us-west-2_EXAMPLE
```

輸出：

```
{
    "Certificate": "[Certificate content]"
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的 SAML 簽署和加密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSigningCertificate。

get-ui-customization

以下程式碼範例顯示如何使用 get-ui-customization。

AWS CLI

顯示應用程式用戶端的傳統託管 UI 自訂設定

下列get-ui-customization範例顯示不從使用者集區繼承設定之應用程式用戶端的傳統託管 UI 自訂設定。

```
aws cognito-idp get-ui-customization \
    --user-pool-id us-west-2_EXAMPLE \
    --client-id 1example23456789
```

輸出：

```
{
    "UICustomization": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ClientId": "1example23456789",
        "ImageUrl": "https://example.cloudfront.net/us-
west-2_EXAMPLE/1example23456789/20250115191928/assets/images/image.jpg",
        "CSS": "\n.logo-customizable {\n  max-width: 80%;\n  max-height: 30%;
\n}\n\n.banner-customizable {\n  padding: 25px 0px 25px 0px;\n  background-color:
 lightgray;\n}\n\n.label-customizable {\n  font-weight: 400;\n}\n\n.textDescription-
customizable {\n  padding-top: 100px;\n  padding-bottom: 10px;\n  display: block;
\n  font-size: 12px;\n}\n\n.idpDescription-customizable {\n  padding-top: 10px;\n
  padding-bottom: 10px;\n  display: block;\n  font-size: 16px;\n}\n\n.legalText-
customizable {\n  color: #747474;\n  font-size: 11px;\n}\n\n.submitButton-
customizable {\n  font-size: 14px;\n  font-weight: bold;\n  margin: 20px 0px
 10px 0px;\n  height: 50px;\n  width: 100%;\n  color: #fff;\n  background-color:
 #337ab7;\n}\n\n.submitButton-customizable:hover {\n  color: #fff;\n  background-
color: #286090;\n}\n\n.errorMessage-customizable {\n  padding: 5px;\n  font-size:
 12px;\n  width: 100%;\n  background: #F5F5F5;\n  border: 2px solid #D64958;\n
  color: #D64958;\n}\n\n.inputField-customizable {\n  width: 100%;\n  height:
 34px;\n  color: #555;\n  background-color: #fff;\n  border: 1px solid #ccc;\n}\n
\n.inputField-customizable:focus {\n  border-color: #66afe9;\n  outline: 0;\n}\n
\n.idpButton-customizable {\n  height: 40px;\n  width: 100%;\n  width: 100%;\n
 text-align: center;\n  margin-bottom: 15px;\n  color: #fff;\n  background-color:
```

```
 #5bc0de;\n  border-color: #46b8da;\n}\n\n.idpButton-customizable:hover {\n
 color: #fff;\n  background-color: #31b0d5;\n}\n\n.socialButton-customizable {\n
  border-radius: 2px;\n  height: 60px;\n  margin-bottom: 15px;\n  padding: 1px;
\n  text-align: left;\n  width: 100%;\n}\n\n.redirect-customizable {\n  text-
align: center;\n}\n\n.passwordCheck-notValid-customizable {\n  color: #DF3312;
\n}\n\n.passwordCheck-valid-customizable {\n  color: #19BF00;\n}\n\n.background-
customizable {\n  background-color: #fff;\n}\n",
        "CSSVersion": "20250115191928"
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的託管 UI（傳統）品牌。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetUiCustomization。

get-user-attribute-verification-code

以下程式碼範例顯示如何使用 get-user-attribute-verification-code。

AWS CLI

傳送屬性驗證碼給目前的使用者

下列get-user-attribute-verification-code範例會將屬性驗證碼傳送至目前登入使用者的電子郵件地址。

```
aws cognito-idp get-user-attribute-verification-code \
    --access-token eyJra456defEXAMPLE \
    --attribute-name email
```

輸出：

```
{
    "CodeDeliveryDetails": {
        "Destination": "a***@e***",
        "DeliveryMedium": "EMAIL",
        "AttributeName": "email"
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的註冊及確認使用者帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetUserAttributeVerificationCode。

get-user-auth-factors

以下程式碼範例顯示如何使用 get-user-auth-factors。

AWS CLI

列出目前使用者可用的身分驗證因素

下列get-user-auth-factors範例列出目前登入使用者可用的身分驗證因素。

```
aws cognito-idp get-user-auth-factors \
    --access-token eyJra456defEXAMPLE
```

輸出：

```
{
    "Username": "testuser",
    "ConfiguredUserAuthFactors": [
        "PASSWORD",
        "EMAIL_OTP",
        "SMS_OTP",
        "WEB_AUTHN"
    ]
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的身分驗證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetUserAuthFactors。

get-user-pool-mfa-config

以下程式碼範例顯示如何使用 get-user-pool-mfa-config。

AWS CLI

顯示使用者集區的多重驗證和 WebAuthn 設定

下列get-user-pool-mfa-config範例顯示請求的使用者集區的 MFA 和 WebAuthn 組態。

```
aws cognito-idp get-user-pool-mfa-config \
    --user-pool-id us-west-2_EXAMPLE
```

輸出：

```
{
    "SmsMfaConfiguration": {
        "SmsAuthenticationMessage": "Your OTP for MFA or sign-in: use {####}.",
        "SmsConfiguration": {
            "SnsCallerArn": "arn:aws:iam::123456789012:role/service-role/my-SMS-
Role",
            "ExternalId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "SnsRegion": "us-west-2"
        }
    },
    "SoftwareTokenMfaConfiguration": {
        "Enabled": true
    },
    "EmailMfaConfiguration": {
        "Message": "Your OTP for MFA or sign-in: use {####}",
        "Subject": "OTP test"
    },
    "MfaConfiguration": "OPTIONAL",
    "WebAuthnConfiguration": {
        "RelyingPartyId": "auth.example.com",
        "UserVerification": "preferred"
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的新增 MFA。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetUserPoolMfaConfig。

get-user

以下程式碼範例顯示如何使用 get-user。

AWS CLI

取得目前使用者的詳細資訊

下列get-user範例顯示目前登入使用者的設定檔。

```
aws cognito-idp get-user \
    --access-token eyJra456defEXAMPLE
```

輸出：

```
{
    "Username": "johndoe",
    "UserAttributes": [
        {
            "Name": "sub",
            "Value": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        },
        {
            "Name": "identities",
            "Value": "[{\"userId\":\"a1b2c3d4-5678-90ab-cdef-EXAMPLE22222\",
\"providerName\":\"SignInWithApple\",\"providerType\":\"SignInWithApple\",\"issuer
\":null,\"primary\":false,\"dateCreated\":1701125599632}]"
        },
        {
            "Name": "email_verified",
            "Value": "true"
        },
        {
            "Name": "custom:state",
            "Value": "Maine"
        },
        {
            "Name": "name",
            "Value": "John Doe"
        },
        {
            "Name": "phone_number_verified",
            "Value": "true"
        },
        {
            "Name": "phone_number",
            "Value": "+12065551212"
        },
        {
            "Name": "preferred_username",
            "Value": "jamesdoe"
        },
        {
            "Name": "locale",
            "Value": "EMEA"
        },
        {
```

```
            "Name": "email",
            "Value": "jamesdoe@example.com"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的管理使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetUser。

global-sign-out

以下程式碼範例顯示如何使用 global-sign-out。

AWS CLI

登出目前的使用者

下列global-sign-out範例會登出目前的使用者。

```
aws cognito-idp global-sign-out \
    --access-token eyJra456defEXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的管理使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GlobalSignOut。

initiate-auth

以下程式碼範例顯示如何使用 initiate-auth。

AWS CLI

登入使用者

下列initiate-auth範例使用基本的使用者名稱密碼流程登入使用者，沒有額外的挑戰。

```
aws cognito-idp initiate-auth \
    --auth-flow USER_PASSWORD_AUTH \
    --client-id 1example23456789 \
    --analytics-metadata AnalyticsEndpointId=d70b2ba36a8c4dc5a04a0451aEXAMPLE \
```

```
    --auth-parameters USERNAME=testuser,PASSWORD=[Password] --user-context-
data EncodedData=mycontextdata --client-metadata MyTestKey=MyTestValue
```

輸出：

```
{
    "AuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "eyJra123abcEXAMPLE",
        "IdToken": "eyJra789ghiEXAMPLE",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-v7w9UcY6"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的身分驗證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 InitiateAuth。

list-devices

以下程式碼範例顯示如何使用 list-devices。

AWS CLI

列出使用者的裝置

下列list-devices範例列出目前使用者已註冊的裝置。

```
aws cognito-idp list-devices \
    --access-token eyJra456defEXAMPLE
```

輸出：

```
{
    "Devices": [
        {
            "DeviceAttributes": [
                {
```

```
                    "Name": "device_status",
                    "Value": "valid"
                },
                {
                    "Name": "device_name",
                    "Value": "Dart-device"
                },
                {
                    "Name": "last_ip_used",
                    "Value": "192.0.2.1"
                }
            ],
            "DeviceCreateDate": 1715100742.022,
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceLastAuthenticatedDate": 1715100742.0,
            "DeviceLastModifiedDate": 1723233651.167
        },
        {
            "DeviceAttributes": [
                {
                    "Name": "device_status",
                    "Value": "valid"
                },
                {
                    "Name": "last_ip_used",
                    "Value": "192.0.2.2"
                }
            ],
            "DeviceCreateDate": 1726856147.993,
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "DeviceLastAuthenticatedDate": 1726856147.0,
            "DeviceLastModifiedDate": 1726856147.993
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的使用裝置。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDevices。

`list-groups`

以下程式碼範例顯示如何使用 `list-groups`。

AWS CLI

列出使用者集區中的群組

下列list-groups範例列出所請求使用者集區中的前兩個群組。

```
aws cognito-idp list-groups \
    --user-pool-id us-west-2_EXAMPLE \
    --max-items 2
```

輸出：

```
{
    "Groups": [
        {
            "CreationDate": 1681760899.633,
            "Description": "My test group",
            "GroupName": "testgroup",
            "LastModifiedDate": 1681760899.633,
            "Precedence": 1,
            "UserPoolId": "us-west-2_EXAMPLE"
        },
        {
            "CreationDate": 1642632749.051,
            "Description": "Autogenerated group for users who sign in using
 Facebook",
            "GroupName": "us-west-2_EXAMPLE_Facebook",
            "LastModifiedDate": 1642632749.051,
            "UserPoolId": "us-west-2_EXAMPLE"
        }
    ],
    "NextToken": "[Pagination token]"
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的將群組新增至使用者集區。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListGroups。

list-identity-providers

以下程式碼範例顯示如何使用 list-identity-providers。

AWS CLI

列出身分提供者

下列list-identity-providers範例列出所請求使用者集區中的前兩個身分提供者。

```
aws cognito-idp list-identity-providers \
    --user-pool-id us-west-2_EXAMPLE \
    --max-items 2
```

輸出：

```
{
    "Providers": [
        {
            "CreationDate": 1619477386.504,
            "LastModifiedDate": 1703798328.142,
            "ProviderName": "Azure",
            "ProviderType": "SAML"
        },
        {
            "CreationDate": 1642698776.175,
            "LastModifiedDate": 1642699086.453,
            "ProviderName": "LoginWithAmazon",
            "ProviderType": "LoginWithAmazon"
        }
    ],
    "NextToken": "[Pagination token]"
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的第三方 IdP 登入。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListIdentityProviders。

list-resource-servers

以下程式碼範例顯示如何使用 list-resource-servers。

AWS CLI

列出資源伺服器

下列list-resource-servers範例列出所請求使用者集區中的前兩個資源伺服器。

```
aws cognito-idp list-resource-servers \
    --user-pool-id us-west-2_EXAMPLE \
    --max-results 2
```

輸出：

```
{
    "ResourceServers": [
        {
            "Identifier": "myapi.example.com",
            "Name": "Example API with custom access control scopes",
            "Scopes": [
                {
                    "ScopeDescription": "International customers",
                    "ScopeName": "international.read"
                },
                {
                    "ScopeDescription": "Domestic customers",
                    "ScopeName": "domestic.read"
                }
            ],
            "UserPoolId": "us-west-2_EXAMPLE"
        },
        {
            "Identifier": "myapi2.example.com",
            "Name": "Another example API for access control",
            "Scopes": [
                {
                    "ScopeDescription": "B2B customers",
                    "ScopeName": "b2b.read"
                }
            ],
            "UserPoolId": "us-west-2_EXAMPLE"
        }
    ],
    "NextToken": "[Pagination token]"
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的使用資源伺服器進行存取控制。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourceServers。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出使用者集區標籤

下列list-tags-for-resource範例會列出指派給具有請求 ARN 之使用者集區的標籤。

```
aws cognito-idp list-tags-for-resource \
    --resource-arn arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-west-2_EXAMPLE
```

輸出：

```
{
    "Tags": {
        "administrator": "Jie",
        "tenant": "ExampleCorp"
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的標記 Amazon Cognito 資源。

Amazon Cognito

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

list-user-import-jobs

以下程式碼範例顯示如何使用 list-user-import-jobs。

AWS CLI

列出使用者匯入任務和狀態

下列list-user-import-jobs範例列出請求的使用者集區中的前三個使用者匯入任務及其詳細資訊。

```
aws cognito-idp list-user-import-jobs \
    --user-pool-id us-west-2_EXAMPLE \
    --max-results 3
```

輸出：


```
{
    "PaginationToken": "us-west-2_EXAMPLE#import-example3#1667948397084",
    "UserImportJobs": [
        {
            "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/service-role/
Cognito-UserImport-Role",
            "CompletionDate": 1735329786.142,
            "CompletionMessage": "The user import job has expired.",
            "CreationDate": 1735241621.022,
            "FailedUsers": 0,
            "ImportedUsers": 0,
            "JobId": "import-example1",
            "JobName": "Test-import-job-1",
            "PreSignedUrl": "https://aws-cognito-idp-user-import-pdx.s3.us-
west-2.amazonaws.com/123456789012/us-west-2_EXAMPLE/import-mAgUtd8PMm?
X-Amz-Security-Token=[token]&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20241226T193341Z&X-Amz-SignedHeaders=host%3Bx-amz-server-side-encryption&X-Amz-
Expires=899&X-Amz-Credential=[credential]&X-Amz-Signature=[signature]",
            "SkippedUsers": 0,
            "Status": "Expired",
            "UserPoolId": "us-west-2_EXAMPLE"
        },
        {
            "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/service-role/
Cognito-UserImport-Role",
            "CompletionDate": 1681509058.408,
            "CompletionMessage": "Too many users have failed or been skipped during
 the import.",
            "CreationDate": 1681509001.477,
            "FailedUsers": 1,
            "ImportedUsers": 0,
            "JobId": "import-example2",
            "JobName": "Test-import-job-2",
            "PreSignedUrl": "https://aws-cognito-idp-user-import-pdx.s3.us-
west-2.amazonaws.com/123456789012/us-west-2_EXAMPLE/import-mAgUtd8PMm?
X-Amz-Security-Token=[token]&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20241226T193341Z&X-Amz-SignedHeaders=host%3Bx-amz-server-side-encryption&X-Amz-
Expires=899&X-Amz-Credential=[credential]&X-Amz-Signature=[signature]",
            "SkippedUsers": 0,
            "StartDate": 1681509057.965,
            "Status": "Failed",
            "UserPoolId": "us-west-2_EXAMPLE"
```

```
        },
        {
            "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/service-role/
Cognito-UserImport-Role",
            "CompletionDate": 1.667864578676E9,
            "CompletionMessage": "Import Job Completed Successfully.",
            "CreationDate": 1.667864480281E9,
            "FailedUsers": 0,
            "ImportedUsers": 6,
            "JobId": "import-example3",
            "JobName": "Test-import-job-3",
            "PreSignedUrl": "https://aws-cognito-idp-user-import-pdx.s3.us-
west-2.amazonaws.com/123456789012/us-west-2_EXAMPLE/import-mAgUtd8PMm?
X-Amz-Security-Token=[token]&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20241226T193341Z&X-Amz-SignedHeaders=host%3Bx-amz-server-side-encryption&X-Amz-
Expires=899&X-Amz-Credential=[credential]&X-Amz-Signature=[signature]",
            "SkippedUsers": 0,
            "StartDate": 1.667864578167E9,
            "Status": "Succeeded",
            "UserPoolId": "us-west-2_EXAMPLE"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的從 CSV 檔案匯入使用者。 Amazon Cognito

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListUserImportJobs。

list-user-pool-clients

以下程式碼範例顯示如何使用 list-user-pool-clients。

AWS CLI

列出應用程式用戶端

下列list-user-pool-clients範例列出所請求使用者集區中的前三個應用程式用戶端。

```
aws cognito-idp list-user-pool-clients \
    --user-pool-id us-west-2_EXAMPLE \
    --max-results 3
```

輸出：

```
{
    "NextToken": "[Pagination token]",
    "UserPoolClients": [
        {
            "ClientId": "1example23456789",
            "ClientName": "app-client-1",
            "UserPoolId": "us-west-2_EXAMPLE"
        },
        {
            "ClientId": "2example34567890",
            "ClientName": "app-client-2",
            "UserPoolId": "us-west-2_EXAMPLE"
        },
        {
            "ClientId": "3example45678901",
            "ClientName": "app-client-3",
            "UserPoolId": "us-west-2_EXAMPLE"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的應用程式用戶端。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListUserPoolClients。

`list-user-pools`

以下程式碼範例顯示如何使用 `list-user-pools`。

AWS CLI

若要列出使用者集區

下列`list-user-pools`範例列出目前 CLI 登入 AWS 資料帳戶中的 3 個可用使用者集區。

```
aws cognito-idp list-user-pools \
    --max-results 3
```

輸出：

```
{
```

```
    "NextToken": "[Pagination token]",
    "UserPools": [
        {
            "CreationDate": 1681502497.741,
            "Id": "us-west-2_EXAMPLE1",
            "LambdaConfig": {
                "CustomMessage": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                "PreSignUp": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                "PreTokenGeneration": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                "PreTokenGenerationConfig": {
                    "LambdaArn": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                    "LambdaVersion": "V1_0"
                }
            },
            "LastModifiedDate": 1681502497.741,
            "Name": "user pool 1"
        },
        {
            "CreationDate": 1686064178.717,
            "Id": "us-west-2_EXAMPLE2",
            "LambdaConfig": {
            },
            "LastModifiedDate": 1686064178.873,
            "Name": "user pool 2"
        },
        {
            "CreationDate": 1627681712.237,
            "Id": "us-west-2_EXAMPLE3",
            "LambdaConfig": {
                "UserMigration": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction"
            },
            "LastModifiedDate": 1678486942.479,
            "Name": "user pool 3"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的Amazon Cognito 使用者集區。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListUserPools。

`list-users-in-group`

以下程式碼範例顯示如何使用 `list-users-in-group`。

AWS CLI

列出群組中的使用者

此範例列出群組 MyGroup 中的使用者。

命令：

```
aws cognito-idp list-users-in-group --user-pool-id us-west-2_aaaaaaaaa --group-
name MyGroup
```

輸出：

```
{
  "Users": [
      {
          "Username": "acf10624-80bb-401a-ac61-607bee2110ec",
          "Attributes": [
              {
                  "Name": "sub",
                  "Value": "acf10624-80bb-401a-ac61-607bee2110ec"
              },
              {
                  "Name": "custom:CustomAttr1",
                  "Value": "New Value!"
              },
              {
                  "Name": "email",
                  "Value": "jane@example.com"
              }
          ],
          "UserCreateDate": 1548102770.284,
          "UserLastModifiedDate": 1548103204.893,
          "Enabled": true,
          "UserStatus": "CONFIRMED"
      },
      {
```

```
            "Username": "22704aa3-fc10-479a-97eb-2af5806bd327",
            "Attributes": [
                {
                    "Name": "sub",
                    "Value": "22704aa3-fc10-479a-97eb-2af5806bd327"
                },
                {
                    "Name": "email_verified",
                    "Value": "true"
                },
                {
                    "Name": "email",
                    "Value": "diego@example.com"
                }
            ],
            "UserCreateDate": 1548089817.683,
            "UserLastModifiedDate": 1548089817.683,
            "Enabled": true,
            "UserStatus": "FORCE_CHANGE_PASSWORD"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListUsersInGroup。

list-users

以下程式碼範例顯示如何使用 list-users。

AWS CLI

範例 1：列出具有伺服器端篩選條件的使用者

下列list-users範例列出請求的使用者集區中的 3 個使用者，其電子郵件地址開頭為 testuser。

```
aws cognito-idp list-users \
    --user-pool-id us-west-2_EXAMPLE \
    --filter email^=\"testuser\" \
    --max-items 3
```

輸出：

```
{
    "PaginationToken": "efgh5678EXAMPLE",
    "Users": [
        {
            "Attributes": [
                {
                    "Name": "sub",
                    "Value": "eaad0219-2117-439f-8d46-4db20e59268f"
                },
                {
                    "Name": "email",
                    "Value": "testuser@example.com"
                }
            ],
            "Enabled": true,
            "UserCreateDate": 1682955829.578,
            "UserLastModifiedDate": 1689030181.63,
            "UserStatus": "CONFIRMED",
            "Username": "testuser"
        },
        {
            "Attributes": [
                {
                    "Name": "sub",
                    "Value": "3b994cfd-0b07-4581-be46-3c82f9a70c90"
                },
                {
                    "Name": "email",
                    "Value": "testuser2@example.com"
                }
            ],
            "Enabled": true,
            "UserCreateDate": 1684427979.201,
            "UserLastModifiedDate": 1684427979.201,
            "UserStatus": "UNCONFIRMED",
            "Username": "testuser2"
        },
        {
            "Attributes": [
                {
                    "Name": "sub",
                    "Value": "5929e0d1-4c34-42d1-9b79-a5ecacfe66f7"
                },
```

```
                {
                    "Name": "email",
                    "Value": "testuser3@example.com"
                }
            ],
            "Enabled": true,
            "UserCreateDate": 1684427823.641,
            "UserLastModifiedDate": 1684427823.641,
            "UserStatus": "UNCONFIRMED",
            "Username": "testuser3@example.com"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的管理和搜尋使用者。

範例 2：列出具有用戶端篩選條件的使用者

下列list-users範例列出三個具有 屬性之使用者的屬性，在此例中是其電子郵件地址，其中包含電子郵件網域 "@example.com"。如果其他屬性包含此字串，則也會顯示這些屬性。第二個使用者沒有符合查詢的屬性，且會從顯示的輸出中排除，但不會從伺服器回應中排除。

```
aws cognito-idp list-users \
    --user-pool-id us-west-2_EXAMPLE \
    --max-items 3
    --query Users\[\*\].Attributes\[\?Value\.contains\(\@\,\'@example.com\'\)\]
```

輸出：

```
[
    [
        {
            "Name": "email",
            "Value": "admin@example.com"
        }
    ],
    [],
    [
        {
            "Name": "email",
            "Value": "operator@example.com"
        }
    ]
```

```
]
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的管理和搜尋使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListUsers。

`list-web-authn-credentials`

以下程式碼範例顯示如何使用 `list-web-authn-credentials`。

AWS CLI

列出通行金鑰登入資料

下列`list-web-authn-credentials`範例列出目前使用者的 passkey 或 WebAuthn 登入資料。他們有一個已註冊的裝置。

```
aws cognito-idp list-web-authn-credentials \
    --access-token eyJra456defEXAMPLE
```

輸出：

```
{
    "Credentials": [
        {
            "AuthenticatorAttachment": "cross-platform",
            "CreatedAt": 1736293876.115,
            "CredentialId": "8LApgk4-lNUFHbhm2w6Und7-
uxcc8coJGsPxiogvHoItc64xWQc3r4CEXAMPLE",
            "FriendlyCredentialName": "Roaming passkey",
            "RelyingPartyId": "auth.example.com"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的 Passkey 登入。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListWebAuthnCredentials。

`resend-confirmation-code`

以下程式碼範例顯示如何使用 `resend-confirmation-code`。

AWS CLI

若要重新傳送確認碼

下列 resend-confirmation-code 範例會傳送確認碼給使用者 jane。

```
aws cognito-idp resend-confirmation-code \
    --client-id 12a3b456c7de890f11g123hijk \
    --username jane
```

輸出：

```
{
    "CodeDeliveryDetails": {
        "Destination": "j***@e***.com",
        "DeliveryMedium": "EMAIL",
        "AttributeName": "email"
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的註冊及確認使用者帳戶。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ResendConfirmationCode。

respond-to-auth-challenge

以下程式碼範例顯示如何使用 respond-to-auth-challenge。

AWS CLI

範例 1：回應 NEW_PASSWORD_REQUIRED 挑戰

下列respond-to-auth-challenge範例會回應 NEW_PASSWORD_REQUIRED 傳回的 initiate-auth 挑戰。它會為使用者 設定密碼jane@example.com。

```
aws cognito-idp respond-to-auth-challenge \
    --client-id 1example23456789 \
    --challenge-name NEW_PASSWORD_REQUIRED \
    --challenge-responses USERNAME=jane@example.com,NEW_PASSWORD=[Password] \
    --session AYABeEv5HklEXAMPLE
```

輸出：

```
{
    "ChallengeParameters": {},
    "AuthenticationResult": {
        "AccessToken": "ACCESS_TOKEN",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "REFRESH_TOKEN",
        "IdToken": "ID_TOKEN",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-wt2ha1Zd"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的身分驗證。

範例 2：回應 SELECT_MFA_TYPE 挑戰

下列respond-to-auth-challenge範例選擇 TOTP MFA 做為目前使用者的 MFA 選項。系統會提示使用者選取 MFA 類型，然後提示使用者輸入其 MFA 代碼。

```
aws cognito-idp respond-to-auth-challenge \
    --client-id 1example23456789
    --session AYABeEv5HklEXAMPLE
    --challenge-name SELECT_MFA_TYPE
    --challenge-responses USERNAME=testuser,ANSWER=SOFTWARE_TOKEN_MFA
```

輸出：

```
{
    "ChallengeName": "SOFTWARE_TOKEN_MFA",
    "Session": "AYABeEv5HklEXAMPLE",
    "ChallengeParameters": {
        "FRIENDLY_DEVICE_NAME": "transparent"
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的新增 MFA。

範例 3：回應 Software_TOKEN_MFA 挑戰

下列respond-to-auth-challenge範例提供 TOTP MFA 程式碼並完成登入。

```
aws cognito-idp respond-to-auth-challenge \
    --client-id 1example23456789 \
    --session AYABeEv5HklEXAMPLE \
    --challenge-name SOFTWARE_TOKEN_MFA \
    --challenge-responses USERNAME=testuser,SOFTWARE_TOKEN_MFA_CODE=123456
```

輸出：

```
{
    "AuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "eyJra123abcEXAMPLE",
        "IdToken": "eyJra789ghiEXAMPLE",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-v7w9UcY6"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的新增 MFA。

- 如需 API 詳細資訊，請參閱《AWS CLI API 參考》中的 RespondToAuthChallenge。

revoke-token

以下程式碼範例顯示如何使用 revoke-token。

AWS CLI

撤銷重新整理字符

以下內容會revoke-token撤銷請求的重新整理字符和相關聯的存取字符。

```
aws cognito-idp revoke-token \
    --token eyJjd123abcEXAMPLE \
    --client-id 1example23456789
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的撤銷權杖。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RevokeToken。

set-log-delivery-configuration

以下程式碼範例顯示如何使用 set-log-delivery-configuration。

AWS CLI

從使用者集區設定日誌匯出

下列set-log-delivery-configuration範例會設定請求的使用者集區，將使用者通知錯誤記錄到日誌群組，並將使用者身分驗證資訊記錄到 S3 儲存貯體。

```
aws cognito-idp set-log-delivery-configuration \
    --user-pool-id us-west-2_EXAMPLE \
    --log-
configurations LogLevel=ERROR,EventSource=userNotification,CloudWatchLogsConfiguration={LogG
west-2:123456789012:log-group:cognito-
exported} LogLevel=INFO,EventSource=userAuthEvents,S3Configuration={BucketArn=arn:aws:s3:::a
s3-demo-bucket1}
```

輸出：

```
{
   "LogDeliveryConfiguration": {
        "LogConfigurations": [
            {
                "CloudWatchLogsConfiguration": {
                    "LogGroupArn": "arn:aws:logs:us-west-2:123456789012:log-
group:cognito-exported"
                },
                "EventSource": "userNotification",
                "LogLevel": "ERROR"
            },
            {
                "EventSource": "userAuthEvents",
                "LogLevel": "INFO",
                "S3Configuration": {
                    "BucketArn": "arn:aws:s3:::amzn-s3-demo-bucket1"
```

```
                }
            }
        ],
        "UserPoolId": "us-west-2_EXAMPLE"
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的匯出使用者集區日誌。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetLogDeliveryConfiguration。

set-risk-configuration

以下程式碼範例顯示如何使用 set-risk-configuration。

AWS CLI

設定威脅防護風險組態

下列set-risk-configuration範例會在請求的應用程式用戶端中設定威脅防護訊息和動作、遭入侵的登入資料，以及 IP 地址例外狀況。由於 NotifyConfiguration 物件的複雜性，JSON 輸入是此命令的最佳實務。

```
aws cognito-idp set-risk-configuration \
    --cli-input-json file://set-risk-configuration.json
```

set-risk-configuration.json 的內容：

```
{
    "AccountTakeoverRiskConfiguration": {
        "Actions": {
            "HighAction": {
                "EventAction": "MFA_REQUIRED",
                "Notify": true
            },
            "LowAction": {
                "EventAction": "NO_ACTION",
                "Notify": true
            },
            "MediumAction": {
                "EventAction": "MFA_IF_CONFIGURED",
                "Notify": true
            }
```

```
        },
        "NotifyConfiguration": {
            "BlockEmail": {
                "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
 email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
 blocked an unrecognized sign-in to your account with this information:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
                "Subject": "Blocked sign-in attempt",
                "TextBody": "We blocked an unrecognized sign-in to your account with
 this information:\nTime: {login-time}\nDevice: {device-name}\nLocation: {city},
 {country}\nIf this sign-in was not by you, you should change your password and
 notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you, you
 can follow {one-click-link-valid} to let us know"
            },
            "From": "admin@example.com",
            "MfaEmail": {
                "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML email
 context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We required
 you to use multi-factor authentication for the following sign-in attempt:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
                "Subject": "New sign-in attempt",
                "TextBody": "We required you to use multi-factor authentication
 for the following sign-in attempt:\nTime: {login-time}\nDevice: {device-
name}\nLocation: {city}, {country}\nIf this sign-in was not by you, you should
 change your password and notify us by clicking on {one-click-link-invalid}\nIf this
 sign-in was by you, you can follow {one-click-link-valid} to let us know"
            },
            "NoActionEmail": {
                "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
 email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
 observed an unrecognized sign-in to your account with this information:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
```

```
                "Subject": "New sign-in attempt",
                "TextBody": "We observed an unrecognized sign-in to your account
 with this information:\nTime: {login-time}\nDevice: {device-name}\nLocation:
 {city}, {country}\nIf this sign-in was not by you, you should change your password
 and notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you,
 you can follow {one-click-link-valid} to let us know"
            },
            "ReplyTo": "admin@example.com",
            "SourceArn": "arn:aws:ses:us-west-2:123456789012:identity/
admin@example.com"
        }
    },
    "ClientId": "1example23456789",
    "CompromisedCredentialsRiskConfiguration": {
        "Actions": {
            "EventAction": "BLOCK"
        },
        "EventFilter": [
            "PASSWORD_CHANGE",
            "SIGN_UP",
            "SIGN_IN"
        ]
    },
    "RiskExceptionConfiguration": {
        "BlockedIPRangeList": [
            "192.0.2.1/32",
            "192.0.2.2/32"
        ],
        "SkippedIPRangeList": [
            "203.0.113.1/32",
            "203.0.113.2/32"
        ]
    },
    "UserPoolId": "us-west-2_EXAMPLE"
}
```

輸出：

```
{
    "RiskConfiguration": {
        "AccountTakeoverRiskConfiguration": {
            "Actions": {
                "HighAction": {
```

```
                    "EventAction": "MFA_REQUIRED",
                    "Notify": true
                },
                "LowAction": {
                    "EventAction": "NO_ACTION",
                    "Notify": true
                },
                "MediumAction": {
                    "EventAction": "MFA_IF_CONFIGURED",
                    "Notify": true
                }
            },
            "NotifyConfiguration": {
                "BlockEmail": {
                    "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
 email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
 blocked an unrecognized sign-in to your account with this information:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
                    "Subject": "Blocked sign-in attempt",
                    "TextBody": "We blocked an unrecognized sign-in to your account
 with this information:\nTime: {login-time}\nDevice: {device-name}\nLocation:
 {city}, {country}\nIf this sign-in was not by you, you should change your password
 and notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you,
 you can follow {one-click-link-valid} to let us know"
                },
                "From": "admin@example.com",
                "MfaEmail": {
                    "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
 email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
 required you to use multi-factor authentication for the following sign-in attempt:
\n<ul>\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location:
 {city}, {country}</li>\n</ul>\nIf this sign-in was not by you, you should change
 your password and notify us by clicking on <a href={one-click-link-invalid}>this
 link</a>\nIf this sign-in was by you, you can follow <a href={one-click-link-
valid}>this link</a> to let us know</pre>\n</body>\n</html>",
                    "Subject": "New sign-in attempt",
                    "TextBody": "We required you to use multi-factor authentication
 for the following sign-in attempt:\nTime: {login-time}\nDevice: {device-
name}\nLocation: {city}, {country}\nIf this sign-in was not by you, you should
```

```
 change your password and notify us by clicking on {one-click-link-invalid}\nIf this
 sign-in was by you, you can follow {one-click-link-valid} to let us know"
                },
                "NoActionEmail": {
                    "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
 email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
 observed an unrecognized sign-in to your account with this information:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
                    "Subject": "New sign-in attempt",
                    "TextBody": "We observed an unrecognized sign-in to your account
 with this information:\nTime: {login-time}\nDevice: {device-name}\nLocation:
 {city}, {country}\nIf this sign-in was not by you, you should change your password
 and notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you,
 you can follow {one-click-link-valid} to let us know"
                },
                "ReplyTo": "admin@example.com",
                "SourceArn": "arn:aws:ses:us-west-2:123456789012:identity/
admin@example.com"
            }
        },
        "ClientId": "1example23456789",
        "CompromisedCredentialsRiskConfiguration": {
            "Actions": {
                "EventAction": "BLOCK"
            },
            "EventFilter": [
                "PASSWORD_CHANGE",
                "SIGN_UP",
                "SIGN_IN"
            ]
        },
        "RiskExceptionConfiguration": {
            "BlockedIPRangeList": [
                "192.0.2.1/32",
                "192.0.2.2/32"
            ],
            "SkippedIPRangeList": [
                "203.0.113.1/32",
                "203.0.113.2/32"
            ]
```

```
        },
        "UserPoolId": "us-west-2_EXAMPLE"
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的威脅防護。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetRiskConfiguration。

`set-ui-customization`

以下程式碼範例顯示如何使用 `set-ui-customization`。

AWS CLI

範例 1：自訂應用程式用戶端的傳統託管 UI

下列`set-ui-customization`範例會使用一些自訂 CSS 設定請求的應用程式用戶端，並以 Amazon Cognito 標誌做為應用程式標誌。

```
aws cognito-idp set-ui-customization \
    --user-pool-id us-west-2_ywDJHlIfU \
    --client-id 14pq32c5q2uq2q7keorloqvb23 \
    --css ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;\n}\n.banner-
customizable {\n\tpadding: 25px 0px 25px 0px;\n\tbackground-color: lightgray;
\n}\n.label-customizable {\n\tfont-weight: 400;\n}\n.textDescription-customizable
 {\n\tpadding-top: 10px;\n\tpadding-bottom: 10px;\n\tdisplay: block;\n\tfont-
size: 16px;\n}\n.idpDescription-customizable {\n\tpadding-top: 10px;\n\tpadding-
bottom: 10px;\n\tdisplay: block;\n\tfont-size: 16px;\n}\n.legalText-customizable
 {\n\tcolor: #747474;\n\tfont-size: 11px;\n}\n.submitButton-customizable {\n\tfont-
size: 11px;\n\tfont-weight: normal;\n\tmargin: 20px -15px 10px -13px;\n\theight:
 40px;\n\twidth: 108%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;\n\ttext-align:
 center;\n}\n.submitButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:
 #286090;\n}\n.errorMessage-customizable {\n\tpadding: 5px;\n\tfont-size: 14px;
\n\twidth: 100%;\n\tbackground: #F5F5F5;\n\tborder: 2px solid #D64958;\n\tcolor:
 #D64958;\n}\n.inputField-customizable {\n\twidth: 100%;\n\theight: 34px;\n\tcolor:
 #555;\n\tbackground-color: #fff;\n\tborder: 1px solid #ccc;\n\tborder-radius:
 0px;\n}\n.inputField-customizable:focus {\n\tborder-color: #66afe9;\n\toutline:
 0;\n}\n.idpButton-customizable {\n\theight: 40px;\n\twidth: 100%;\n\twidth: 100%;
\n\ttext-align: center;\n\tmargin-bottom: 15px;\n\tcolor: #fff;\n\tbackground-
color: #5bc0de;\n\tborder-color: #46b8da;\n}\n.idpButton-customizable:hover {\n
\tcolor: #fff;\n\tbackground-color: #31b0d5;\n}\n.socialButton-customizable {\n
\tborder-radius: 2px;\n\theight: 40px;\n\tmargin-bottom: 15px;\n\tpadding: 1px;
```

```
\n\ttext-align: left;\n\twidth: 100%;\n}\n.redirect-customizable {\n\ttext-
align: center;\n}\n.passwordCheck-notValid-customizable {\n\tcolor: #DF3312;
\n}\n.passwordCheck-valid-customizable {\n\tcolor: #19BF00;\n}\n.background-
customizable {\n\tbackground-color: #fff;\n}\n" \
    --image-
file iVBORw0KGgoAAAANSUhEUgAAAFAAAABQCAMAAAC5zwKfAAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAA
Cmsfvm6f3y9P////fM0uqAj
+yNmu6ZpvnZ3eNabuFNYuZneehzhPKzvPTAxwAAAOiMMlkAAAASdFJOU//////////////////////
AOK/vxIAAAAJcEhZcwAADsMAAA7DAcdvqGQAAAKDSURBVFhH7ZfpkoMgEISDHKuEw/
d/2u2BQWMiBrG29o+fVsKatdPMAeZxc3Nz8w+ISekzmB++sYIw/I/
tjHzrPpO2Tx62EbR2PNxFac+jVuKxRaV50IzXkUe76NOCoUuwlvnQKei02gNF0ykotOLRBq/
nboeWRxAISx2EbsHFoRhK6Igk2JJlwScfQjgt06dOaWWiTbEDAe/
iq8N9kqCw2uCbHkHlYkaXEF8EYeL9RDqT4FhC6XMIIEifdcUwCc4leNyhabadWU6OlKYJE1Oac3NSPhB5rlaXlSgmr/1
F0L6Q5pZiSG0SfZTSTCOUhxOCH1AdIoCpTTIjtd
+VpEjUDDytQH/0Fpc661Aisas/4qmyUItD557pSCOSQQzlx27J
+meyDGc5zZgfhWuXE1lGgmVOMwmWdeGdzhjqZV14x5vSj7vsC5JDz/Cl0Vhp56n2NQt1wQIpury1EPbwyaYm
+IhmAQKoajkH51wg4cMZ1wQ3QG9efKWWOaDhYWnU6jXjCMdRmm21PArI
+Pb5DYoH93hq0ZCPlxeGJho/DI15C6sQc/L2sTC47UFBKZGHT6k+zlXg7WebA0Nr0HTcLMfk/
Y4Rc65D3iG6WDd7YLSlVqk87bVhUwhnClrx11RsVQwlAA818Mn
+QEs71BhSFU6orsUfKhHp72XMGYXi4q9c64RXRvzkWurRfG2vI2be/VaNcNgpX0Evb/
vio7nPMmj5qujkpQgSaPd1UcVqciHFDNZpOcGlcOPyi+AamCbIL9fitxAGeFN2Dl
+3vZubm5u/4fH4Bd14HhIPdwZPAAAAAElFTkSuQmCC
```

輸出：

```
{
    "UICustomization": {
        "UserPoolId": "us-west-2_ywDJHlIfU",
        "ClientId": "14pq32c5q2uq2q7keorloqvb23",
        "ImageUrl": "https://
cf.thewrong.club/14pq32c5q2uq2q7keorloqvb23/20250117005911/assets/images/image.jpg",
        "CSS": ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;
\n}\n.banner-customizable {\n\tpadding: 25px 0px 25px 0px;\n\tbackground-color:
 lightgray;\n}\n.label-customizable {\n\tfont-weight: 400;\n}\n.textDescription-
customizable {\n\tpadding-top: 10px;\n\tpadding-bottom: 10px;\n\tdisplay: block;
\n\tfont-size: 16px;\n}\n.idpDescription-customizable {\n\tpadding-top: 10px;\n
\tpadding-bottom: 10px;\n\tdisplay: block;\n\tfont-size: 16px;\n}\n.legalText-
customizable {\n\tcolor: #747474;\n\tfont-size: 11px;\n}\n.submitButton-customizable
 {\n\tfont-size: 11px;\n\tfont-weight: normal;\n\tmargin: 20px -15px 10px -13px;
\n\theight: 40px;\n\twidth: 108%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;
\n\ttext-align: center;\n}\n.submitButton-customizable:hover {\n\tcolor: #fff;
\n\tbackground-color: #286090;\n}\n.errorMessage-customizable {\n\tpadding:
 5px;\n\tfont-size: 14px;\n\twidth: 100%;\n\tbackground: #F5F5F5;\n\tborder:
```

```
 2px solid #D64958;\n\tcolor: #D64958;\n}\n.inputField-customizable {\n\twidth:
 100%;\n\theight: 34px;\n\tcolor: #555;\n\tbackground-color: #fff;\n\tborder:
 1px solid #ccc;\n\tborder-radius: 0px;\n}\n.inputField-customizable:focus {\n
\tborder-color: #66afe9;\n\toutline: 0;\n}\n.idpButton-customizable {\n\theight:
 40px;\n\twidth: 100%;\n\twidth: 100%;\n\ttext-align: center;\n\tmargin-bottom:
 15px;\n\tcolor: #fff;\n\tbackground-color: #5bc0de;\n\tborder-color: #46b8da;
\n}\n.idpButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color: #31b0d5;
\n}\n.socialButton-customizable {\n\tborder-radius: 2px;\n\theight: 40px;\n\tmargin-
bottom: 15px;\n\tpadding: 1px;\n\ttext-align: left;\n\twidth: 100%;\n}\n.redirect-
customizable {\n\ttext-align: center;\n}\n.passwordCheck-notValid-customizable
 {\n\tcolor: #DF3312;\n}\n.passwordCheck-valid-customizable {\n\tcolor: #19BF00;
\n}\n.background-customizable {\n\tbackground-color: #fff;\n}\n",
        "CSSVersion": "20250117005911"
    }
}
```

範例 2：為所有應用程式用戶端設定預設 UI 自訂

下列set-ui-customization範例會為沒有用戶端特定組態的所有應用程式用戶端設定請求的使用者集區。命令會套用一些自訂 CSS，並以 Amazon Cognito 標誌做為應用程式標誌。

```
aws cognito-idp set-ui-customization \
--user-pool-id us-west-2_ywDJHlIfU \
--client-id ALL \
--css ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;\n}\n.banner-
customizable {\n\tpadding: 25px 0px 25px 0px;\n\tbackground-color: lightgray;
\n}\n.label-customizable {\n\tfont-weight: 400;\n}\n.textDescription-customizable
 {\n\tpadding-top: 10px;\n\tpadding-bottom: 10px;\n\tdisplay: block;\n\tfont-
size: 16px;\n}\n.idpDescription-customizable {\n\tpadding-top: 10px;\n\tpadding-
bottom: 10px;\n\tdisplay: block;\n\tfont-size: 16px;\n}\n.legalText-customizable
 {\n\tcolor: #747474;\n\tfont-size: 11px;\n}\n.submitButton-customizable {\n\tfont-
size: 11px;\n\tfont-weight: normal;\n\tmargin: 20px -15px 10px -13px;\n\theight:
 40px;\n\twidth: 108%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;\n\ttext-align:
 center;\n}\n.submitButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:
 #286090;\n}\n.errorMessage-customizable {\n\tpadding: 5px;\n\tfont-size: 14px;
\n\twidth: 100%;\n\tbackground: #F5F5F5;\n\tborder: 2px solid #D64958;\n\tcolor:
 #D64958;\n}\n.inputField-customizable {\n\twidth: 100%;\n\theight: 34px;\n\tcolor:
 #555;\n\tbackground-color: #fff;\n\tborder: 1px solid #ccc;\n\tborder-radius:
 0px;\n}\n.inputField-customizable:focus {\n\tborder-color: #66afe9;\n\toutline:
 0;\n}\n.idpButton-customizable {\n\theight: 40px;\n\twidth: 100%;\n\twidth: 100%;
\n\ttext-align: center;\n\tmargin-bottom: 15px;\n\tcolor: #fff;\n\tbackground-
color: #5bc0de;\n\tborder-color: #46b8da;\n}\n.idpButton-customizable:hover {\n
\tcolor: #fff;\n\tbackground-color: #31b0d5;\n}\n.socialButton-customizable {\n
```

```
\tborder-radius: 2px;\n\theight: 40px;\n\tmargin-bottom: 15px;\n\tpadding: 1px;
\n\ttext-align: left;\n\twidth: 100%;\n}\n.redirect-customizable {\n\ttext-
align: center;\n}\n.passwordCheck-notValid-customizable {\n\tcolor: #DF3312;
\n}\n.passwordCheck-valid-customizable {\n\tcolor: #19BF00;\n}\n.background-
customizable {\n\tbackground-color: #fff;\n}\n" \
--image-
file iVBORw0KGgoAAAANSUhEUgAAAFAAAABQCAMAAAC5zwKfAAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAA
Cmsfvm6f3y9P////fM0uqAj
+yNmu6ZpvnZ3eNabuFNYuZneehzhPKzvPTAxwAAAOiMMlkAAAASdFJOU//////////////////////
AOK/vxIAAAAJcEhZcwAADsMAAA7DAcdvqGQAAAKDSURBVFhH7ZfpkoMgEISDHKuEw/
d/2u2BQWMiBrG29o+fVsKatdPMAeZxc3Nz8w+ISekzmB++sYIw/I/
tjHzrPpO2Tx62EbR2PNxFac+jVuKxRaV50IzXkUe76NOCoUuwlvnQKei02gNF0ykotOLRBq/
nboeWRxAISx2EbsHFoRhK6Igk2JJlwScfQjgt06dOaWWiTbEDAe/
iq8N9kqCw2uCbHkHlYkaXEF8EYeL9RDqT4FhC6XMIIEifdcUwCc4leNyhabadWU6OlKYJE1Oac3NSPhB5rlaXlSgmr/1
F0L6Q5pZiSG0SfZTSTCOUhxOCH1AdIoCpTTIjtd
+VpEjUDDytQH/0Fpc661Aisas/4qmyUItD557pSCOSQQzlx27J
+meyDGc5zZgfhWuXE1lGgmVOMwmWdeGdzhjqZV14x5vSj7vsC5JDz/Cl0Vhp56n2NQt1wQIpury1EPbwyaYm
+IhmAQKoajkH51wg4cMZ1wQ3QG9efKWWOaDhYWnU6jXjCMdRmm21PArI
+Pb5DYoH93hq0ZCPlxeGJho/DI15C6sQc/L2sTC47UFBKZGHT6k+zlXg7WebA0Nr0HTcLMfk/
Y4Rc65D3iG6WDd7YLSlVqk87bVhUwhnClrx11RsVQwlAA818Mn
+QEs71BhSFU6orsUfKhHp72XMGYXi4q9c64RXRvzkWurRfG2vI2be/VaNcNgpX0Evb/
vio7nPMmj5qujkpQgSaPd1UcVqciHFDNZpOcGlcOPyi+AamCbIL9fitxAGeFN2Dl
+3vZubm5u/4fH4Bd14HhIPdwZPAAAAAElFTkSuQmCC
```

輸出：

```
{
    "UICustomization": {
        "UserPoolId": "us-west-2_ywDJHlIfU",
        "ClientId": "14pq32c5q2uq2q7keorloqvb23",
        "ImageUrl": "https://
cf.thewrong.club/14pq32c5q2uq2q7keorloqvb23/20250117005911/assets/images/image.jpg",
        "CSS": ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;
\n}\n.banner-customizable {\n\tpadding: 25px 0px 25px 0px;\n\tbackground-color:
 lightgray;\n}\n.label-customizable {\n\tfont-weight: 400;\n}\n.textDescription-
customizable {\n\tpadding-top: 10px;\n\tpadding-bottom: 10px;\n\tdisplay: block;
\n\tfont-size: 16px;\n}\n.idpDescription-customizable {\n\tpadding-top: 10px;\n
\tpadding-bottom: 10px;\n\tdisplay: block;\n\tfont-size: 16px;\n}\n.legalText-
customizable {\n\tcolor: #747474;\n\tfont-size: 11px;\n}\n.submitButton-customizable
 {\n\tfont-size: 11px;\n\tfont-weight: normal;\n\tmargin: 20px -15px 10px -13px;
\n\theight: 40px;\n\twidth: 108%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;
\n\ttext-align: center;\n}\n.submitButton-customizable:hover {\n\tcolor: #fff;
\n\tbackground-color: #286090;\n}\n.errorMessage-customizable {\n\tpadding:
```

```
 5px;\n\tfont-size: 14px;\n\twidth: 100%;\n\tbackground: #F5F5F5;\n\tborder:
 2px solid #D64958;\n\tcolor: #D64958;\n}\n.inputField-customizable {\n\twidth:
 100%;\n\theight: 34px;\n\tcolor: #555;\n\tbackground-color: #fff;\n\tborder:
 1px solid #ccc;\n\tborder-radius: 0px;\n}\n.inputField-customizable:focus {\n
\tborder-color: #66afe9;\n\toutline: 0;\n}\n.idpButton-customizable {\n\theight:
 40px;\n\twidth: 100%;\n\twidth: 100%;\n\ttext-align: center;\n\tmargin-bottom:
 15px;\n\tcolor: #fff;\n\tbackground-color: #5bc0de;\n\tborder-color: #46b8da;
\n}\n.idpButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color: #31b0d5;
\n}\n.socialButton-customizable {\n\tborder-radius: 2px;\n\theight: 40px;\n\tmargin-
bottom: 15px;\n\tpadding: 1px;\n\ttext-align: left;\n\twidth: 100%;\n}\n.redirect-
customizable {\n\ttext-align: center;\n}\n.passwordCheck-notValid-customizable
 {\n\tcolor: #DF3312;\n}\n.passwordCheck-valid-customizable {\n\tcolor: #19BF00;
\n}\n.background-customizable {\n\tbackground-color: #fff;\n}\n",
        "CSSVersion": "20250117005911"
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的託管 UI（傳統）品牌。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetUiCustomization。

set-user-mfa-preference

以下程式碼範例顯示如何使用 set-user-mfa-preference。

AWS CLI

設定使用者的 MFA 偏好設定

下列set-user-mfa-preference範例會將目前使用者設定為使用 TOTP MFA，並停用所有其他 MFA 因素。

```
aws cognito-idp set-user-mfa-preference \
    --access-token eyJra456defEXAMPLE \
    --software-token-mfa-settings Enabled=true,PreferredMfa=true \
    --sms-mfa-settings Enabled=false,PreferredMfa=false \
    --email-mfa-settings Enabled=false,PreferredMfa=false
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的新增 MFA。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetUserMfaPreference。

set-user-pool-mfa-config

以下程式碼範例顯示如何使用 set-user-pool-mfa-config。

AWS CLI

設定使用者集區 MFA 和 WebAuthn

下列set-user-pool-mfa-config範例會使用所有可用的 MFA 方法，使用選用的 MFA 設定請求的使用者集區，並設定 WebAuthn 組態。

```
aws cognito-idp set-user-pool-mfa-config \
    --user-pool-id us-west-2_EXAMPLE \
    --sms-mfa-configuration "SmsAuthenticationMessage=\"Your OTP for MFA or sign-
in: use {####}.\",SmsConfiguration={SnsCallerArn=arn:aws:iam::123456789012:role/
service-role/test-SMS-Role,ExternalId=a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111,SnsRegion=us-west-2}" \
    --software-token-mfa-configuration Enabled=true \
    --email-mfa-configuration "Message=\"Your OTP for MFA or sign-in: use
 {####}\",Subject=\"OTP test\"" \
    --mfa-configuration OPTIONAL \
    --web-authn-
configuration RelyingPartyId=auth.example.com,UserVerification=preferred
```

輸出：

```
{
    "EmailMfaConfiguration": {
        "Message": "Your OTP for MFA or sign-in: use {####}",
        "Subject": "OTP test"
    },
    "MfaConfiguration": "OPTIONAL",
    "SmsMfaConfiguration": {
        "SmsAuthenticationMessage": "Your OTP for MFA or sign-in: use {####}.",
        "SmsConfiguration": {
            "ExternalId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "SnsCallerArn": "arn:aws:iam::123456789012:role/service-role/test-SMS-
Role",
            "SnsRegion": "us-west-2"
        }
    },
    "SoftwareTokenMfaConfiguration": {
        "Enabled": true
```

```
    },
    "WebAuthnConfiguration": {
        "RelyingPartyId": "auth.example.com",
        "UserVerification": "preferred"
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的新增 MFA 和通行金鑰登入。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetUserPoolMfaConfig。

set-user-settings

以下程式碼範例顯示如何使用 set-user-settings。

AWS CLI

設定使用者設定

此範例會將 MFA 交付偏好設定設定為 EMAIL。

命令：

```
aws cognito-idp set-user-settings --access-token ACCESS_TOKEN --mfa-
options DeliveryMedium=EMAIL
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetUserSettings。

sign-up

以下程式碼範例顯示如何使用 sign-up。

AWS CLI

若要將使用者登出

此範例註冊了 jane@example.com。

命令：

```
aws cognito-idp sign-up --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username jane@example.com --password PASSWORD --user-attributes
 Name="email",Value="jane@example.com" Name="name",Value="Jane"
```

輸出：

```
{
  "UserConfirmed": false,
  "UserSub": "e04d60a6-45dc-441c-a40b-e25a787d4862"
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 SignUp。

start-user-import-job

以下程式碼範例顯示如何使用 start-user-import-job。

AWS CLI

啟動匯入任務

下列start-user-import-job範例會在請求的使用者集區中啟動請求的匯入任務。

```
aws cognito-idp start-user-import-job \
    --user-pool-id us-west-2_EXAMPLE \
    --job-id import-mAgUtd8PMm
```

輸出：

```
{
    "UserImportJob": {
        "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/example-cloudwatch-
logs-role",
        "CreationDate": 1736442975.904,
        "FailedUsers": 0,
        "ImportedUsers": 0,
        "JobId": "import-mAgUtd8PMm",
        "JobName": "Customer import",
        "PreSignedUrl": "https://aws-cognito-idp-user-import-pdx.s3.us-
west-2.amazonaws.com/123456789012/us-west-2_EXAMPLE/import-mAgUtd8PMm?
X-Amz-Security-Token=[token]&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20241226T193341Z&X-Amz-SignedHeaders=host%3Bx-amz-server-side-encryption&X-Amz-
Expires=899&X-Amz-Credential=[credential]&X-Amz-Signature=[signature]",
        "SkippedUsers": 0,
        "StartDate": 1736443020.081,
        "Status": "Pending",
```

```
        "UserPoolId": "us-west-2_EXAMPLE"
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的將使用者匯入使用者集區。Amazon Cognito

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartUserImportJob。

`start-web-authn-registration`

以下程式碼範例顯示如何使用 `start-web-authn-registration`。

AWS CLI

取得登入使用者的通行金鑰註冊資訊

下列`start-web-authn-registration`範例會為目前使用者產生 WebAuthn 註冊選項。

```
aws cognito-idp start-web-authn-registration \
    --access-token eyJra456defEXAMPLE
```

輸出：

```
{
    "CredentialCreationOptions": {
        "authenticatorSelection": {
            "requireResidentKey": true,
            "residentKey": "required",
            "userVerification": "preferred"
        },
        "challenge": "wxvbDicyqQqvF2EXAMPLE",
        "excludeCredentials": [
            {
                "id": "8LApgk4-lNUFHbhm2w6Und7-
uxcc8coJGsPxiogvHoItc64xWQc3r4CEXAMPLE",
                "type": "public-key"
            }
        ],
        "pubKeyCredParams": [
            {
                "alg": -7,
                "type": "public-key"
```

```
            },
            {
                "alg": -257,
                "type": "public-key"
            }
        ],
        "rp": {
            "id": "auth.example.com",
            "name": "auth.example.com"
        },
        "timeout": 60000,
        "user": {
            "displayName": "testuser",
            "id": "ZWFhZDAyMTktMjExNy00MzlmLThkNDYtNGRiMjBlNEXAMPLE",
            "name": "testuser"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的通行金鑰登入。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartWebAuthnRegistration。

stop-user-import-job

以下程式碼範例顯示如何使用 stop-user-import-job。

AWS CLI

停止匯入任務

下列stop-user-import-job範例會在請求的使用者集區中停止請求的執行中使用者匯入任務。

```
aws cognito-idp stop-user-import-job \
    --user-pool-id us-west-2_EXAMPLE \
    --job-id import-mAgUtd8PMm
```

輸出：

```
{
    "UserImportJob": {
        "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/example-cloudwatch-
logs-role",
```

```
        "CompletionDate": 1736443496.379,
        "CompletionMessage": "The Import Job was stopped by the developer.",
        "CreationDate": 1736443471.781,
        "FailedUsers": 0,
        "ImportedUsers": 0,
        "JobId": "import-mAgUtd8PMm",
        "JobName": "Customer import",
        "PreSignedUrl": "https://aws-cognito-idp-user-import-pdx.s3.us-
west-2.amazonaws.com/123456789012/us-west-2_EXAMPLE/import-mAgUtd8PMm?
X-Amz-Security-Token=[token]&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20241226T193341Z&X-Amz-SignedHeaders=host%3Bx-amz-server-side-encryption&X-Amz-
Expires=899&X-Amz-Credential=[credential]&X-Amz-Signature=[signature]",
        "SkippedUsers": 0,
        "StartDate": 1736443494.154,
        "Status": "Stopped",
        "UserPoolId": "us-west-2_EXAMPLE"
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的將使用者匯入使用者集區。Amazon Cognito

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopUserImportJob。

`tag-resource`

以下程式碼範例顯示如何使用 `tag-resource`。

AWS CLI

標記使用者集區

下列`tag-resource`範例會將 `administrator`和 `department`標籤套用至請求的使用者集區。

```
aws cognito-idp tag-resource \
    --resource-arn arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_EXAMPLE \
    --tags administrator=Jie,tenant=ExampleCorp
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的標記 Amazon Cognito 資源。Amazon Cognito

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從使用者集區移除標籤

下列untag-resource範例會從請求的使用者集區中移除 administrator和 department 標籤。

```
aws cognito-idp untag-resource \
    --resource-arn arn:aws:cognito-idp:us-west-2:767671399759:userpool/us-west-2_l5cxwdm2K \
    --tag-keys administrator tenant
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的標記 Amazon Cognito 資源。
Amazon Cognito

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-auth-event-feedback

以下程式碼範例顯示如何使用 update-auth-event-feedback。

AWS CLI

更新身分驗證事件意見回饋

此範例會更新授權事件意見回饋。它會標記事件「有效」。

命令：

```
aws cognito-idp update-auth-event-feedback --user-pool-id us-west-2_aaaaaaaaa --username diego@example.com --event-id EVENT_ID --feedback-token FEEDBACK_TOKEN --feedback-value "Valid"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAuthEventFeedback。

update-device-status

以下程式碼範例顯示如何使用 update-device-status。

AWS CLI

更新裝置狀態

此範例會將裝置的狀態更新為「not_remembered」。

命令：

```
aws cognito-idp update-device-status --access-token ACCESS_TOKEN --device-
key DEVICE_KEY --device-remembered-status "not_remembered"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDeviceStatus。

update-group

以下程式碼範例顯示如何使用 update-group。

AWS CLI

更新群組

此範例會更新 MyGroup 的描述和優先順序。

命令：

```
aws cognito-idp update-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroup
 --description "New description" --precedence 2
```

輸出：

```
{
  "Group": {
      "GroupName": "MyGroup",
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "Description": "New description",
      "RoleArn": "arn:aws:iam::111111111111:role/MyRole",
      "Precedence": 2,
      "LastModifiedDate": 1548800862.812,
```

```
      "CreationDate": 1548097827.125
  }
}
```

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UpdateGroup。

update-identity-provider

以下程式碼範例顯示如何使用 update-identity-provider。

AWS CLI

更新使用者集區身分提供者

下列update-identity-provider範例會在請求的使用者集區中更新 OIDC 提供者 "MyOIDCIdP"。

```
aws cognito-idp update-identity-provider \
    --cli-input-json file://update-identity-provider.json
```

update-identity-provider.json 的內容：

```
{
   "AttributeMapping": {
            "email": "idp_email",
            "email_verified": "idp_email_verified",
            "username": "sub"
    },
    "CreationDate": 1.701129701653E9,
    "IdpIdentifiers": [
        "corp",
        "dev"
    ],
    "LastModifiedDate": 1.701129701653E9,
    "ProviderDetails": {
        "attributes_request_method": "GET",
        "attributes_url": "https://example.com/userInfo",
        "attributes_url_add_attributes": "false",
        "authorize_scopes": "openid profile",
        "authorize_url": "https://example.com/authorize",
        "client_id": "idpexampleclient123",
        "client_secret": "idpexamplesecret456",
```

```
        "jwks_uri": "https://example.com/.well-known/jwks.json",
        "oidc_issuer": "https://example.com",
        "token_url": "https://example.com/token"
    },
    "ProviderName": "MyOIDCIdP",
    "UserPoolId": "us-west-2_EXAMPLE"
}
```

輸出：

```
{
    "IdentityProvider": {
        "AttributeMapping": {
            "email": "idp_email",
            "email_verified": "idp_email_verified",
            "username": "sub"
        },
        "CreationDate": 1701129701.653,
        "IdpIdentifiers": [
            "corp",
            "dev"
        ],
        "LastModifiedDate": 1736444278.211,
        "ProviderDetails": {
            "attributes_request_method": "GET",
            "attributes_url": "https://example.com/userInfo",
            "attributes_url_add_attributes": "false",
            "authorize_scopes": "openid profile",
            "authorize_url": "https://example.com/authorize",
            "client_id": "idpexampleclient123",
            "client_secret": "idpexamplesecret456",
            "jwks_uri": "https://example.com/.well-known/jwks.json",
            "oidc_issuer": "https://example.com",
            "token_url": "https://example.com/token"
        },
        "ProviderName": "MyOIDCIdP",
        "ProviderType": "OIDC",
        "UserPoolId": "us-west-2_EXAMPLE"
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的設定網域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateIdentityProvider。

update-managed-login-branding

以下程式碼範例顯示如何使用 update-managed-login-branding。

AWS CLI

更新受管登入品牌樣式

下列update-managed-login-branding範例會更新請求的應用程式用戶端品牌樣式。

```
aws cognito-idp update-managed-login-branding \
    --cli-input-json file://update-managed-login-branding.json
```

update-managed-login-branding.json 的內容：

```
{
    "Assets": [
        {
            "Bytes":
 "PHN2ZyB3aWR0aD0iMjAwMDAiIGhlaWdodD0iNDAwIiB2aWV3Qm94PSIwIDAgMjAwMDAgNDAwIiBmaWxsPSJub25lIi
+CjxyZWN0IHdpZHRoPSIyMDAwMCIgaGVpZ2h0PSI0MDAiIGZpbGw9InVybCgjcGFpbnQwX2xpbmVhcl8xNzI1OV8yMzY
+CjxsaW5lYXJHcmFkaWVudCBpZD0icGFpbnQwX2xpbmVhcl8xNzI1OV8yMzY2NzQiIHgxPSItODk0LjI0OSIgeTE9IjE
+Cjwvc3ZnPgo=",
            "Category": "PAGE_FOOTER_BACKGROUND",
            "ColorMode": "DARK",
            "Extension": "SVG"
        }
    ],
    "ManagedLoginBrandingId": "63f30090-6b1f-4278-b885-2bbb81f8e545",
    "Settings": {
        "categories": {
            "auth": {
                "authMethodOrder": [
                    [
                        {
                            "display": "BUTTON",
                            "type": "FEDERATED"
                        },
                        {
                            "display": "INPUT",
                            "type": "USERNAME_PASSWORD"
                        }
                    ]
                ],
```

```
                "federation": {
                    "interfaceStyle": "BUTTON_LIST",
                    "order": [
                    ]
                }
            },
            "form": {
                "displayGraphics": true,
                "instructions": {
                    "enabled": false
                },
                "languageSelector": {
                    "enabled": false
                },
                "location": {
                    "horizontal": "CENTER",
                    "vertical": "CENTER"
                },
                "sessionTimerDisplay": "NONE"
            },
            "global": {
                "colorSchemeMode": "LIGHT",
                "pageFooter": {
                    "enabled": false
                },
                "pageHeader": {
                    "enabled": false
                },
                "spacingDensity": "REGULAR"
            },
            "signUp": {
                "acceptanceElements": [
                    {
                        "enforcement": "NONE",
                        "textKey": "en"
                    }
                ]
            }
        },
        "componentClasses": {
            "buttons": {
                "borderRadius": 8.0
            },
            "divider": {
```

```
                "darkMode": {
                    "borderColor": "232b37ff"
                },
                "lightMode": {
                    "borderColor": "ebebf0ff"
                }
            },
            "dropDown": {
                "borderRadius": 8.0,
                "darkMode": {
                    "defaults": {
                        "itemBackgroundColor": "192534ff"
                    },
                    "hover": {
                        "itemBackgroundColor": "081120ff",
                        "itemBorderColor": "5f6b7aff",
                        "itemTextColor": "e9ebedff"
                    },
                    "match": {
                        "itemBackgroundColor": "d1d5dbff",
                        "itemTextColor": "89bdeeff"
                    }
                },
                "lightMode": {
                    "defaults": {
                        "itemBackgroundColor": "ffffffff"
                    },
                    "hover": {
                        "itemBackgroundColor": "f4f4f4ff",
                        "itemBorderColor": "7d8998ff",
                        "itemTextColor": "000716ff"
                    },
                    "match": {
                        "itemBackgroundColor": "414d5cff",
                        "itemTextColor": "0972d3ff"
                    }
                }
            },
            "focusState": {
                "darkMode": {
                    "borderColor": "539fe5ff"
                },
                "lightMode": {
                    "borderColor": "0972d3ff"
```

```
            }
        },
        "idpButtons": {
            "icons": {
                "enabled": true
            }
        },
        "input": {
            "borderRadius": 8.0,
            "darkMode": {
                "defaults": {
                    "backgroundColor": "0f1b2aff",
                    "borderColor": "5f6b7aff"
                },
                "placeholderColor": "8d99a8ff"
            },
            "lightMode": {
                "defaults": {
                    "backgroundColor": "ffffffff",
                    "borderColor": "7d8998ff"
                },
                "placeholderColor": "5f6b7aff"
            }
        },
        "inputDescription": {
            "darkMode": {
                "textColor": "8d99a8ff"
            },
            "lightMode": {
                "textColor": "5f6b7aff"
            }
        },
        "inputLabel": {
            "darkMode": {
                "textColor": "d1d5dbff"
            },
            "lightMode": {
                "textColor": "000716ff"
            }
        },
        "link": {
            "darkMode": {
                "defaults": {
                    "textColor": "539fe5ff"
```

```
                },
                "hover": {
                    "textColor": "89bdeeff"
                }
            },
            "lightMode": {
                "defaults": {
                    "textColor": "0972d3ff"
                },
                "hover": {
                    "textColor": "033160ff"
                }
            }
        },
        "optionControls": {
            "darkMode": {
                "defaults": {
                    "backgroundColor": "0f1b2aff",
                    "borderColor": "7d8998ff"
                },
                "selected": {
                    "backgroundColor": "539fe5ff",
                    "foregroundColor": "000716ff"
                }
            },
            "lightMode": {
                "defaults": {
                    "backgroundColor": "ffffffff",
                    "borderColor": "7d8998ff"
                },
                "selected": {
                    "backgroundColor": "0972d3ff",
                    "foregroundColor": "ffffffff"
                }
            }
        },
        "statusIndicator": {
            "darkMode": {
                "error": {
                    "backgroundColor": "1a0000ff",
                    "borderColor": "eb6f6fff",
                    "indicatorColor": "eb6f6fff"
                },
                "pending": {
```

```
                    "indicatorColor": "AAAAAAAA"
                },
                "success": {
                    "backgroundColor": "001a02ff",
                    "borderColor": "29ad32ff",
                    "indicatorColor": "29ad32ff"
                },
                "warning": {
                    "backgroundColor": "1d1906ff",
                    "borderColor": "e0ca57ff",
                    "indicatorColor": "e0ca57ff"
                }
            },
            "lightMode": {
                "error": {
                    "backgroundColor": "fff7f7ff",
                    "borderColor": "d91515ff",
                    "indicatorColor": "d91515ff"
                },
                "pending": {
                    "indicatorColor": "AAAAAAAA"
                },
                "success": {
                    "backgroundColor": "f2fcf3ff",
                    "borderColor": "037f0cff",
                    "indicatorColor": "037f0cff"
                },
                "warning": {
                    "backgroundColor": "fffce9ff",
                    "borderColor": "8d6605ff",
                    "indicatorColor": "8d6605ff"
                }
            }
        }
    },
    "components": {
        "alert": {
            "borderRadius": 12.0,
            "darkMode": {
                "error": {
                    "backgroundColor": "1a0000ff",
                    "borderColor": "eb6f6fff"
                }
            },
```

```
                "lightMode": {
                    "error": {
                        "backgroundColor": "fff7f7ff",
                        "borderColor": "d91515ff"
                    }
                }
            },
            "favicon": {
                "enabledTypes": [
                    "ICO",
                    "SVG"
                ]
            },
            "form": {
                "backgroundImage": {
                    "enabled": false
                },
                "borderRadius": 8.0,
                "darkMode": {
                    "backgroundColor": "0f1b2aff",
                    "borderColor": "424650ff"
                },
                "lightMode": {
                    "backgroundColor": "ffffffff",
                    "borderColor": "c6c6cdff"
                },
                "logo": {
                    "enabled": false,
                    "formInclusion": "IN",
                    "location": "CENTER",
                    "position": "TOP"
                }
            },
            "idpButton": {
                "custom": {
                },
                "standard": {
                    "darkMode": {
                        "active": {
                            "backgroundColor": "354150ff",
                            "borderColor": "89bdeeff",
                            "textColor": "89bdeeff"
                        },
                        "defaults": {
```

```
                        "backgroundColor": "0f1b2aff",
                        "borderColor": "c6c6cdff",
                        "textColor": "c6c6cdff"
                    },
                    "hover": {
                        "backgroundColor": "192534ff",
                        "borderColor": "89bdeeff",
                        "textColor": "89bdeeff"
                    }
                },
                "lightMode": {
                    "active": {
                        "backgroundColor": "d3e7f9ff",
                        "borderColor": "033160ff",
                        "textColor": "033160ff"
                    },
                    "defaults": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "424650ff",
                        "textColor": "424650ff"
                    },
                    "hover": {
                        "backgroundColor": "f2f8fdff",
                        "borderColor": "033160ff",
                        "textColor": "033160ff"
                    }
                }
            }
        },
        "pageBackground": {
            "darkMode": {
                "color": "0f1b2aff"
            },
            "image": {
                "enabled": true
            },
            "lightMode": {
                "color": "ffffffff"
            }
        },
        "pageFooter": {
            "backgroundImage": {
                "enabled": false
            },
```

```
                "darkMode": {
                    "background": {
                        "color": "0f141aff"
                    },
                    "borderColor": "424650ff"
                },
                "lightMode": {
                    "background": {
                        "color": "fafafaff"
                    },
                    "borderColor": "d5dbdbff"
                },
                "logo": {
                    "enabled": false,
                    "location": "START"
                }
            },
            "pageHeader": {
                "backgroundImage": {
                    "enabled": false
                },
                "darkMode": {
                    "background": {
                        "color": "0f141aff"
                    },
                    "borderColor": "424650ff"
                },
                "lightMode": {
                    "background": {
                        "color": "fafafaff"
                    },
                    "borderColor": "d5dbdbff"
                },
                "logo": {
                    "enabled": false,
                    "location": "START"
                }
            },
            "pageText": {
                "darkMode": {
                    "bodyColor": "b6bec9ff",
                    "descriptionColor": "b6bec9ff",
                    "headingColor": "d1d5dbff"
                },
```

```
                "lightMode": {
                    "bodyColor": "414d5cff",
                    "descriptionColor": "414d5cff",
                    "headingColor": "000716ff"
                }
            },
            "phoneNumberSelector": {
                "displayType": "TEXT"
            },
            "primaryButton": {
                "darkMode": {
                    "active": {
                        "backgroundColor": "539fe5ff",
                        "textColor": "000716ff"
                    },
                    "defaults": {
                        "backgroundColor": "539fe5ff",
                        "textColor": "000716ff"
                    },
                    "disabled": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "ffffffff"
                    },
                    "hover": {
                        "backgroundColor": "89bdeeff",
                        "textColor": "000716ff"
                    }
                },
                "lightMode": {
                    "active": {
                        "backgroundColor": "033160ff",
                        "textColor": "ffffffff"
                    },
                    "defaults": {
                        "backgroundColor": "0972d3ff",
                        "textColor": "ffffffff"
                    },
                    "disabled": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "ffffffff"
                    },
                    "hover": {
                        "backgroundColor": "033160ff",
                        "textColor": "ffffffff"
```

```
                    }
                }
            },
            "secondaryButton": {
                "darkMode": {
                    "active": {
                        "backgroundColor": "354150ff",
                        "borderColor": "89bdeeff",
                        "textColor": "89bdeeff"
                    },
                    "defaults": {
                        "backgroundColor": "0f1b2aff",
                        "borderColor": "539fe5ff",
                        "textColor": "539fe5ff"
                    },
                    "hover": {
                        "backgroundColor": "192534ff",
                        "borderColor": "89bdeeff",
                        "textColor": "89bdeeff"
                    }
                },
                "lightMode": {
                    "active": {
                        "backgroundColor": "d3e7f9ff",
                        "borderColor": "033160ff",
                        "textColor": "033160ff"
                    },
                    "defaults": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "0972d3ff",
                        "textColor": "0972d3ff"
                    },
                    "hover": {
                        "backgroundColor": "f2f8fdff",
                        "borderColor": "033160ff",
                        "textColor": "033160ff"
                    }
                }
            }
        }
    },
    "UseCognitoProvidedValues": false,
    "UserPoolId": "ca-central-1_EXAMPLE"
```

```
}
```

輸出：

```
{
    "ManagedLoginBranding": {
        "Assets": [
            {
                "Bytes":
 "PHN2ZyB3aWR0aD0iMjAwMDAiIGhlaWdodD0iNDAwIiB2aWV3Qm94PSIwIDAgMjAwMDAgNDAwIiBmaWxsPSJub25lIi
+CjxyZWN0IHdpZHRoPSIyMDAwMCIgaGVpZ2h0PSI0MDAiIGZpbGw9InVybCgjcGFpbnQwX2xpbmVhcl8xNzI1OV8yMzY
+CjxsaW5lYXJHcmFkaWVudCBpZD0icGFpbnQwX2xpbmVhcl8xNzI1OV8yMzY2NzQiIHgxPSItODk0LjI0OSIgeTE9IjE
+Cjwvc3ZnPgo=",
                "Category": "PAGE_FOOTER_BACKGROUND",
                "ColorMode": "DARK",
                "Extension": "SVG"
            }
        ],
        "CreationDate": 1732138490.642,
        "LastModifiedDate": 1732140420.301,
        "ManagedLoginBrandingId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Settings": {
            "categories": {
                "auth": {
                    "authMethodOrder": [
                        [
                            {
                                "display": "BUTTON",
                                "type": "FEDERATED"
                            },
                            {
                                "display": "INPUT",
                                "type": "USERNAME_PASSWORD"
                            }
                        ]
                    ],
                    "federation": {
                        "interfaceStyle": "BUTTON_LIST",
                        "order": [
                        ]
                    }
                },
                "form": {
```

```
                "displayGraphics": true,
                "instructions": {
                    "enabled": false
                },
                "languageSelector": {
                    "enabled": false
                },
                "location": {
                    "horizontal": "CENTER",
                    "vertical": "CENTER"
                },
                "sessionTimerDisplay": "NONE"
            },
            "global": {
                "colorSchemeMode": "LIGHT",
                "pageFooter": {
                    "enabled": false
                },
                "pageHeader": {
                    "enabled": false
                },
                "spacingDensity": "REGULAR"
            },
            "signUp": {
                "acceptanceElements": [
                    {
                        "enforcement": "NONE",
                        "textKey": "en"
                    }
                ]
            }
        },
        "componentClasses": {
            "buttons": {
                "borderRadius": 8.0
            },
            "divider": {
                "darkMode": {
                    "borderColor": "232b37ff"
                },
                "lightMode": {
                    "borderColor": "ebebf0ff"
                }
            },
```

```
                "dropDown": {
                    "borderRadius": 8.0,
                    "darkMode": {
                        "defaults": {
                            "itemBackgroundColor": "192534ff"
                        },
                        "hover": {
                            "itemBackgroundColor": "081120ff",
                            "itemBorderColor": "5f6b7aff",
                            "itemTextColor": "e9ebedff"
                        },
                        "match": {
                            "itemBackgroundColor": "d1d5dbff",
                            "itemTextColor": "89bdeeff"
                        }
                    },
                    "lightMode": {
                        "defaults": {
                            "itemBackgroundColor": "ffffffff"
                        },
                        "hover": {
                            "itemBackgroundColor": "f4f4f4ff",
                            "itemBorderColor": "7d8998ff",
                            "itemTextColor": "000716ff"
                        },
                        "match": {
                            "itemBackgroundColor": "414d5cff",
                            "itemTextColor": "0972d3ff"
                        }
                    }
                },
                "focusState": {
                    "darkMode": {
                        "borderColor": "539fe5ff"
                    },
                    "lightMode": {
                        "borderColor": "0972d3ff"
                    }
                },
                "idpButtons": {
                    "icons": {
                        "enabled": true
                    }
                },
```

```
                "input": {
                    "borderRadius": 8.0,
                    "darkMode": {
                        "defaults": {
                            "backgroundColor": "0f1b2aff",
                            "borderColor": "5f6b7aff"
                        },
                        "placeholderColor": "8d99a8ff"
                    },
                    "lightMode": {
                        "defaults": {
                            "backgroundColor": "ffffffff",
                            "borderColor": "7d8998ff"
                        },
                        "placeholderColor": "5f6b7aff"
                    }
                },
                "inputDescription": {
                    "darkMode": {
                        "textColor": "8d99a8ff"
                    },
                    "lightMode": {
                        "textColor": "5f6b7aff"
                    }
                },
                "inputLabel": {
                    "darkMode": {
                        "textColor": "d1d5dbff"
                    },
                    "lightMode": {
                        "textColor": "000716ff"
                    }
                },
                "link": {
                    "darkMode": {
                        "defaults": {
                            "textColor": "539fe5ff"
                        },
                        "hover": {
                            "textColor": "89bdeeff"
                        }
                    },
                    "lightMode": {
                        "defaults": {
```

```
                        "textColor": "0972d3ff"
                    },
                    "hover": {
                        "textColor": "033160ff"
                    }
                }
            },
            "optionControls": {
                "darkMode": {
                    "defaults": {
                        "backgroundColor": "0f1b2aff",
                        "borderColor": "7d8998ff"
                    },
                    "selected": {
                        "backgroundColor": "539fe5ff",
                        "foregroundColor": "000716ff"
                    }
                },
                "lightMode": {
                    "defaults": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "7d8998ff"
                    },
                    "selected": {
                        "backgroundColor": "0972d3ff",
                        "foregroundColor": "ffffffff"
                    }
                }
            },
            "statusIndicator": {
                "darkMode": {
                    "error": {
                        "backgroundColor": "1a0000ff",
                        "borderColor": "eb6f6fff",
                        "indicatorColor": "eb6f6fff"
                    },
                    "pending": {
                        "indicatorColor": "AAAAAAAA"
                    },
                    "success": {
                        "backgroundColor": "001a02ff",
                        "borderColor": "29ad32ff",
                        "indicatorColor": "29ad32ff"
                    },
```

```
                    "warning": {
                        "backgroundColor": "1d1906ff",
                        "borderColor": "e0ca57ff",
                        "indicatorColor": "e0ca57ff"
                    }
                },
                "lightMode": {
                    "error": {
                        "backgroundColor": "fff7f7ff",
                        "borderColor": "d91515ff",
                        "indicatorColor": "d91515ff"
                    },
                    "pending": {
                        "indicatorColor": "AAAAAAAA"
                    },
                    "success": {
                        "backgroundColor": "f2fcf3ff",
                        "borderColor": "037f0cff",
                        "indicatorColor": "037f0cff"
                    },
                    "warning": {
                        "backgroundColor": "fffce9ff",
                        "borderColor": "8d6605ff",
                        "indicatorColor": "8d6605ff"
                    }
                }
            }
        },
        "components": {
            "alert": {
                "borderRadius": 12.0,
                "darkMode": {
                    "error": {
                        "backgroundColor": "1a0000ff",
                        "borderColor": "eb6f6fff"
                    }
                },
                "lightMode": {
                    "error": {
                        "backgroundColor": "fff7f7ff",
                        "borderColor": "d91515ff"
                    }
                }
            },
```

```
                "favicon": {
                    "enabledTypes": [
                        "ICO",
                        "SVG"
                    ]
                },
                "form": {
                    "backgroundImage": {
                        "enabled": false
                    },
                    "borderRadius": 8.0,
                    "darkMode": {
                        "backgroundColor": "0f1b2aff",
                        "borderColor": "424650ff"
                    },
                    "lightMode": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "c6c6cdff"
                    },
                    "logo": {
                        "enabled": false,
                        "formInclusion": "IN",
                        "location": "CENTER",
                        "position": "TOP"
                    }
                },
                "idpButton": {
                    "custom": {
                    },
                    "standard": {
                        "darkMode": {
                            "active": {
                                "backgroundColor": "354150ff",
                                "borderColor": "89bdeeff",
                                "textColor": "89bdeeff"
                            },
                            "defaults": {
                                "backgroundColor": "0f1b2aff",
                                "borderColor": "c6c6cdff",
                                "textColor": "c6c6cdff"
                            },
                            "hover": {
                                "backgroundColor": "192534ff",
                                "borderColor": "89bdeeff",
```

```
                        "textColor": "89bdeeff"
                    }
                },
                "lightMode": {
                    "active": {
                        "backgroundColor": "d3e7f9ff",
                        "borderColor": "033160ff",
                        "textColor": "033160ff"
                    },
                    "defaults": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "424650ff",
                        "textColor": "424650ff"
                    },
                    "hover": {
                        "backgroundColor": "f2f8fdff",
                        "borderColor": "033160ff",
                        "textColor": "033160ff"
                    }
                }
            }
        },
        "pageBackground": {
            "darkMode": {
                "color": "0f1b2aff"
            },
            "image": {
                "enabled": true
            },
            "lightMode": {
                "color": "ffffffff"
            }
        },
        "pageFooter": {
            "backgroundImage": {
                "enabled": false
            },
            "darkMode": {
                "background": {
                    "color": "0f141aff"
                },
                "borderColor": "424650ff"
            },
            "lightMode": {
```

```
                    "background": {
                        "color": "fafafaff"
                    },
                    "borderColor": "d5dbdbff"
                },
                "logo": {
                    "enabled": false,
                    "location": "START"
                }
            },
            "pageHeader": {
                "backgroundImage": {
                    "enabled": false
                },
                "darkMode": {
                    "background": {
                        "color": "0f141aff"
                    },
                    "borderColor": "424650ff"
                },
                "lightMode": {
                    "background": {
                        "color": "fafafaff"
                    },
                    "borderColor": "d5dbdbff"
                },
                "logo": {
                    "enabled": false,
                    "location": "START"
                }
            },
            "pageText": {
                "darkMode": {
                    "bodyColor": "b6bec9ff",
                    "descriptionColor": "b6bec9ff",
                    "headingColor": "d1d5dbff"
                },
                "lightMode": {
                    "bodyColor": "414d5cff",
                    "descriptionColor": "414d5cff",
                    "headingColor": "000716ff"
                }
            },
            "phoneNumberSelector": {
```

```
                    "displayType": "TEXT"
                },
                "primaryButton": {
                    "darkMode": {
                        "active": {
                            "backgroundColor": "539fe5ff",
                            "textColor": "000716ff"
                        },
                        "defaults": {
                            "backgroundColor": "539fe5ff",
                            "textColor": "000716ff"
                        },
                        "disabled": {
                            "backgroundColor": "ffffffff",
                            "borderColor": "ffffffff"
                        },
                        "hover": {
                            "backgroundColor": "89bdeeff",
                            "textColor": "000716ff"
                        }
                    },
                    "lightMode": {
                        "active": {
                            "backgroundColor": "033160ff",
                            "textColor": "ffffffff"
                        },
                        "defaults": {
                            "backgroundColor": "0972d3ff",
                            "textColor": "ffffffff"
                        },
                        "disabled": {
                            "backgroundColor": "ffffffff",
                            "borderColor": "ffffffff"
                        },
                        "hover": {
                            "backgroundColor": "033160ff",
                            "textColor": "ffffffff"
                        }
                    }
                },
                "secondaryButton": {
                    "darkMode": {
                        "active": {
                            "backgroundColor": "354150ff",
```

```
                            "borderColor": "89bdeeff",
                            "textColor": "89bdeeff"
                        },
                        "defaults": {
                            "backgroundColor": "0f1b2aff",
                            "borderColor": "539fe5ff",
                            "textColor": "539fe5ff"
                        },
                        "hover": {
                            "backgroundColor": "192534ff",
                            "borderColor": "89bdeeff",
                            "textColor": "89bdeeff"
                        }
                    },
                    "lightMode": {
                        "active": {
                            "backgroundColor": "d3e7f9ff",
                            "borderColor": "033160ff",
                            "textColor": "033160ff"
                        },
                        "defaults": {
                            "backgroundColor": "ffffffff",
                            "borderColor": "0972d3ff",
                            "textColor": "0972d3ff"
                        },
                        "hover": {
                            "backgroundColor": "f2f8fdff",
                            "borderColor": "033160ff",
                            "textColor": "033160ff"
                        }
                    }
                }
            },
            "UseCognitoProvidedValues": false,
            "UserPoolId": "ca-central-1_EXAMPLE"
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的將品牌套用至受管登入頁面。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateManagedLoginBranding。

update-resource-server

以下程式碼範例顯示如何使用 update-resource-server。

AWS CLI

更新資源伺服器

此範例會更新資源伺服器天氣。它會新增新的範圍。

命令：

```
aws cognito-idp update-resource-server --user-pool-id us-west-2_aaaaaaaaa
 --identifier weather.example.com --name Weather --scopes
 ScopeName=NewScope,ScopeDescription="New scope description"
```

輸出：

```
{
  "ResourceServer": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "Identifier": "weather.example.com",
      "Name": "Happy",
      "Scopes": [
          {
              "ScopeName": "NewScope",
              "ScopeDescription": "New scope description"
          }
      ]
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateResourceServer。

update-user-attributes

以下程式碼範例顯示如何使用 update-user-attributes。

AWS CLI

更新使用者屬性

此範例會更新使用者屬性 "nickname"。

命令：

```
aws cognito-idp update-user-attributes --access-token ACCESS_TOKEN --user-attributes
 Name="nickname",Value="Dan"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateUserAttributes。

update-user-pool-client

以下程式碼範例顯示如何使用 update-user-pool-client。

AWS CLI

更新應用程式用戶端

下列update-user-pool-client範例會更新所請求應用程式用戶端的組態。

```
aws cognito-idp update-user-pool-client \
    --user-pool-id us-west-2_EXAMPLE \
    --client-id 1example23456789 \
    --client-name my-test-app \
    --refresh-token-validity 30 \
    --access-token-validity 60 \
    --id-token-validity 60 \
    --token-validity-units AccessToken=minutes,IdToken=minutes,RefreshToken=days \
    --read-
attributes "address" "birthdate" "email" "email_verified" "family_name" "gender" "locale" "m
 \
    --write-
attributes "address" "birthdate" "email" "family_name" "gender" "locale" "middle_name" "name
 \
    --explicit-auth-
flows "ALLOW_ADMIN_USER_PASSWORD_AUTH" "ALLOW_CUSTOM_AUTH" "ALLOW_REFRESH_TOKEN_AUTH" "ALLOW
 \
    --supported-identity-providers "MySAML" "COGNITO" "Google" \
    --callback-urls "https://www.example.com" "https://app2.example.com" \
    --logout-urls "https://auth.example.com/login?
client_id=1example23456789&response_type=code&redirect_uri=https%3A%2F
%2Fwww.example.com" "https://example.com/logout" \
    --default-redirect-uri "https://www.example.com" \
```

```
    --allowed-o-auth-flows "code" "implicit" \
    --allowed-o-auth-scopes "openid" "profile" "aws.cognito.signin.user.admin" \
    --allowed-o-auth-flows-user-pool-client \
    --prevent-user-existence-errors ENABLED \
    --enable-token-revocation \
    --no-enable-propagate-additional-user-context-data \
    --auth-session-validity 3
```

輸出：

```
{
    "UserPoolClient": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ClientName": "my-test-app",
        "ClientId": "1example23456789",
        "LastModifiedDate": "2025-01-31T14:40:12.498000-08:00",
        "CreationDate": "2023-09-13T16:26:34.408000-07:00",
        "RefreshTokenValidity": 30,
        "AccessTokenValidity": 60,
        "IdTokenValidity": 60,
        "TokenValidityUnits": {
            "AccessToken": "minutes",
            "IdToken": "minutes",
            "RefreshToken": "days"
        },
        "ReadAttributes": [
            "website",
            "zoneinfo",
            "address",
            "birthdate",
            "email_verified",
            "gender",
            "profile",
            "phone_number_verified",
            "preferred_username",
            "locale",
            "middle_name",
            "picture",
            "updated_at",
            "name",
            "nickname",
            "phone_number",
            "family_name",
```

```
            "email"
        ],
        "WriteAttributes": [
            "website",
            "zoneinfo",
            "address",
            "birthdate",
            "gender",
            "profile",
            "preferred_username",
            "locale",
            "middle_name",
            "picture",
            "updated_at",
            "name",
            "nickname",
            "phone_number",
            "family_name",
            "email"
        ],
        "ExplicitAuthFlows": [
            "ALLOW_CUSTOM_AUTH",
            "ALLOW_USER_PASSWORD_AUTH",
            "ALLOW_ADMIN_USER_PASSWORD_AUTH",
            "ALLOW_USER_SRP_AUTH",
            "ALLOW_REFRESH_TOKEN_AUTH"
        ],
        "SupportedIdentityProviders": [
            "Google",
            "COGNITO",
            "MySAML"
        ],
        "CallbackURLs": [
            "https://www.example.com",
            "https://app2.example.com"
        ],
        "LogoutURLs": [
            "https://example.com/logout",
            "https://auth.example.com/login?
client_id=1example23456789&response_type=code&redirect_uri=https%3A%2F
%2Fwww.example.com"
        ],
        "DefaultRedirectURI": "https://www.example.com",
        "AllowedOAuthFlows": [
```

```
            "implicit",
            "code"
        ],
        "AllowedOAuthScopes": [
            "aws.cognito.signin.user.admin",
            "openid",
            "profile"
        ],
        "AllowedOAuthFlowsUserPoolClient": true,
        "PreventUserExistenceErrors": "ENABLED",
        "EnableTokenRevocation": true,
        "EnablePropagateAdditionalUserContextData": false,
        "AuthSessionValidity": 3
    }
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的應用程式特定設定與應用程式用戶端。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateUserPoolClient。

update-user-pool-domain

以下程式碼範例顯示如何使用 update-user-pool-domain。

AWS CLI

更新自訂網域

下列update-user-pool-domain範例會設定所請求使用者集區之自訂網域的品牌版本和憑證。

```
aws cognito-idp update-user-pool-domain \
    --user-pool-id ca-central-1_EXAMPLE \
    --domain auth.example.com \
    --managed-login-version 2 \
    --custom-domain-config CertificateArn=arn:aws:acm:us-
east-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "CloudFrontDomain": "example.cloudfront.net",
    "ManagedLoginVersion": 2
```

```
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的受管登入和設定網域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateUserPoolDomain。

update-user-pool

以下程式碼範例顯示如何使用 update-user-pool。

AWS CLI

更新使用者集區

下列update-user-pool範例會使用每個可用組態選項的範例語法來修改使用者集區。若要更新使用者集區，您必須指定所有先前設定的選項，否則它們會重設為預設值。

```
aws cognito-idp update-user-pool --user-pool-id us-west-2_EXAMPLE \
    --policies PasswordPolicy=
\{MinimumLength=6,RequireUppercase=true,RequireLowercase=true,RequireNumbers=true,RequireSym
 \
    --deletion-protection ACTIVE \
    --lambda-config PreSignUp="arn:aws:lambda:us-
west-2:123456789012:function:cognito-test-presignup-
function",PreTokenGeneration="arn:aws:lambda:us-
west-2:123456789012:function:cognito-test-pretoken-function" \
    --auto-verified-attributes "phone_number" "email" \
    --verification-message-template \{\"SmsMessage\":\""Your code is
 {####}"\",\"EmailMessage\":\""Your code is {####}"\",\"EmailSubject\":\""Your
 verification code"\",\"EmailMessageByLink\":\""Click {##here##} to verify
 your email address."\",\"EmailSubjectByLink\":\""Your verification link"\",
\"DefaultEmailOption\":\"CONFIRM_WITH_LINK\"\} \
    --sms-authentication-message "Your code is {####}" \
    --user-attribute-update-settings
 AttributesRequireVerificationBeforeUpdate="email","phone_number" \
    --mfa-configuration "OPTIONAL" \
    --device-
configuration ChallengeRequiredOnNewDevice=true,DeviceOnlyRememberedOnUserPrompt=true
 \
    --email-configuration SourceArn="arn:aws:ses:us-
west-2:123456789012:identity/admin@example.com",ReplyToEmailAddress="amdin
+noreply@example.com",EmailSendingAccount=DEVELOPER,From="admin@amazon.com",ConfigurationSet
configuration-set" \
```

```
    --sms-configuration SnsCallerArn="arn:aws:iam::123456789012:role/service-role/
SNS-SMS-Role",ExternalId="12345",SnsRegion="us-west-2" \
    --admin-create-user-config AllowAdminCreateUserOnly=false,InviteMessageTemplate=
\{SMSMessage=\""Welcome {username}. Your confirmation code is
 {####}"\",EmailMessage=\""Welcome {username}. Your confirmation code is
 {####}"\",EmailSubject=\""Welcome to MyMobileGame"\"\} \
    --user-pool-tags "Function"="MyMobileGame","Developers"="Berlin" \
    --admin-create-user-config AllowAdminCreateUserOnly=false,InviteMessageTemplate=
\{SMSMessage=\""Welcome {username}. Your confirmation code is
 {####}"\",EmailMessage=\""Welcome {username}. Your confirmation code is
 {####}"\",EmailSubject=\""Welcome to MyMobileGame"\"\} \
    --user-pool-add-ons AdvancedSecurityMode="AUDIT" \
    --account-recovery-setting RecoveryMechanisms=
\[\{Priority=1,Name="verified_email"\},\{Priority=2,Name="verified_phone_number"\}\]
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的更新使用者集區組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateUserPool。

verify-software-token

以下程式碼範例顯示如何使用 `verify-software-token`。

AWS CLI

確認註冊 TOTP 驗證器

下列`verify-software-token`範例會完成目前使用者的 TOTP 註冊。

```
aws cognito-idp verify-software-token \
    --access-token eyJra456defEXAMPLE \
    --user-code 123456
```

輸出：

```
{
    "Status": "SUCCESS"
}
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的將 MFA 新增至使用者集區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 VerifySoftwareToken。

verify-user-attribute

以下程式碼範例顯示如何使用 verify-user-attribute。

AWS CLI

驗證屬性變更

下列verify-user-attribute範例會驗證目前使用者的電子郵件屬性的變更。

```
aws cognito-idp verify-user-attribute \
    --access-token eyJra456defEXAMPLE \
    --attribute-name email \
    --code 123456
```

如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的設定電子郵件或電話驗證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 VerifyUserAttribute。

使用 的 Amazon Comprehend 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon Comprehend 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

batch-detect-dominant-language

以下程式碼範例顯示如何使用 batch-detect-dominant-language。

AWS CLI

偵測多個輸入文字的主要語言

下列batch-detect-dominant-language範例會分析多個輸入文字，並傳回每個輸入文字的主要語言。每個預測也會輸出預先訓練的模型可信度分數。

```
aws comprehend batch-detect-dominant-language \
    --text-list "Physics is the natural science that involves the study of matter
 and its motion and behavior through space and time, along with related concepts
 such as energy and force."
```

輸出：

```
{
    "ResultList": [
        {
            "Index": 0,
            "Languages": [
                {
                    "LanguageCode": "en",
                    "Score": 0.9986501932144165
                }
            ]
        }
    ],
    "ErrorList": []
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的主要語言。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDetectDominantLanguage。

batch-detect-entities

以下程式碼範例顯示如何使用 batch-detect-entities。

AWS CLI

從多個輸入文字偵測實體

下列batch-detect-entities範例會分析多個輸入文字，並傳回每個輸入文字的具名實體。每個預測也會輸出預先訓練模型的可信度分數。

```
aws comprehend batch-detect-entities \
    --language-code en \
    --text-list "Dear Jane, Your AnyCompany Financial Services LLC credit card account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July 31st." "Please send customer feedback to Sunshine Spa, 123 Main St, Anywhere or to Alice at AnySpa@example.com."
```

輸出：

```
{
    "ResultList": [
        {
            "Index": 0,
            "Entities": [
                {
                    "Score": 0.9985517859458923,
                    "Type": "PERSON",
                    "Text": "Jane",
                    "BeginOffset": 5,
                    "EndOffset": 9
                },
                {
                    "Score": 0.9767839312553406,
                    "Type": "ORGANIZATION",
                    "Text": "AnyCompany Financial Services, LLC",
                    "BeginOffset": 16,
                    "EndOffset": 50
                },
                {
                    "Score": 0.9856694936752319,
                    "Type": "OTHER",
                    "Text": "1111-XXXX-1111-XXXX",
                    "BeginOffset": 71,
                    "EndOffset": 90
                },
                {
                    "Score": 0.9652159810066223,
                    "Type": "QUANTITY",
                    "Text": ".53",
                    "BeginOffset": 116,
                    "EndOffset": 119
                },
                {
```

```
                "Score": 0.9986667037010193,
                "Type": "DATE",
                "Text": "July 31st",
                "BeginOffset": 135,
                "EndOffset": 144
            }
        ]
    },
    {
        "Index": 1,
        "Entities": [
            {
                "Score": 0.720084547996521,
                "Type": "ORGANIZATION",
                "Text": "Sunshine Spa",
                "BeginOffset": 33,
                "EndOffset": 45
            },
            {
                "Score": 0.9865870475769043,
                "Type": "LOCATION",
                "Text": "123 Main St",
                "BeginOffset": 47,
                "EndOffset": 58
            },
            {
                "Score": 0.5895616412162781,
                "Type": "LOCATION",
                "Text": "Anywhere",
                "BeginOffset": 60,
                "EndOffset": 68
            },
            {
                "Score": 0.6809214353561401,
                "Type": "PERSON",
                "Text": "Alice",
                "BeginOffset": 75,
                "EndOffset": 80
            },
            {
                "Score": 0.9979087114334106,
                "Type": "OTHER",
                "Text": "AnySpa@example.com",
                "BeginOffset": 84,
```

```
                    "EndOffset": 99
                }
            ]
        }
    ],
    "ErrorList": []
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的實體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDetectEntities。

batch-detect-key-phrases

以下程式碼範例顯示如何使用 batch-detect-key-phrases。

AWS CLI

偵測多個文字輸入的金鑰片語

下列batch-detect-key-phrases範例會分析多個輸入文字，並傳回每個輸入文字的金鑰名詞片語。也會輸出每個預測的預先訓練模型可信度分數。

```
aws comprehend batch-detect-key-phrases \
    --language-code en \
    --text-list "Hello Zhang Wei, I am John, writing to you about the trip for
 next Saturday." "Dear Jane, Your AnyCompany Financial Services LLC credit card
 account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July
 31st." "Please send customer feedback to Sunshine Spa, 123 Main St, Anywhere or to
 Alice at AnySpa@example.com."
```

輸出：

```
{
    "ResultList": [
        {
            "Index": 0,
            "KeyPhrases": [
                {
                    "Score": 0.99700927734375,
                    "Text": "Zhang Wei",
                    "BeginOffset": 6,
                    "EndOffset": 15
```

```
                },
                {
                    "Score": 0.9929308891296387,
                    "Text": "John",
                    "BeginOffset": 22,
                    "EndOffset": 26
                },
                {
                    "Score": 0.9997230172157288,
                    "Text": "the trip",
                    "BeginOffset": 49,
                    "EndOffset": 57
                },
                {
                    "Score": 0.9999470114707947,
                    "Text": "next Saturday",
                    "BeginOffset": 62,
                    "EndOffset": 75
                }
            ]
        },
        {
            "Index": 1,
            "KeyPhrases": [
                {
                    "Score": 0.8358274102210999,
                    "Text": "Dear Jane",
                    "BeginOffset": 0,
                    "EndOffset": 9
                },
                {
                    "Score": 0.989359974861145,
                    "Text": "Your AnyCompany Financial Services",
                    "BeginOffset": 11,
                    "EndOffset": 45
                },
                {
                    "Score": 0.8812323808670044,
                    "Text": "LLC credit card account 1111-XXXX-1111-XXXX",
                    "BeginOffset": 47,
                    "EndOffset": 90
                },
                {
                    "Score": 0.9999381899833679,
```

```
                    "Text": "a minimum payment",
                    "BeginOffset": 95,
                    "EndOffset": 112
                },
                {
                    "Score": 0.9997439980506897,
                    "Text": ".53",
                    "BeginOffset": 116,
                    "EndOffset": 119
                },
                {
                    "Score": 0.996875524520874,
                    "Text": "July 31st",
                    "BeginOffset": 135,
                    "EndOffset": 144
                }
            ]
        },
        {
            "Index": 2,
            "KeyPhrases": [
                {
                    "Score": 0.9990295767784119,
                    "Text": "customer feedback",
                    "BeginOffset": 12,
                    "EndOffset": 29
                },
                {
                    "Score": 0.9994127750396729,
                    "Text": "Sunshine Spa",
                    "BeginOffset": 33,
                    "EndOffset": 45
                },
                {
                    "Score": 0.9892991185188293,
                    "Text": "123 Main St",
                    "BeginOffset": 47,
                    "EndOffset": 58
                },
                {
                    "Score": 0.9969810843467712,
                    "Text": "Alice",
                    "BeginOffset": 75,
                    "EndOffset": 80
```

```
                },
                {
                    "Score": 0.9703696370124817,
                    "Text": "AnySpa@example.com",
                    "BeginOffset": 84,
                    "EndOffset": 99
                }
            ]
        }
    ],
    "ErrorList": []
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的關鍵詞。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDetectKeyPhrases。

batch-detect-sentiment

以下程式碼範例顯示如何使用 batch-detect-sentiment。

AWS CLI

偵測多個輸入文字的普遍情緒

下列batch-detect-sentiment範例會分析多個輸入文字，並傳回普遍的情緒 (POSITIVE、MIXED、 NEUTRAL或 NEGATIVE)。

```
aws comprehend batch-detect-sentiment \
    --text-list "That movie was very boring, I can't believe it was over four hours long." "It is a beautiful day for hiking today." "My meal was okay, I'm excited to try other restaurants." \
    --language-code en
```

輸出：

```
{
    "ResultList": [
        {
            "Index": 0,
            "Sentiment": "NEGATIVE",
            "SentimentScore": {
                "Positive": 0.00011316669406369328,
```

```
                "Negative": 0.9995445609092712,
                "Neutral": 0.00014722718333359808,
                "Mixed": 0.00019498742767609656
            }
        },
        {
            "Index": 1,
            "Sentiment": "POSITIVE",
            "SentimentScore": {
                "Positive": 0.9981263279914856,
                "Negative": 0.00015240783977787942,
                "Neutral": 0.0013876151060685515,
                "Mixed": 0.00033366199932061136
            }
        },
        {
            "Index": 2,
            "Sentiment": "MIXED",
            "SentimentScore": {
                "Positive": 0.15930435061454773,
                "Negative": 0.11471917480230331,
                "Neutral": 0.26897063851356506,
                "Mixed": 0.45700588822364807
            }
        }
    ],
    "ErrorList": []
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的情緒。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDetectSentiment。

batch-detect-syntax

以下程式碼範例顯示如何使用 batch-detect-syntax。

AWS CLI

檢查多個輸入文字中單字的語法和語音部分

下列batch-detect-syntax範例會分析多個輸入文字的語法，並傳回語音的不同部分。每個預測也會輸出預先訓練模型的可信度分數。

```
aws comprehend batch-detect-syntax \
    --text-list "It is a beautiful day." "Can you please pass the salt?" "Please pay
 the bill before the 31st." \
    --language-code en
```

輸出：

```
{
    "ResultList": [
        {
            "Index": 0,
            "SyntaxTokens": [
                {
                    "TokenId": 1,
                    "Text": "It",
                    "BeginOffset": 0,
                    "EndOffset": 2,
                    "PartOfSpeech": {
                        "Tag": "PRON",
                        "Score": 0.9999740719795227
                    }
                },
                {
                    "TokenId": 2,
                    "Text": "is",
                    "BeginOffset": 3,
                    "EndOffset": 5,
                    "PartOfSpeech": {
                        "Tag": "VERB",
                        "Score": 0.999937117099762
                    }
                },
                {
                    "TokenId": 3,
                    "Text": "a",
                    "BeginOffset": 6,
                    "EndOffset": 7,
                    "PartOfSpeech": {
                        "Tag": "DET",
                        "Score": 0.9999926686286926
                    }
                },
                {
```

```
                    "TokenId": 4,
                    "Text": "beautiful",
                    "BeginOffset": 8,
                    "EndOffset": 17,
                    "PartOfSpeech": {
                        "Tag": "ADJ",
                        "Score": 0.9987891912460327
                    }
                },
                {
                    "TokenId": 5,
                    "Text": "day",
                    "BeginOffset": 18,
                    "EndOffset": 21,
                    "PartOfSpeech": {
                        "Tag": "NOUN",
                        "Score": 0.9999778866767883
                    }
                },
                {
                    "TokenId": 6,
                    "Text": ".",
                    "BeginOffset": 21,
                    "EndOffset": 22,
                    "PartOfSpeech": {
                        "Tag": "PUNCT",
                        "Score": 0.9999974966049194
                    }
                }
            ]
        },
        {
            "Index": 1,
            "SyntaxTokens": [
                {
                    "TokenId": 1,
                    "Text": "Can",
                    "BeginOffset": 0,
                    "EndOffset": 3,
                    "PartOfSpeech": {
                        "Tag": "AUX",
                        "Score": 0.9999770522117615
                    }
                },
```

```
                {
                    "TokenId": 2,
                    "Text": "you",
                    "BeginOffset": 4,
                    "EndOffset": 7,
                    "PartOfSpeech": {
                        "Tag": "PRON",
                        "Score": 0.9999986886978149
                    }
                },
                {
                    "TokenId": 3,
                    "Text": "please",
                    "BeginOffset": 8,
                    "EndOffset": 14,
                    "PartOfSpeech": {
                        "Tag": "INTJ",
                        "Score": 0.9681622385978699
                    }
                },
                {
                    "TokenId": 4,
                    "Text": "pass",
                    "BeginOffset": 15,
                    "EndOffset": 19,
                    "PartOfSpeech": {
                        "Tag": "VERB",
                        "Score": 0.9999874830245972
                    }
                },
                {
                    "TokenId": 5,
                    "Text": "the",
                    "BeginOffset": 20,
                    "EndOffset": 23,
                    "PartOfSpeech": {
                        "Tag": "DET",
                        "Score": 0.9999827146530151
                    }
                },
                {
                    "TokenId": 6,
                    "Text": "salt",
                    "BeginOffset": 24,
```

```
                    "EndOffset": 28,
                    "PartOfSpeech": {
                        "Tag": "NOUN",
                        "Score": 0.9995040893554688
                    }
                },
                {
                    "TokenId": 7,
                    "Text": "?",
                    "BeginOffset": 28,
                    "EndOffset": 29,
                    "PartOfSpeech": {
                        "Tag": "PUNCT",
                        "Score": 0.999998152256012
                    }
                }
            ]
        },
        {
            "Index": 2,
            "SyntaxTokens": [
                {
                    "TokenId": 1,
                    "Text": "Please",
                    "BeginOffset": 0,
                    "EndOffset": 6,
                    "PartOfSpeech": {
                        "Tag": "INTJ",
                        "Score": 0.9997857809066772
                    }
                },
                {
                    "TokenId": 2,
                    "Text": "pay",
                    "BeginOffset": 7,
                    "EndOffset": 10,
                    "PartOfSpeech": {
                        "Tag": "VERB",
                        "Score": 0.9999252557754517
                    }
                },
                {
                    "TokenId": 3,
                    "Text": "the",
```

```
                    "BeginOffset": 11,
                    "EndOffset": 14,
                    "PartOfSpeech": {
                        "Tag": "DET",
                        "Score": 0.9999842643737793
                    }
                },
                {
                    "TokenId": 4,
                    "Text": "bill",
                    "BeginOffset": 15,
                    "EndOffset": 19,
                    "PartOfSpeech": {
                        "Tag": "NOUN",
                        "Score": 0.9999588131904602
                    }
                },
                {
                    "TokenId": 5,
                    "Text": "before",
                    "BeginOffset": 20,
                    "EndOffset": 26,
                    "PartOfSpeech": {
                        "Tag": "ADP",
                        "Score": 0.9958304762840271
                    }
                },
                {
                    "TokenId": 6,
                    "Text": "the",
                    "BeginOffset": 27,
                    "EndOffset": 30,
                    "PartOfSpeech": {
                        "Tag": "DET",
                        "Score": 0.9999947547912598
                    }
                },
                {
                    "TokenId": 7,
                    "Text": "31st",
                    "BeginOffset": 31,
                    "EndOffset": 35,
                    "PartOfSpeech": {
                        "Tag": "NOUN",
```

```
                    "Score": 0.9924124479293823
                }
            },
            {
                "TokenId": 8,
                "Text": ".",
                "BeginOffset": 35,
                "EndOffset": 36,
                "PartOfSpeech": {
                    "Tag": "PUNCT",
                    "Score": 0.9999955892562866
                }
            }
        ]
    }
],
"ErrorList": []
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的語法分析。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDetectSyntax。

batch-detect-targeted-sentiment

以下程式碼範例顯示如何使用 batch-detect-targeted-sentiment。

AWS CLI

針對多個輸入文字偵測情緒和每個具名實體

下列batch-detect-targeted-sentiment範例會分析多個輸入文字，並傳回具名實體以及連接到每個實體的現行情緒。每個預測也會輸出預先訓練模型的可信度分數。

```
aws comprehend batch-detect-targeted-sentiment \
    --language-code en \
    --text-list "That movie was really boring, the original was way more
 entertaining" "The trail is extra beautiful today." "My meal was just okay."
```

輸出：

```
{
    "ResultList": [
```

```
        {
            "Index": 0,
            "Entities": [
                {
                    "DescriptiveMentionIndex": [
                        0
                    ],
                    "Mentions": [
                        {
                            "Score": 0.9999009966850281,
                            "GroupScore": 1.0,
                            "Text": "movie",
                            "Type": "MOVIE",
                            "MentionSentiment": {
                                "Sentiment": "NEGATIVE",
                                "SentimentScore": {
                                    "Positive": 0.13887299597263336,
                                    "Negative": 0.8057460188865662,
                                    "Neutral": 0.05525200068950653,
                                    "Mixed": 0.00012799999967683107
                                }
                            },
                            "BeginOffset": 5,
                            "EndOffset": 10
                        }
                    ]
                },
                {
                    "DescriptiveMentionIndex": [
                        0
                    ],
                    "Mentions": [
                        {
                            "Score": 0.9921110272407532,
                            "GroupScore": 1.0,
                            "Text": "original",
                            "Type": "MOVIE",
                            "MentionSentiment": {
                                "Sentiment": "POSITIVE",
                                "SentimentScore": {
                                    "Positive": 0.9999989867210388,
                                    "Negative": 9.999999974752427e-07,
                                    "Neutral": 0.0,
                                    "Mixed": 0.0
```

```
                            }
                        },
                        "BeginOffset": 34,
                        "EndOffset": 42
                    }
                ]
            }
        ]
    },
    {
        "Index": 1,
        "Entities": [
            {
                "DescriptiveMentionIndex": [
                    0
                ],
                "Mentions": [
                    {
                        "Score": 0.7545599937438965,
                        "GroupScore": 1.0,
                        "Text": "trail",
                        "Type": "OTHER",
                        "MentionSentiment": {
                            "Sentiment": "POSITIVE",
                            "SentimentScore": {
                                "Positive": 1.0,
                                "Negative": 0.0,
                                "Neutral": 0.0,
                                "Mixed": 0.0
                            }
                        },
                        "BeginOffset": 4,
                        "EndOffset": 9
                    }
                ]
            },
            {
                "DescriptiveMentionIndex": [
                    0
                ],
                "Mentions": [
                    {
                        "Score": 0.9999960064888,
                        "GroupScore": 1.0,
```

```
                        "Text": "today",
                        "Type": "DATE",
                        "MentionSentiment": {
                            "Sentiment": "NEUTRAL",
                            "SentimentScore": {
                                "Positive": 9.000000318337698e-06,
                                "Negative": 1.9999999949504854e-06,
                                "Neutral": 0.9999859929084778,
                                "Mixed": 3.999999989900971e-06
                            }
                        },
                        "BeginOffset": 29,
                        "EndOffset": 34
                    }
                ]
            }
        ]
    },
    {
        "Index": 2,
        "Entities": [
            {
                "DescriptiveMentionIndex": [
                    0
                ],
                "Mentions": [
                    {
                        "Score": 0.9999880194664001,
                        "GroupScore": 1.0,
                        "Text": "My",
                        "Type": "PERSON",
                        "MentionSentiment": {
                            "Sentiment": "NEUTRAL",
                            "SentimentScore": {
                                "Positive": 0.0,
                                "Negative": 0.0,
                                "Neutral": 1.0,
                                "Mixed": 0.0
                            }
                        },
                        "BeginOffset": 0,
                        "EndOffset": 2
                    }
                ]
```

```
                },
                {
                    "DescriptiveMentionIndex": [
                        0
                    ],
                    "Mentions": [
                        {
                            "Score": 0.9995260238647461,
                            "GroupScore": 1.0,
                            "Text": "meal",
                            "Type": "OTHER",
                            "MentionSentiment": {
                                "Sentiment": "NEUTRAL",
                                "SentimentScore": {
                                    "Positive": 0.04695599898695946,
                                    "Negative": 0.003226999891921878,
                                    "Neutral": 0.6091709733009338,
                                    "Mixed": 0.34064599871635437
                                }
                            },
                            "BeginOffset": 3,
                            "EndOffset": 7
                        }
                    ]
                }
            ]
        }
    ],
    "ErrorList": []
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的目標情緒。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDetectTargetedSentiment。

`classify-document`

以下程式碼範例顯示如何使用 `classify-document`。

AWS CLI

使用模型特定的端點對文件進行分類

下列classify-document範例會分類具有自訂模型端點的文件。此範例中的模型是在資料集上訓練，其中包含標示為垃圾郵件或非垃圾郵件的 sms 訊息，或是 "ham"。

```
aws comprehend classify-document \
    --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-endpoint/example-classifier-endpoint \
    --text "CONGRATULATIONS! TXT 1235550100 to win $5000"
```

輸出：

```
{
    "Classes": [
        {
            "Name": "spam",
            "Score": 0.9998599290847778
        },
        {
            "Name": "ham",
            "Score": 0.00014001205272506922
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的自訂分類。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ClassifyDocument。

contains-pii-entities

以下程式碼範例顯示如何使用 contains-pii-entities。

AWS CLI

分析是否有 PII 資訊的輸入文字

下列contains-pii-entities範例會分析輸入文字中是否存在個人身分識別資訊 (PII)，並傳回已識別 PII 實體類型的標籤，例如名稱、地址、銀行帳戶號碼或電話號碼。

```
aws comprehend contains-pii-entities \
    --language-code en \
    --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card
```

```
       account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by
July 31st. Based on your autopay settings,
       we will withdraw your payment on the due date from your bank account number
XXXXXX1111 with the routing number XXXXX0000.
       Customer feedback for Sunshine Spa, 100 Main St, Anywhere. Send comments to
Alice at AnySpa@example.com."
```

輸出：

```
{
    "Labels": [
        {
            "Name": "NAME",
            "Score": 1.0
        },
        {
            "Name": "EMAIL",
            "Score": 1.0
        },
        {
            "Name": "BANK_ACCOUNT_NUMBER",
            "Score": 0.9995794296264648
        },
        {
            "Name": "BANK_ROUTING",
            "Score": 0.9173126816749573
        },
        {
            "Name": "CREDIT_DEBIT_NUMBER",
            "Score": 1.0
        }
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的個人身分識別資訊 (PII)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ContainsPiiEntities。

create-dataset

以下程式碼範例顯示如何使用 create-dataset。

AWS CLI

建立飛輪資料集

下列create-dataset範例會建立飛輪的資料集。此資料集將用作--dataset-type標籤指定的其他訓練資料。

```
aws comprehend create-dataset \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-entity \
    --dataset-name example-dataset \
    --dataset-type "TRAIN" \
    --input-data-config file://inputConfig.json
```

file://inputConfig.json 的內容：

```
{
    "DataFormat": "COMPREHEND_CSV",
    "DocumentClassifierInputDataConfig": {
        "S3Uri": "s3://amzn-s3-demo-bucket/training-data.csv"
    }
}
```

輸出：

```
{
    "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-entity/dataset/example-dataset"
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的飛輪概觀。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDataset。

create-document-classifier

以下程式碼範例顯示如何使用 create-document-classifier。

AWS CLI

建立文件分類器以分類文件

下列create-document-classifier範例會開始文件分類器模型的訓練程序。訓練資料檔案training.csv位於 --input-data-config標籤。 training.csv 是兩欄文件，其中標籤 或分類在第一欄提供，文件在第二欄提供。

```
aws comprehend create-document-classifier \
    --document-classifier-name example-classifier \
    --data-access-arn arn:aws:comprehend:us-west-2:111122223333:pii-entities-
detection-job/123456abcdeb0e11022f22a11EXAMPLE \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --language-code en
```

輸出：

```
{
    "DocumentClassifierArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier"
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的自訂分類。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDocumentClassifier。

create-endpoint

以下程式碼範例顯示如何使用 create-endpoint。

AWS CLI

為自訂模型建立端點

下列create-endpoint範例會為先前訓練的自訂模型建立同步推論的端點。

```
aws comprehend create-endpoint \
    --endpoint-name example-classifier-endpoint-1 \
    --model-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier \
    --desired-inference-units 1
```

輸出：

```
{
```

```
    "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier-
endpoint/example-classifier-endpoint-1"
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的管理 Amazon Comprehend 端點。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateEndpoint。

create-entity-recognizer

以下程式碼範例顯示如何使用 create-entity-recognizer。

AWS CLI

建立自訂實體辨識器

下列create-entity-recognizer範例會開始自訂實體辨識器模型的訓練程序。此範例使用包含訓練文件、 raw_text.csv和 CSV 實體清單的 CSV 檔案entity_list.csv來訓練模型。entity-list.csv包含下列資料欄：文字和類型。

```
aws comprehend create-entity-recognizer \
    --recognizer-name example-entity-recognizer
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
    --input-data-config "EntityTypes=[{Type=DEVICE}],Documents={S3Uri=s3://amzn-s3-
demo-bucket/trainingdata/raw_text.csv},EntityList={S3Uri=s3://amzn-s3-demo-bucket/
trainingdata/entity_list.csv}"
    --language-code en
```

輸出：

```
{
    "EntityRecognizerArn": "arn:aws:comprehend:us-west-2:111122223333:example-
entity-recognizer/entityrecognizer1"
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的自訂實體辨識。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateEntityRecognizer。

create-flywheel

以下程式碼範例顯示如何使用 create-flywheel。

AWS CLI

建立飛輪

下列create-flywheel範例會建立飛輪，以協調文件分類或實體辨識模型的持續訓練。此範例中的飛輪是用來管理 --active-model-arn標籤指定的現有訓練模型。飛輪建立時，會在 --input-data-lake標籤建立資料湖。

```
aws comprehend create-flywheel \
    --flywheel-name example-flywheel \
    --active-model-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-model/version/1 \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role \
    --data-lake-s3-uri "s3://amzn-s3-demo-bucket"
```

輸出：

```
{
    "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel"
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的飛輪概觀。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFlywheel。

delete-document-classifier

以下程式碼範例顯示如何使用 delete-document-classifier。

AWS CLI

刪除自訂文件分類器

下列delete-document-classifier範例會刪除自訂文件分類器模型。

```
aws comprehend delete-document-classifier \
```

```
    --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier-1
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的管理 Amazon Comprehend 端點。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDocumentClassifier。

delete-endpoint

以下程式碼範例顯示如何使用 delete-endpoint。

AWS CLI

刪除自訂模型的端點

下列delete-endpoint範例會刪除模型特定的端點。必須刪除所有端點，才能刪除模型。

```
aws comprehend delete-endpoint \
    --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-
endpoint/example-classifier-endpoint-1
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的管理 Amazon Comprehend 端點。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteEndpoint。

delete-entity-recognizer

以下程式碼範例顯示如何使用 delete-entity-recognizer。

AWS CLI

刪除自訂實體辨識器模型

下列delete-entity-recognizer範例會刪除自訂實體辨識器模型。

```
aws comprehend delete-entity-recognizer \
```

```
    --entity-recognizer-arn arn:aws:comprehend:us-west-2:111122223333:entity-
recognizer/example-entity-recognizer-1
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的管理 Amazon Comprehend 端點。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteEntityRecognizer。

delete-flywheel

以下程式碼範例顯示如何使用 delete-flywheel。

AWS CLI

刪除飛輪

下列delete-flywheel範例會刪除飛輪。不會刪除與飛輪相關聯的資料湖或模型。

```
aws comprehend delete-flywheel \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-
flywheel-1
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的飛輪概觀。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFlywheel。

delete-resource-policy

以下程式碼範例顯示如何使用 delete-resource-policy。

AWS CLI

刪除以資源為基礎的政策

下列delete-resource-policy範例會從 Amazon Comprehend 資源刪除資源型政策。

```
aws comprehend delete-resource-policy \
```

```
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier-1/version/1
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的在AWS 帳戶之間複製自訂模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteResourcePolicy。

describe-dataset

以下程式碼範例顯示如何使用 describe-dataset。

AWS CLI

描述飛輪資料集

下列describe-dataset範例會取得飛輪資料集的屬性。

```
aws comprehend describe-dataset \
    --dataset-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-
entity/dataset/example-dataset
```

輸出：

```
{
    "DatasetProperties": {
        "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-
entity/dataset/example-dataset",
        "DatasetName": "example-dataset",
        "DatasetType": "TRAIN",
        "DatasetS3Uri": "s3://amzn-s3-demo-bucket/flywheel-entity/
schemaVersion=1/12345678A123456Z/datasets/example-dataset/20230616T203710Z/",
        "Status": "CREATING",
        "CreationTime": "2023-06-16T20:37:10.400000+00:00"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的飛輪概觀。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDataset。

describe-document-classification-job

以下程式碼範例顯示如何使用 describe-document-classification-job。

AWS CLI

描述文件分類任務

下列describe-document-classification-job範例會取得非同步文件分類任務的屬性。

```
aws comprehend describe-document-classification-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

輸出：

```
{
    "DocumentClassificationJobProperties": {
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classification-job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "exampleclassificationjob",
        "JobStatus": "COMPLETED",
        "SubmitTime": "2023-06-14T17:09:51.788000+00:00",
        "EndTime": "2023-06-14T17:15:58.582000+00:00",
        "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/mymodel/version/1",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/jobdata/",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-destination-bucket/testfolder/111122223333-
CLN-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
        },
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的自訂分類。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDocumentClassificationJob。

describe-document-classifier

以下程式碼範例顯示如何使用 describe-document-classifier。

AWS CLI

描述文件分類器

下列describe-document-classifier範例會取得自訂文件分類器模型的屬性。

```
aws comprehend describe-document-classifier \
    --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier-1
```

輸出：

```
{
    "DocumentClassifierProperties": {
        "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/example-classifier-1",
        "LanguageCode": "en",
        "Status": "TRAINED",
        "SubmitTime": "2023-06-13T19:04:15.735000+00:00",
        "EndTime": "2023-06-13T19:42:31.752000+00:00",
        "TrainingStartTime": "2023-06-13T19:08:20.114000+00:00",
        "TrainingEndTime": "2023-06-13T19:41:35.080000+00:00",
        "InputDataConfig": {
            "DataFormat": "COMPREHEND_CSV",
            "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata"
        },
        "OutputDataConfig": {},
        "ClassifierMetadata": {
            "NumberOfLabels": 3,
            "NumberOfTrainedDocuments": 5016,
            "NumberOfTestDocuments": 557,
            "EvaluationMetrics": {
                "Accuracy": 0.9856,
                "Precision": 0.9919,
                "Recall": 0.9459,
                "F1Score": 0.9673,
                "MicroPrecision": 0.9856,
                "MicroRecall": 0.9856,
                "MicroF1Score": 0.9856,
                "HammingLoss": 0.0144
```

```
            }
        },
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
        "Mode": "MULTI_CLASS"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的建立和管理自訂模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDocumentClassifier。

describe-dominant-language-detection-job

以下程式碼範例顯示如何使用 describe-dominant-language-detection-job。

AWS CLI

描述主要語言偵測偵測任務。

下列describe-dominant-language-detection-job範例會取得非同步主要語言偵測任務的屬性。

```
aws comprehend describe-dominant-language-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

輸出：

```
{
    "DominantLanguageDetectionJobProperties": {
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "languageanalysis1",
        "JobStatus": "IN_PROGRESS",
        "SubmitTime": "2023-06-09T18:10:38.037000+00:00",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-destination-bucket/testfolder/111122223333-
LANGUAGE-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
```

```
        },
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDominantLanguageDetectionJob。

describe-endpoint

以下程式碼範例顯示如何使用 describe-endpoint。

AWS CLI

描述特定端點

下列describe-endpoint範例會取得模型特定端點的屬性。

```
aws comprehend describe-endpoint \
    --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-
endpoint/example-classifier-endpoint
```

輸出：

```
{
    "EndpointProperties": {
        "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier-endpoint/example-classifier-endpoint",
        "Status": "IN_SERVICE",
        "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/
exampleclassifier1",
        "DesiredModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier1",
        "DesiredInferenceUnits": 1,
        "CurrentInferenceUnits": 1,
        "CreationTime": "2023-06-13T20:32:54.526000+00:00",
        "LastModifiedTime": "2023-06-13T20:32:54.526000+00:00"
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的管理 Amazon Comprehend 端點。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEndpoint。

describe-entities-detection-job

以下程式碼範例顯示如何使用 describe-entities-detection-job。

AWS CLI

描述實體偵測任務

下列describe-entities-detection-job範例會取得非同步實體偵測任務的屬性。

```
aws comprehend describe-entities-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

輸出：

```
{
    "EntitiesDetectionJobProperties": {
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "example-entity-detector",
        "JobStatus": "COMPLETED",
        "SubmitTime": "2023-06-08T21:30:15.323000+00:00",
        "EndTime": "2023-06-08T21:40:23.509000+00:00",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/AsyncBatchJobs/",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/thefolder/111122223333-
NER-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::12345678012:role/service-role/
AmazonComprehendServiceRole-example-role"
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEntitiesDetectionJob。

describe-entity-recognizer

以下程式碼範例顯示如何使用 describe-entity-recognizer。

AWS CLI

描述實體辨識器

下列describe-entity-recognizer範例會取得自訂實體辨識器模型的屬性。

```
aws comprehend describe-entity-recognizer \
    entity-recognizer-arn arn:aws:comprehend:us-west-2:111122223333:entity-
recognizer/business-recongizer-1/version/1
```

輸出：

```
{
    "EntityRecognizerProperties": {
        "EntityRecognizerArn": "arn:aws:comprehend:us-west-2:111122223333:entity-
recognizer/business-recongizer-1/version/1",
        "LanguageCode": "en",
        "Status": "TRAINED",
        "SubmitTime": "2023-06-14T20:44:59.631000+00:00",
        "EndTime": "2023-06-14T20:59:19.532000+00:00",
        "TrainingStartTime": "2023-06-14T20:48:52.811000+00:00",
        "TrainingEndTime": "2023-06-14T20:58:11.473000+00:00",
        "InputDataConfig": {
            "DataFormat": "COMPREHEND_CSV",
            "EntityTypes": [
                {
                    "Type": "BUSINESS"
                }
            ],
            "Documents": {
                "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata/dataset/",
                "InputFormat": "ONE_DOC_PER_LINE"
```

```
            },
            "EntityList": {
                "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata/entity.csv"
            }
        },
        "RecognizerMetadata": {
            "NumberOfTrainedDocuments": 1814,
            "NumberOfTestDocuments": 486,
            "EvaluationMetrics": {
                "Precision": 100.0,
                "Recall": 100.0,
                "F1Score": 100.0
            },
            "EntityTypes": [
                {
                    "Type": "BUSINESS",
                    "EvaluationMetrics": {
                        "Precision": 100.0,
                        "Recall": 100.0,
                        "F1Score": 100.0
                    },
                    "NumberOfTrainMentions": 1520
                }
            ]
        },
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
        "VersionName": "1"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的自訂實體辨識。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEntityRecognizer。

describe-events-detection-job

以下程式碼範例顯示如何使用 describe-events-detection-job。

AWS CLI

描述事件偵測任務。

下列describe-events-detection-job範例會取得非同步事件偵測任務的屬性。

```
aws comprehend describe-events-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

輸出：

```
{
    "EventsDetectionJobProperties": {
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:events-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "events_job_1",
        "JobStatus": "IN_PROGRESS",
        "SubmitTime": "2023-06-12T18:45:56.054000+00:00",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/EventsData",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-destination-bucket/testfolder/111122223333-
EVENTS-123456abcdeb0e11022f22a11EXAMPLE/output/"
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
        "TargetEventTypes": [
            "BANKRUPTCY",
            "EMPLOYMENT",
            "CORPORATE_ACQUISITION",
            "CORPORATE_MERGER",
            "INVESTMENT_GENERAL"
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEventsDetectionJob。

describe-flywheel-iteration

以下程式碼範例顯示如何使用 describe-flywheel-iteration。

AWS CLI

描述飛輪反覆運算

下列describe-flywheel-iteration範例會取得飛輪反覆運算的屬性。

```
aws comprehend describe-flywheel-iteration \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-
flywheel \
    --flywheel-iteration-id 20232222AEXAMPLE
```

輸出：

```
{
    "FlywheelIterationProperties": {
        "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-
entity",
        "FlywheelIterationId": "20232222AEXAMPLE",
        "CreationTime": "2023-06-16T21:10:26.385000+00:00",
        "EndTime": "2023-06-16T23:33:16.827000+00:00",
        "Status": "COMPLETED",
        "Message": "FULL_ITERATION: Flywheel iteration performed all functions
 successfully.",
        "EvaluatedModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier/version/1",
        "EvaluatedModelMetrics": {
            "AverageF1Score": 0.7742663922375772,
            "AveragePrecision": 0.8287636394041166,
            "AverageRecall": 0.7427084833645399,
            "AverageAccuracy": 0.8795394154118689
        },
        "TrainedModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier/version/Comprehend-Generated-v1-bb52d585",
        "TrainedModelMetrics": {
            "AverageF1Score": 0.9767700253081214,
            "AveragePrecision": 0.9767700253081214,
            "AverageRecall": 0.9767700253081214,
            "AverageAccuracy": 0.9858281665190434
        },
        "EvaluationManifestS3Prefix": "s3://amzn-s3-demo-destination-bucket/
flywheel-entity/schemaVersion=1/20230616T200543Z/evaluation/20230616T211026Z/"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的飛輪概觀。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFlywheelIteration。

describe-flywheel

以下程式碼範例顯示如何使用 describe-flywheel。

AWS CLI

描述飛輪

下列describe-flywheel範例會取得飛輪的屬性。在此範例中，與飛輪相關聯的模型是自訂分類器模型，經過訓練，可將文件分類為垃圾郵件或非垃圾郵件，或「ham」。

```
aws comprehend describe-flywheel \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel
```

輸出：

```
{
    "FlywheelProperties": {
        "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel",
        "ActiveModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-model/version/1",
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role",
        "TaskConfig": {
            "LanguageCode": "en",
            "DocumentClassificationConfig": {
                "Mode": "MULTI_CLASS",
                "Labels": [
                    "ham",
                    "spam"
                ]
            }
        },
        "DataLakeS3Uri": "s3://amzn-s3-demo-bucket/example-flywheel/schemaVersion=1/20230616T200543Z/",
        "DataSecurityConfig": {},
        "Status": "ACTIVE",
```

```
        "ModelType": "DOCUMENT_CLASSIFIER",
        "CreationTime": "2023-06-16T20:05:43.242000+00:00",
        "LastModifiedTime": "2023-06-16T20:21:43.567000+00:00"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的飛輪概觀。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFlywheel。

describe-key-phrases-detection-job

以下程式碼範例顯示如何使用 describe-key-phrases-detection-job。

AWS CLI

描述關鍵片語偵測任務

下列describe-key-phrases-detection-job範例會取得非同步金鑰片語偵測任務的屬性。

```
aws comprehend describe-key-phrases-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

輸出：

```
{
    "KeyPhrasesDetectionJobProperties": {
        "JobId": "69aa080c00fc68934a6a98f10EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-
job/69aa080c00fc68934a6a98f10EXAMPLE",
        "JobName": "example-key-phrases-detection-job",
        "JobStatus": "COMPLETED",
        "SubmitTime": 1686606439.177,
        "EndTime": 1686606806.157,
        "InputDataConfig": {
            "S3Uri": "s3://dereksbucket1001/EventsData/",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://dereksbucket1002/testfolder/111122223333-
KP-69aa080c00fc68934a6a98f10EXAMPLE/output/output.tar.gz"
        },
```

```
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-testrole"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeKeyPhrasesDetectionJob。

describe-pii-entities-detection-job

以下程式碼範例顯示如何使用 describe-pii-entities-detection-job。

AWS CLI

描述 PII 實體偵測任務

下列describe-pii-entities-detection-job範例會取得非同步 pii 實體偵測任務的屬性。

```
aws comprehend describe-pii-entities-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

輸出：

```
{
    "PiiEntitiesDetectionJobProperties": {
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "example-pii-entities-job",
        "JobStatus": "IN_PROGRESS",
        "SubmitTime": "2023-06-08T21:30:15.323000+00:00",
        "EndTime": "2023-06-08T21:40:23.509000+00:00",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/AsyncBatchJobs/",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/thefolder/111122223333-
NER-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
```

```
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::12345678012:role/service-role/
AmazonComprehendServiceRole-example-role"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePiiEntitiesDetectionJob。

describe-resource-policy

以下程式碼範例顯示如何使用 describe-resource-policy。

AWS CLI

描述連接到模型的資源政策

下列describe-resource-policy範例會取得連接至模型之資源型政策的屬性。

```
aws comprehend describe-resource-policy \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1
```

輸出：

```
{
    "ResourcePolicy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::444455556666:root\"},\"Action\":
\"comprehend:ImportModel\",\"Resource\":\"*\"}]}",
    "CreationTime": "2023-06-19T18:44:26.028000+00:00",
    "LastModifiedTime": "2023-06-19T18:53:02.002000+00:00",
    "PolicyRevisionId": "baa675d069d07afaa2aa3106ae280f61"
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的在AWS 帳戶之間複製自訂模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeResourcePolicy。

describe-sentiment-detection-job

以下程式碼範例顯示如何使用 describe-sentiment-detection-job。

AWS CLI

描述情緒偵測任務

下列describe-sentiment-detection-job範例會取得非同步情緒偵測任務的屬性。

```
aws comprehend describe-sentiment-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

輸出：

```
{
    "SentimentDetectionJobProperties": {
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "movie_review_analysis",
        "JobStatus": "IN_PROGRESS",
        "SubmitTime": "2023-06-09T23:16:15.956000+00:00",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/MovieData",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-destination-bucket/testfolder/111122223333-
TS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSentimentDetectionJob。

describe-targeted-sentiment-detection-job

以下程式碼範例顯示如何使用 describe-targeted-sentiment-detection-job。

AWS CLI

描述目標情緒偵測任務

下列describe-targeted-sentiment-detection-job範例會取得非同步目標情緒偵測任務的屬性。

```
aws comprehend describe-targeted-sentiment-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

輸出：

```
{
    "TargetedSentimentDetectionJobProperties": {
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "movie_review_analysis",
        "JobStatus": "IN_PROGRESS",
        "SubmitTime": "2023-06-09T23:16:15.956000+00:00",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/MovieData",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-destination-bucket/testfolder/111122223333-
TS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTargetedSentimentDetectionJob。

describe-topics-detection-job

以下程式碼範例顯示如何使用 describe-topics-detection-job。

AWS CLI

描述主題偵測任務

下列describe-topics-detection-job範例會取得非同步主題偵測任務的屬性。

```
aws comprehend describe-topics-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

輸出：

```
{
    "TopicsDetectionJobProperties": {
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "example_topics_detection",
        "JobStatus": "IN_PROGRESS",
        "SubmitTime": "2023-06-09T18:44:43.414000+00:00",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-destination-bucket/testfolder/111122223333-
TOPICS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
        },
        "NumberOfTopics": 10,
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-examplerole"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTopicsDetectionJob。

detect-dominant-language

以下程式碼範例顯示如何使用 detect-dominant-language。

AWS CLI

偵測輸入文字的主要語言

以下內容會detect-dominant-language分析輸入文字並識別主要語言。也會輸出預先訓練模型的可信度分數。

```
aws comprehend detect-dominant-language \
    --text "It is a beautiful day in Seattle."
```

輸出：

```
{
    "Languages": [
        {
            "LanguageCode": "en",
            "Score": 0.9877256155014038
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的主要語言。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetectDominantLanguage。

detect-entities

以下程式碼範例顯示如何使用 detect-entities。

AWS CLI

在輸入文字中偵測具名實體

下列detect-entities範例會分析輸入文字並傳回具名實體。每個預測也會輸出預先訓練模型的可信度分數。

```
aws comprehend detect-entities \
    --language-code en \
```

```
    --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC
credit card \
    account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July
31st. Based on your autopay settings, \
    we will withdraw your payment on the due date from your bank account number
XXXXXX1111 with the routing number XXXXX0000. \
    Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to
Alice at AnySpa@example.com."
```

輸出：

```
{
    "Entities": [
        {
            "Score": 0.9994556307792664,
            "Type": "PERSON",
            "Text": "Zhang Wei",
            "BeginOffset": 6,
            "EndOffset": 15
        },
        {
            "Score": 0.9981022477149963,
            "Type": "PERSON",
            "Text": "John",
            "BeginOffset": 22,
            "EndOffset": 26
        },
        {
            "Score": 0.9986887574195862,
            "Type": "ORGANIZATION",
            "Text": "AnyCompany Financial Services, LLC",
            "BeginOffset": 33,
            "EndOffset": 67
        },
        {
            "Score": 0.9959119558334351,
            "Type": "OTHER",
            "Text": "1111-XXXX-1111-XXXX",
            "BeginOffset": 88,
            "EndOffset": 107
        },
        {
            "Score": 0.9708039164543152,
```

```
            "Type": "QUANTITY",
            "Text": ".53",
            "BeginOffset": 133,
            "EndOffset": 136
        },
        {
            "Score": 0.9987268447875977,
            "Type": "DATE",
            "Text": "July 31st",
            "BeginOffset": 152,
            "EndOffset": 161
        },
        {
            "Score": 0.9858865737915039,
            "Type": "OTHER",
            "Text": "XXXXXX1111",
            "BeginOffset": 271,
            "EndOffset": 281
        },
        {
            "Score": 0.9700471758842468,
            "Type": "OTHER",
            "Text": "XXXXX0000",
            "BeginOffset": 306,
            "EndOffset": 315
        },
        {
            "Score": 0.9591118693351746,
            "Type": "ORGANIZATION",
            "Text": "Sunshine Spa",
            "BeginOffset": 340,
            "EndOffset": 352
        },
        {
            "Score": 0.9797496795654297,
            "Type": "LOCATION",
            "Text": "123 Main St",
            "BeginOffset": 354,
            "EndOffset": 365
        },
        {
            "Score": 0.994929313659668,
            "Type": "PERSON",
            "Text": "Alice",
```

```
            "BeginOffset": 394,
            "EndOffset": 399
        },
        {
            "Score": 0.9949769377708435,
            "Type": "OTHER",
            "Text": "AnySpa@example.com",
            "BeginOffset": 403,
            "EndOffset": 418
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的實體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetectEntities。

detect-key-phrases

以下程式碼範例顯示如何使用 detect-key-phrases。

AWS CLI

偵測輸入文字中的金鑰片語

下列detect-key-phrases範例會分析輸入文字，並識別金鑰名詞片語。每個預測也會輸出預先訓練模型的可信度分數。

```
aws comprehend detect-key-phrases \
    --language-code en \
    --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC
 credit card \
       account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by
 July 31st. Based on your autopay settings, \
       we will withdraw your payment on the due date from your bank account number
 XXXXXX1111 with the routing number XXXXX0000. \
       Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to
 Alice at AnySpa@example.com."
```

輸出：

```
{
    "KeyPhrases": [
```

```
        {
            "Score": 0.8996376395225525,
            "Text": "Zhang Wei",
            "BeginOffset": 6,
            "EndOffset": 15
        },
        {
            "Score": 0.9992469549179077,
            "Text": "John",
            "BeginOffset": 22,
            "EndOffset": 26
        },
        {
            "Score": 0.988385021686554,
            "Text": "Your AnyCompany Financial Services",
            "BeginOffset": 28,
            "EndOffset": 62
        },
        {
            "Score": 0.8740853071212769,
            "Text": "LLC credit card account 1111-XXXX-1111-XXXX",
            "BeginOffset": 64,
            "EndOffset": 107
        },
        {
            "Score": 0.9999437928199768,
            "Text": "a minimum payment",
            "BeginOffset": 112,
            "EndOffset": 129
        },
        {
            "Score": 0.9998900890350342,
            "Text": ".53",
            "BeginOffset": 133,
            "EndOffset": 136
        },
        {
            "Score": 0.9979453086853027,
            "Text": "July 31st",
            "BeginOffset": 152,
            "EndOffset": 161
        },
        {
            "Score": 0.9983011484146118,
```

```
            "Text": "your autopay settings",
            "BeginOffset": 172,
            "EndOffset": 193
        },
        {
            "Score": 0.9996572136878967,
            "Text": "your payment",
            "BeginOffset": 211,
            "EndOffset": 223
        },
        {
            "Score": 0.9995037317276001,
            "Text": "the due date",
            "BeginOffset": 227,
            "EndOffset": 239
        },
        {
            "Score": 0.9702621698379517,
            "Text": "your bank account number XXXXXX1111",
            "BeginOffset": 245,
            "EndOffset": 280
        },
        {
            "Score": 0.9179925918579102,
            "Text": "the routing number XXXXX0000.Customer feedback",
            "BeginOffset": 286,
            "EndOffset": 332
        },
        {
            "Score": 0.9978160858154297,
            "Text": "Sunshine Spa",
            "BeginOffset": 337,
            "EndOffset": 349
        },
        {
            "Score": 0.9706913232803345,
            "Text": "123 Main St",
            "BeginOffset": 351,
            "EndOffset": 362
        },
        {
            "Score": 0.9941995143890381,
            "Text": "comments",
            "BeginOffset": 379,
```

```
            "EndOffset": 387
        },
        {
            "Score": 0.9759287238121033,
            "Text": "Alice",
            "BeginOffset": 391,
            "EndOffset": 396
        },
        {
            "Score": 0.8376792669296265,
            "Text": "AnySpa@example.com",
            "BeginOffset": 400,
            "EndOffset": 415
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的關鍵詞。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetectKeyPhrases。

`detect-pii-entities`

以下程式碼範例顯示如何使用 `detect-pii-entities`。

AWS CLI

偵測輸入文字中的 pii 實體

下列`detect-pii-entities`範例會分析輸入文字，並識別包含個人身分識別資訊 (PII) 的實體。每個預測也會輸出預先訓練模型的可信度分數。

```
aws comprehend detect-pii-entities \
    --language-code en \
    --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC
 credit card \
       account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by
 July 31st. Based on your autopay settings, \
       we will withdraw your payment on the due date from your bank account number
 XXXXXX1111 with the routing number XXXXX0000. \
       Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to
 Alice at AnySpa@example.com."
```

輸出：


```
{
    "Entities": [
        {
            "Score": 0.9998322129249573,
            "Type": "NAME",
            "BeginOffset": 6,
            "EndOffset": 15
        },
        {
            "Score": 0.9998878240585327,
            "Type": "NAME",
            "BeginOffset": 22,
            "EndOffset": 26
        },
        {
            "Score": 0.9994089603424072,
            "Type": "CREDIT_DEBIT_NUMBER",
            "BeginOffset": 88,
            "EndOffset": 107
        },
        {
            "Score": 0.9999760985374451,
            "Type": "DATE_TIME",
            "BeginOffset": 152,
            "EndOffset": 161
        },
        {
            "Score": 0.9999449253082275,
            "Type": "BANK_ACCOUNT_NUMBER",
            "BeginOffset": 271,
            "EndOffset": 281
        },
        {
            "Score": 0.9999847412109375,
            "Type": "BANK_ROUTING",
            "BeginOffset": 306,
            "EndOffset": 315
        },
        {
            "Score": 0.999925434589386,
            "Type": "ADDRESS",
            "BeginOffset": 354,
```

```
            "EndOffset": 365
        },
        {
            "Score": 0.9989161491394043,
            "Type": "NAME",
            "BeginOffset": 394,
            "EndOffset": 399
        },
        {
            "Score": 0.9994171857833862,
            "Type": "EMAIL",
            "BeginOffset": 403,
            "EndOffset": 418
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的個人身分識別資訊 (PII)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetectPiiEntities。

`detect-sentiment`

以下程式碼範例顯示如何使用 `detect-sentiment`。

AWS CLI

偵測輸入文字的情緒

下列`detect-sentiment`範例會分析輸入文字，並傳回普遍情緒的推論 (`POSITIVE`、`MIXED`、`NEUTRAL`或 `NEGATIVE`)。

```
aws comprehend detect-sentiment \
    --language-code en \
    --text "It is a beautiful day in Seattle"
```

輸出：

```
{
    "Sentiment": "POSITIVE",
    "SentimentScore": {
        "Positive": 0.9976957440376282,
        "Negative": 9.653854067437351e-05,
```

```
        "Neutral": 0.002169104292988777,
        "Mixed": 3.857641786453314e-05
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的情緒

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetectSentiment。

`detect-syntax`

以下程式碼範例顯示如何使用 `detect-syntax`。

AWS CLI

偵測輸入文字中的語音部分

下列`detect-syntax`範例會分析輸入文字的語法，並傳回語音的不同部分。每個預測也會輸出預先訓練模型的可信度分數。

```
aws comprehend detect-syntax \
    --language-code en \
    --text "It is a beautiful day in Seattle."
```

輸出：

```
{
    "SyntaxTokens": [
        {
            "TokenId": 1,
            "Text": "It",
            "BeginOffset": 0,
            "EndOffset": 2,
            "PartOfSpeech": {
                "Tag": "PRON",
                "Score": 0.9999740719795227
            }
        },
        {
            "TokenId": 2,
            "Text": "is",
            "BeginOffset": 3,
            "EndOffset": 5,
```

```
            "PartOfSpeech": {
                "Tag": "VERB",
                "Score": 0.999901294708252
            }
        },
        {
            "TokenId": 3,
            "Text": "a",
            "BeginOffset": 6,
            "EndOffset": 7,
            "PartOfSpeech": {
                "Tag": "DET",
                "Score": 0.9999938607215881
            }
        },
        {
            "TokenId": 4,
            "Text": "beautiful",
            "BeginOffset": 8,
            "EndOffset": 17,
            "PartOfSpeech": {
                "Tag": "ADJ",
                "Score": 0.9987351894378662
            }
        },
        {
            "TokenId": 5,
            "Text": "day",
            "BeginOffset": 18,
            "EndOffset": 21,
            "PartOfSpeech": {
                "Tag": "NOUN",
                "Score": 0.9999796748161316
            }
        },
        {
            "TokenId": 6,
            "Text": "in",
            "BeginOffset": 22,
            "EndOffset": 24,
            "PartOfSpeech": {
                "Tag": "ADP",
                "Score": 0.9998047947883606
            }
```

```
        },
        {
            "TokenId": 7,
            "Text": "Seattle",
            "BeginOffset": 25,
            "EndOffset": 32,
            "PartOfSpeech": {
                "Tag": "PROPN",
                "Score": 0.9940530061721802
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的語法分析。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetectSyntax。

`detect-targeted-sentiment`

以下程式碼範例顯示如何使用 `detect-targeted-sentiment`。

AWS CLI

在輸入文字中偵測具名實體的目標情緒

下列`detect-targeted-sentiment`範例會分析輸入文字，並傳回具名實體，以及與每個實體相關聯的目標情緒。也會輸出每個預測的預先訓練模型可信度分數。

```
aws comprehend detect-targeted-sentiment \
    --language-code en \
    --text "I do not enjoy January because it is too cold but August is the perfect temperature"
```

輸出：

```
{
    "Entities": [
        {
            "DescriptiveMentionIndex": [
                0
            ],
            "Mentions": [
```

```
                {
                    "Score": 0.9999979734420776,
                    "GroupScore": 1.0,
                    "Text": "I",
                    "Type": "PERSON",
                    "MentionSentiment": {
                        "Sentiment": "NEUTRAL",
                        "SentimentScore": {
                            "Positive": 0.0,
                            "Negative": 0.0,
                            "Neutral": 1.0,
                            "Mixed": 0.0
                        }
                    },
                    "BeginOffset": 0,
                    "EndOffset": 1
                }
            ]
        },
        {
            "DescriptiveMentionIndex": [
                0
            ],
            "Mentions": [
                {
                    "Score": 0.9638869762420654,
                    "GroupScore": 1.0,
                    "Text": "January",
                    "Type": "DATE",
                    "MentionSentiment": {
                        "Sentiment": "NEGATIVE",
                        "SentimentScore": {
                            "Positive": 0.0031610000878572464,
                            "Negative": 0.9967250227928162,
                            "Neutral": 0.00011100000119768083,
                            "Mixed": 1.9999999949504854e-06
                        }
                    },
                    "BeginOffset": 15,
                    "EndOffset": 22
                }
            ]
        },
        {
```

```
            "DescriptiveMentionIndex": [
                0
            ],
            "Mentions": [
                {
                {
                    "Score": 0.9664419889450073,
                    "GroupScore": 1.0,
                    "Text": "August",
                    "Type": "DATE",
                    "MentionSentiment": {
                        "Sentiment": "POSITIVE",
                        "SentimentScore": {
                            "Positive": 0.9999549984931946,
                            "Negative": 3.999999989900971e-06,
                            "Neutral": 4.099999932805076e-05,
                            "Mixed": 0.0
                        }
                    },
                    "BeginOffset": 50,
                    "EndOffset": 56
                }
            ]
        },
        {
            "DescriptiveMentionIndex": [
                0
            ],
            "Mentions": [
                {
                    "Score": 0.9803199768066406,
                    "GroupScore": 1.0,
                    "Text": "temperature",
                    "Type": "ATTRIBUTE",
                    "MentionSentiment": {
                        "Sentiment": "POSITIVE",
                        "SentimentScore": {
                            "Positive": 1.0,
                            "Negative": 0.0,
                            "Neutral": 0.0,
                            "Mixed": 0.0
                        }
                    },
                    "BeginOffset": 77,
```

```
                    "EndOffset": 88
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的目標情緒。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetectTargetedSentiment。

import-model

以下程式碼範例顯示如何使用 import-model。

AWS CLI

匯入模型

下列import-model範例會從不同的 AWS 帳戶匯入模型。帳戶中的文件分類器模型444455556666具有資源型政策111122223333，可讓帳戶匯入模型。

```
aws comprehend import-model \
    --source-model-arn arn:aws:comprehend:us-west-2:444455556666:document-classifier/example-classifier
```

輸出：

```
{
    "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier"
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的在AWS 帳戶之間複製自訂模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ImportModel。

list-datasets

以下程式碼範例顯示如何使用 list-datasets。

AWS CLI

列出所有飛輪資料集

下列list-datasets範例列出與飛輪相關聯的所有資料集。

```
aws comprehend list-datasets \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-
entity
```

輸出：

```
{
    "DatasetPropertiesList": [
        {
            "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
flywheel-entity/dataset/example-dataset-1",
            "DatasetName": "example-dataset-1",
            "DatasetType": "TRAIN",
            "DatasetS3Uri": "s3://amzn-s3-demo-bucket/flywheel-entity/
schemaVersion=1/20230616T200543Z/datasets/example-dataset-1/20230616T203710Z/",
            "Status": "CREATING",
            "CreationTime": "2023-06-16T20:37:10.400000+00:00"
        },
        {
            "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
flywheel-entity/dataset/example-dataset-2",
            "DatasetName": "example-dataset-2",
            "DatasetType": "TRAIN",
            "DatasetS3Uri": "s3://amzn-s3-demo-bucket/flywheel-entity/
schemaVersion=1/20230616T200543Z/datasets/example-dataset-2/20230616T200607Z/",
            "Description": "TRAIN Dataset created by Flywheel creation.",
            "Status": "COMPLETED",
            "NumberOfDocuments": 5572,
            "CreationTime": "2023-06-16T20:06:07.722000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的飛輪概觀。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDatasets。

list-document-classification-jobs

以下程式碼範例顯示如何使用 list-document-classification-jobs。

AWS CLI

列出所有文件分類任務

下列list-document-classification-jobs範例列出所有文件分類任務。

```
aws comprehend list-document-classification-jobs
```

輸出：

```
{
    "DocumentClassificationJobPropertiesList": [
        {
            "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:1234567890101:document-
classification-job/123456abcdeb0e11022f22a11EXAMPLE",
            "JobName": "exampleclassificationjob",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-14T17:09:51.788000+00:00",
            "EndTime": "2023-06-14T17:15:58.582000+00:00",
            "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:1234567890101:document-classifier/mymodel/version/12",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/jobdata/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/1234567890101-CLN-e758dd56b824aa717ceab551f11749fb/output/output.tar.gz"
            },
            "DataAccessRoleArn": "arn:aws:iam::1234567890101:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "123456abcdeb0e11022f22a1EXAMPLE2",
            "JobArn": "arn:aws:comprehend:us-west-2:1234567890101:document-
classification-job/123456abcdeb0e11022f22a1EXAMPLE2",
            "JobName": "exampleclassificationjob2",
            "JobStatus": "COMPLETED",
```

```
            "SubmitTime": "2023-06-14T17:22:39.829000+00:00",
            "EndTime": "2023-06-14T17:28:46.107000+00:00",
            "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:1234567890101:document-classifier/mymodel/version/12",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/jobdata/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/1234567890101-CLN-123456abcdeb0e11022f22a1EXAMPLE2/output/output.tar.gz"
            },
            "DataAccessRoleArn": "arn:aws:iam::1234567890101:role/service-role/
AmazonComprehendServiceRole-example-role"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的自訂分類。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDocumentClassificationJobs。

list-document-classifier-summaries

以下程式碼範例顯示如何使用 list-document-classifier-summaries。

AWS CLI

列出所有已建立文件分類器的摘要

下列list-document-classifier-summaries範例列出所有建立的文件分類器摘要。

```
aws comprehend list-document-classifier-summaries
```

輸出：

```
{
    "DocumentClassifierSummariesList": [
        {
            "DocumentClassifierName": "example-classifier-1",
            "NumberOfVersions": 1,
            "LatestVersionCreatedAt": "2023-06-13T22:07:59.825000+00:00",
```

```
            "LatestVersionName": "1",
            "LatestVersionStatus": "TRAINED"
        },
        {
            "DocumentClassifierName": "example-classifier-2",
            "NumberOfVersions": 2,
            "LatestVersionCreatedAt": "2023-06-13T21:54:59.589000+00:00",
            "LatestVersionName": "2",
            "LatestVersionStatus": "TRAINED"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的建立和管理自訂模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDocumentClassifierSummaries。

`list-document-classifiers`

以下程式碼範例顯示如何使用 `list-document-classifiers`。

AWS CLI

列出所有文件分類器

下列`list-document-classifiers`範例列出所有訓練和訓練中文件分類器模型。

```
aws comprehend list-document-classifiers
```

輸出：

```
{
    "DocumentClassifierPropertiesList": [
        {
            "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/exampleclassifier1",
            "LanguageCode": "en",
            "Status": "TRAINED",
            "SubmitTime": "2023-06-13T19:04:15.735000+00:00",
            "EndTime": "2023-06-13T19:42:31.752000+00:00",
            "TrainingStartTime": "2023-06-13T19:08:20.114000+00:00",
            "TrainingEndTime": "2023-06-13T19:41:35.080000+00:00",
            "InputDataConfig": {
```

```
                "DataFormat": "COMPREHEND_CSV",
                "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata"
            },
            "OutputDataConfig": {},
            "ClassifierMetadata": {
                "NumberOfLabels": 3,
                "NumberOfTrainedDocuments": 5016,
                "NumberOfTestDocuments": 557,
                "EvaluationMetrics": {
                    "Accuracy": 0.9856,
                    "Precision": 0.9919,
                    "Recall": 0.9459,
                    "F1Score": 0.9673,
                    "MicroPrecision": 0.9856,
                    "MicroRecall": 0.9856,
                    "MicroF1Score": 0.9856,
                    "HammingLoss": 0.0144
                }
            },
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-testorle",
            "Mode": "MULTI_CLASS"
        },
        {
            "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/exampleclassifier2",
            "LanguageCode": "en",
            "Status": "TRAINING",
            "SubmitTime": "2023-06-13T21:20:28.690000+00:00",
            "InputDataConfig": {
                "DataFormat": "COMPREHEND_CSV",
                "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata"
            },
            "OutputDataConfig": {},
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-testorle",
            "Mode": "MULTI_CLASS"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的建立和管理自訂模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDocumentClassifiers。

list-dominant-language-detection-jobs

以下程式碼範例顯示如何使用 list-dominant-language-detection-jobs。

AWS CLI

列出所有主要語言偵測任務

下列list-dominant-language-detection-jobs範例列出所有進行中和已完成的非同步主要語言偵測任務。

```
aws comprehend list-dominant-language-detection-jobs
```

輸出：

```
{
    "DominantLanguageDetectionJobPropertiesList": [
        {
            "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
            "JobName": "languageanalysis1",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-09T18:10:38.037000+00:00",
            "EndTime": "2023-06-09T18:18:45.498000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-LANGUAGE-123456abcdeb0e11022f22a11EXAMPLE/output/
output.tar.gz"
            },
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
            "JobName": "languageanalysis2",
            "JobStatus": "STOPPED",
```

```
            "SubmitTime": "2023-06-09T18:16:33.690000+00:00",
            "EndTime": "2023-06-09T18:24:40.608000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-LANGUAGE-123456abcdeb0e11022f22a11EXAMPLE/output/
output.tar.gz"
            },
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDominantLanguageDetectionJobs。

`list-endpoints`

以下程式碼範例顯示如何使用 `list-endpoints`。

AWS CLI

列出所有端點

下列`list-endpoints`範例列出所有作用中的模型特定端點。

```
aws comprehend list-endpoints
```

輸出：

```
{
    "EndpointPropertiesList": [
        {
            "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier-endpoint/ExampleClassifierEndpoint",
```

```
            "Status": "IN_SERVICE",
            "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier1",
            "DesiredModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier1",
            "DesiredInferenceUnits": 1,
            "CurrentInferenceUnits": 1,
            "CreationTime": "2023-06-13T20:32:54.526000+00:00",
            "LastModifiedTime": "2023-06-13T20:32:54.526000+00:00"
        },
        {
            "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier-endpoint/ExampleClassifierEndpoint2",
            "Status": "IN_SERVICE",
            "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier2",
            "DesiredModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier2",
            "DesiredInferenceUnits": 1,
            "CurrentInferenceUnits": 1,
            "CreationTime": "2023-06-13T20:32:54.526000+00:00",
            "LastModifiedTime": "2023-06-13T20:32:54.526000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的管理 Amazon Comprehend 端點。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEndpoints。

`list-entities-detection-jobs`

以下程式碼範例顯示如何使用 `list-entities-detection-jobs`。

AWS CLI

列出所有實體偵測任務

下列`list-entities-detection-jobs`範例列出所有非同步實體偵測任務。

```
aws comprehend list-entities-detection-jobs
```

輸出：

```
{
    "EntitiesDetectionJobPropertiesList": [
        {
            "JobId": "468af39c28ab45b83eb0c4ab9EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/468af39c28ab45b83eb0c4ab9EXAMPLE",
            "JobName": "example-entities-detection",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-08T20:57:46.476000+00:00",
            "EndTime": "2023-06-08T21:05:53.718000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/111122223333-NER-468af39c28ab45b83eb0c4ab9EXAMPLE/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "809691caeaab0e71406f80a28EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/809691caeaab0e71406f80a28EXAMPLE",
            "JobName": "example-entities-detection-2",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-08T21:30:15.323000+00:00",
            "EndTime": "2023-06-08T21:40:23.509000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/111122223333-NER-809691caeaab0e71406f80a28EXAMPLE/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
```

```
        {
            "JobId": "e00597c36b448b91d70dea165EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/e00597c36b448b91d70dea165EXAMPLE",
            "JobName": "example-entities-detection-3",
            "JobStatus": "STOPPED",
            "SubmitTime": "2023-06-08T22:19:28.528000+00:00",
            "EndTime": "2023-06-08T22:27:33.991000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/111122223333-NER-e00597c36b448b91d70dea165EXAMPLE/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的實體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEntitiesDetectionJobs。

`list-entity-recognizer-summaries`

以下程式碼範例顯示如何使用 `list-entity-recognizer-summaries`。

AWS CLI

若要列出所有已建立實體辨識器的摘要

下列`list-entity-recognizer-summaries`範例列出所有實體辨識器摘要。

```
aws comprehend list-entity-recognizer-summaries
```

輸出：

```
{
    "EntityRecognizerSummariesList": [
```

```
        {
            "RecognizerName": "entity-recognizer-3",
            "NumberOfVersions": 2,
            "LatestVersionCreatedAt": "2023-06-15T23:15:07.621000+00:00",
            "LatestVersionName": "2",
            "LatestVersionStatus": "STOP_REQUESTED"
        },
        {
            "RecognizerName": "entity-recognizer-2",
            "NumberOfVersions": 1,
            "LatestVersionCreatedAt": "2023-06-14T22:55:27.805000+00:00",
            "LatestVersionName": "2"
            "LatestVersionStatus": "TRAINED"
        },
        {
            "RecognizerName": "entity-recognizer-1",
            "NumberOfVersions": 1,
            "LatestVersionCreatedAt": "2023-06-14T20:44:59.631000+00:00",
            "LatestVersionName": "1",
            "LatestVersionStatus": "TRAINED"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的自訂實體辨識。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEntityRecognizerSummaries。

list-entity-recognizers

以下程式碼範例顯示如何使用 list-entity-recognizers。

AWS CLI

列出所有自訂實體辨識器

下列list-entity-recognizers範例列出所有建立的自訂實體識別器。

```
aws comprehend list-entity-recognizers
```

輸出：

```
{
```

```
    "EntityRecognizerPropertiesList": [
        {
            "EntityRecognizerArn": "arn:aws:comprehend:us-
west-2:111122223333:entity-recognizer/EntityRecognizer/version/1",
            "LanguageCode": "en",
            "Status": "TRAINED",
            "SubmitTime": "2023-06-14T20:44:59.631000+00:00",
            "EndTime": "2023-06-14T20:59:19.532000+00:00",
            "TrainingStartTime": "2023-06-14T20:48:52.811000+00:00",
            "TrainingEndTime": "2023-06-14T20:58:11.473000+00:00",
            "InputDataConfig": {
                "DataFormat": "COMPREHEND_CSV",
                "EntityTypes": [
                    {
                        "Type": "BUSINESS"
                    }
                ],
                "Documents": {
                    "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata/dataset/",
                    "InputFormat": "ONE_DOC_PER_LINE"
                },
                "EntityList": {
                    "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata/entity.csv"
                }
            },
            "RecognizerMetadata": {
                "NumberOfTrainedDocuments": 1814,
                "NumberOfTestDocuments": 486,
                "EvaluationMetrics": {
                    "Precision": 100.0,
                    "Recall": 100.0,
                    "F1Score": 100.0
                },
                "EntityTypes": [
                    {
                        "Type": "BUSINESS",
                        "EvaluationMetrics": {
                            "Precision": 100.0,
                            "Recall": 100.0,
                            "F1Score": 100.0
                        },
                        "NumberOfTrainMentions": 1520
                    }
                ]
```

```
            },
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole",
            "VersionName": "1"
        },
        {
            "EntityRecognizerArn": "arn:aws:comprehend:us-
west-2:111122223333:entity-recognizer/entityrecognizer3",
            "LanguageCode": "en",
            "Status": "TRAINED",
            "SubmitTime": "2023-06-14T22:57:51.056000+00:00",
            "EndTime": "2023-06-14T23:14:13.894000+00:00",
            "TrainingStartTime": "2023-06-14T23:01:33.984000+00:00",
            "TrainingEndTime": "2023-06-14T23:13:02.984000+00:00",
            "InputDataConfig": {
                "DataFormat": "COMPREHEND_CSV",
                "EntityTypes": [
                    {
                        "Type": "DEVICE"
                    }
                ],
                "Documents": {
                    "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata/raw_txt.csv",
                    "InputFormat": "ONE_DOC_PER_LINE"
                },
                "EntityList": {
                    "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata/entity_list.csv"
                }
            },
            "RecognizerMetadata": {
                "NumberOfTrainedDocuments": 4616,
                "NumberOfTestDocuments": 3489,
                "EvaluationMetrics": {
                    "Precision": 98.54227405247813,
                    "Recall": 100.0,
                    "F1Score": 99.26578560939794
                },
                "EntityTypes": [
                    {
                        "Type": "DEVICE",
                        "EvaluationMetrics": {
                            "Precision": 98.54227405247813,
                            "Recall": 100.0,
                            "F1Score": 99.26578560939794
```

```
                        },
                        "NumberOfTrainMentions": 2764
                    }
                ]
            },
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的自訂實體辨識。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEntityRecognizers。

list-events-detection-jobs

以下程式碼範例顯示如何使用 list-events-detection-jobs。

AWS CLI

列出所有事件偵測任務

下列list-events-detection-jobs範例列出所有非同步事件偵測任務。

```
aws comprehend list-events-detection-jobs
```

輸出：

```
{
    "EventsDetectionJobPropertiesList": [
        {
            "JobId": "aa9593f9203e84f3ef032ce18EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:1111222233333:events-detection-
job/aa9593f9203e84f3ef032ce18EXAMPLE",
            "JobName": "events_job_1",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-12T19:14:57.751000+00:00",
            "EndTime": "2023-06-12T19:21:04.962000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-source-bucket/EventsData/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
```

```
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/1111222233333-EVENTS-aa9593f9203e84f3ef032ce18EXAMPLE/output/"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::1111222233333:role/service-role/
AmazonComprehendServiceRole-example-role",
            "TargetEventTypes": [
                "BANKRUPTCY",
                "EMPLOYMENT",
                "CORPORATE_ACQUISITION",
                "CORPORATE_MERGER",
                "INVESTMENT_GENERAL"
            ]
        },
        {
            "JobId": "4a990a2f7e82adfca6e171135EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:1111222233333:events-detection-
job/4a990a2f7e82adfca6e171135EXAMPLE",
            "JobName": "events_job_2",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-12T19:55:43.702000+00:00",
            "EndTime": "2023-06-12T20:03:49.893000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-source-bucket/EventsData/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/1111222233333-EVENTS-4a990a2f7e82adfca6e171135EXAMPLE/output/"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::1111222233333:role/service-role/
AmazonComprehendServiceRole-example-role",
            "TargetEventTypes": [
                "BANKRUPTCY",
                "EMPLOYMENT",
                "CORPORATE_ACQUISITION",
                "CORPORATE_MERGER",
                "INVESTMENT_GENERAL"
            ]
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEventsDetectionJobs。

`list-flywheel-iteration-history`

以下程式碼範例顯示如何使用 `list-flywheel-iteration-history`。

AWS CLI

列出所有飛輪反覆運算歷史記錄

下列`list-flywheel-iteration-history`範例列出飛輪的所有反覆運算。

```
aws comprehend list-flywheel-iteration-history
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-
flywheel
```

輸出：

```
{
    "FlywheelIterationPropertiesList": [
        {
            "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
example-flywheel",
            "FlywheelIterationId": "20230619TEXAMPLE",
            "CreationTime": "2023-06-19T04:00:32.594000+00:00",
            "EndTime": "2023-06-19T04:00:49.248000+00:00",
            "Status": "COMPLETED",
            "Message": "FULL_ITERATION: Flywheel iteration performed all functions
 successfully.",
            "EvaluatedModelArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/example-classifier/version/1",
            "EvaluatedModelMetrics": {
                "AverageF1Score": 0.7742663922375772,
                "AverageF1Score": 0.9876464664646313,
                "AveragePrecision": 0.9800000253081214,
                "AverageRecall": 0.9445600253081214,
                "AverageAccuracy": 0.9997281665190434
```

```
            },
            "EvaluationManifestS3Prefix": "s3://amzn-s3-demo-bucket/example-
flywheel/schemaVersion=1/20230619TEXAMPLE/evaluation/20230619TEXAMPLE/"
        },
        {
            "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
example-flywheel-2",
            "FlywheelIterationId": "20230616TEXAMPLE",
            "CreationTime": "2023-06-16T21:10:26.385000+00:00",
            "EndTime": "2023-06-16T23:33:16.827000+00:00",
            "Status": "COMPLETED",
            "Message": "FULL_ITERATION: Flywheel iteration performed all functions
 successfully.",
            "EvaluatedModelArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/spamvshamclassify/version/1",
            "EvaluatedModelMetrics": {
                "AverageF1Score": 0.7742663922375772,
                "AverageF1Score": 0.9767700253081214,
                "AveragePrecision": 0.9767700253081214,
                "AverageRecall": 0.9767700253081214,
                "AverageAccuracy": 0.9858281665190434
            },
            "EvaluationManifestS3Prefix": "s3://amzn-s3-demo-bucket/example-
flywheel-2/schemaVersion=1/20230616TEXAMPLE/evaluation/20230616TEXAMPLE/"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的飛輪概觀。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFlywheelIterationHistory。

list-flywheels

以下程式碼範例顯示如何使用 list-flywheels。

AWS CLI

列出所有飛輪

下列list-flywheels範例列出所有建立的飛輪。

```
aws comprehend list-flywheels
```

輸出：

```
{
    "FlywheelSummaryList": [
        {
            "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
example-flywheel-1",
            "ActiveModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier/version/1",
            "DataLakeS3Uri": "s3://amzn-s3-demo-bucket/example-flywheel-1/
schemaVersion=1/20230616T200543Z/",
            "Status": "ACTIVE",
            "ModelType": "DOCUMENT_CLASSIFIER",
            "CreationTime": "2023-06-16T20:05:43.242000+00:00",
            "LastModifiedTime": "2023-06-19T04:00:43.027000+00:00",
            "LatestFlywheelIteration": "20230619T040032Z"
        },
        {
            "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
example-flywheel-2",
            "ActiveModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier2/version/1",
            "DataLakeS3Uri": "s3://amzn-s3-demo-bucket/example-flywheel-2/
schemaVersion=1/20220616T200543Z/",
            "Status": "ACTIVE",
            "ModelType": "DOCUMENT_CLASSIFIER",
            "CreationTime": "2022-06-16T20:05:43.242000+00:00",
            "LastModifiedTime": "2022-06-19T04:00:43.027000+00:00",
            "LatestFlywheelIteration": "20220619T040032Z"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的飛輪概觀。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFlywheels。

list-key-phrases-detection-jobs

以下程式碼範例顯示如何使用 list-key-phrases-detection-jobs。

AWS CLI

列出所有關鍵片語偵測任務

下列list-key-phrases-detection-jobs範例列出所有進行中和已完成的非同步金鑰片語偵測任務。

```
aws comprehend list-key-phrases-detection-jobs
```

輸出：

```
{
    "KeyPhrasesDetectionJobPropertiesList": [
        {
            "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
            "JobName": "keyphrasesanalysis1",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-08T22:31:43.767000+00:00",
            "EndTime": "2023-06-08T22:39:52.565000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-source-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-KP-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "123456abcdeb0e11022f22a33EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-
detection-job/123456abcdeb0e11022f22a33EXAMPLE",
            "JobName": "keyphrasesanalysis2",
            "JobStatus": "STOPPED",
            "SubmitTime": "2023-06-08T22:57:52.154000+00:00",
            "EndTime": "2023-06-08T23:05:48.385000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/AsyncBatchJobs/",
```

```
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-KP-123456abcdeb0e11022f22a33EXAMPLE/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "123456abcdeb0e11022f22a44EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-
detection-job/123456abcdeb0e11022f22a44EXAMPLE",
            "JobName": "keyphrasesanalysis3",
            "JobStatus": "FAILED",
            "Message": "NO_READ_ACCESS_TO_INPUT: The provided data access role does
 not have proper access to the input data.",
            "SubmitTime": "2023-06-09T16:47:04.029000+00:00",
            "EndTime": "2023-06-09T16:47:18.413000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-KP-123456abcdeb0e11022f22a44EXAMPLE/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListKeyPhrasesDetectionJobs。

`list-pii-entities-detection-jobs`

以下程式碼範例顯示如何使用 `list-pii-entities-detection-jobs`。

AWS CLI

列出所有 pii 實體偵測任務

下列list-pii-entities-detection-jobs範例列出所有進行中和已完成的非同步 pii 偵測任務。

```
aws comprehend list-pii-entities-detection-jobs
```

輸出：

```
{
    "PiiEntitiesDetectionJobPropertiesList": [
        {
            "JobId": "6f9db0c42d0c810e814670ee4EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-
detection-job/6f9db0c42d0c810e814670ee4EXAMPLE",
            "JobName": "example-pii-detection-job",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-09T21:02:46.241000+00:00",
            "EndTime": "2023-06-09T21:12:52.602000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-source-bucket/111122223333-
PII-6f9db0c42d0c810e814670ee4EXAMPLE/output/"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
            "Mode": "ONLY_OFFSETS"
        },
        {
            "JobId": "d927562638cfa739331a99b3cEXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-
detection-job/d927562638cfa739331a99b3cEXAMPLE",
            "JobName": "example-pii-detection-job-2",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-09T21:20:58.211000+00:00",
            "EndTime": "2023-06-09T21:31:06.027000+00:00",
            "InputDataConfig": {
```

```
                "S3Uri": "s3://amzn-s3-demo-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/111122223333-PII-d927562638cfa739331a99b3cEXAMPLE/output/"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
            "Mode": "ONLY_OFFSETS"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPiiEntitiesDetectionJobs。

`list-sentiment-detection-jobs`

以下程式碼範例顯示如何使用 `list-sentiment-detection-jobs`。

AWS CLI

列出所有情緒偵測任務

下列`list-sentiment-detection-jobs`範例列出所有進行中和已完成的非同步情緒偵測任務。

```
aws comprehend list-sentiment-detection-jobs
```

輸出：

```
{
    "SentimentDetectionJobPropertiesList": [
        {
            "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
            "JobName": "example-sentiment-detection-job",
            "JobStatus": "IN_PROGRESS",
```

```
            "SubmitTime": "2023-06-09T22:42:20.545000+00:00",
            "EndTime": "2023-06-09T22:52:27.416000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/MovieData",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-TS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "123456abcdeb0e11022f22a1EXAMPLE2",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-
detection-job/123456abcdeb0e11022f22a1EXAMPLE2",
            "JobName": "example-sentiment-detection-job-2",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-09T23:16:15.956000+00:00",
            "EndTime": "2023-06-09T23:26:00.168000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/MovieData2",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-TS-123456abcdeb0e11022f22a1EXAMPLE2/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSentimentDetectionJobs。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出資源的標籤

下列list-tags-for-resource範例列出 Amazon Comprehend 資源的標籤。

```
aws comprehend list-tags-for-resource \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1
```

輸出：

```
{
    "ResourceArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1",
    "Tags": [
        {
            "Key": "Department",
            "Value": "Finance"
        },
        {
            "Key": "location",
            "Value": "Seattle"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的標記您的 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

list-targeted-sentiment-detection-jobs

以下程式碼範例顯示如何使用 list-targeted-sentiment-detection-jobs。

AWS CLI

列出所有目標情緒偵測任務

下列list-targeted-sentiment-detection-jobs範例列出所有進行中和已完成的非同步目標情緒偵測任務。

```
aws comprehend list-targeted-sentiment-detection-jobs
```

輸出：

```
{
    "TargetedSentimentDetectionJobPropertiesList": [
        {
            "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
            "JobName": "example-targeted-sentiment-detection-job",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-09T22:42:20.545000+00:00",
            "EndTime": "2023-06-09T22:52:27.416000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/MovieData",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-TS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-IOrole"
        },
        {
            "JobId": "123456abcdeb0e11022f22a1EXAMPLE2",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-
detection-job/123456abcdeb0e11022f22a1EXAMPLE2",
            "JobName": "example-targeted-sentiment-detection-job-2",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-09T23:16:15.956000+00:00",
            "EndTime": "2023-06-09T23:26:00.168000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/MovieData2",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
```

```
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-TS-123456abcdeb0e11022f22a1EXAMPLE2/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTargetedSentimentDetectionJobs。

`list-topics-detection-jobs`

以下程式碼範例顯示如何使用 `list-topics-detection-jobs`。

AWS CLI

列出所有主題偵測任務

下列`list-topics-detection-jobs`範例列出所有進行中和已完成的非同步主題偵測任務。

```
aws comprehend list-topics-detection-jobs
```

輸出：

```
{
    "TopicsDetectionJobPropertiesList": [
        {
            "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
            "JobName" "topic-analysis-1"
            "JobStatus": "IN_PROGRESS",
            "SubmitTime": "2023-06-09T18:40:35.384000+00:00",
            "EndTime": "2023-06-09T18:46:41.936000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
```

```
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
            },
            "NumberOfTopics": 10,
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "123456abcdeb0e11022f22a1EXAMPLE2",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-
job/123456abcdeb0e11022f22a1EXAMPLE2",
            "JobName": "topic-analysis-2",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-09T18:44:43.414000+00:00",
            "EndTime": "2023-06-09T18:50:50.872000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a1EXAMPLE2/output/output.tar.gz"
            },
            "NumberOfTopics": 10,
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "123456abcdeb0e11022f22a1EXAMPLE3",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-
job/123456abcdeb0e11022f22a1EXAMPLE3",
            "JobName": "topic-analysis-2",
            "JobStatus": "IN_PROGRESS",
            "SubmitTime": "2023-06-09T18:50:56.737000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a1EXAMPLE3/output/output.tar.gz"
            },
            "NumberOfTopics": 10,
```

```
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTopicsDetectionJobs。

put-resource-policy

以下程式碼範例顯示如何使用 put-resource-policy。

AWS CLI

連接以資源為基礎的政策

下列put-resource-policy範例會將資源型政策連接至模型，讓另一個 AWS 帳戶可以匯入 。政策會連接到帳戶中的模型，111122223333並允許帳戶444455556666匯入模型。

```
aws comprehend put-resource-policy \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1 \
    --resource-policy '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Action":"comprehend:ImportModel","Resource":"*","Principal":
{"AWS":["arn:aws:iam::444455556666:root"]}}]}'
```

Ouput：

```
{
    "PolicyRevisionId": "aaa111d069d07afaa2aa3106aEXAMPLE"
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的在AWS 帳戶之間複製自訂模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutResourcePolicy。

start-document-classification-job

以下程式碼範例顯示如何使用 start-document-classification-job。

AWS CLI

啟動文件分類任務

下列start-document-classification-job範例會在--input-data-config標籤指定的地址的所有檔案上，使用自訂模型啟動文件分類任務。在此範例中，輸入 S3 儲存貯體包含 SampleSMStext1.txt、 SampleSMStext2.txt和 SampleSMStext3.txt。此模型先前已針對垃圾郵件和非垃圾郵件的文件分類，或「ham」簡訊進行訓練。當任務完成時，output.tar.gz 會放在 --output-data-config標籤指定的位置。 output.tar.gz包含 predictions.jsonl，其中列出每個文件的分類。Json 輸出會列印在每個檔案的一行上，但在此格式化為可讀性。

```
aws comprehend start-document-classification-job \
    --job-name exampleclassificationjob \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket-INPUT/jobdata/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
    --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-
classifier/mymodel/version/12
```

SampleSMStext1.txt 的內容：

```
"CONGRATULATIONS! TXT 2155550100 to win $5000"
```

SampleSMStext2.txt 的內容：

```
"Hi, when do you want me to pick you up from practice?"
```

SampleSMStext3.txt 的內容：

```
"Plz send bank account # to 2155550100 to claim prize!!"
```

輸出：

```
{
    "JobId": "e758dd56b824aa717ceab551fEXAMPLE",
```

```
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:document-classification-
job/e758dd56b824aa717ceab551fEXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

predictions.jsonl 的內容：

```
{"File": "SampleSMSText1.txt", "Line": "0", "Classes": [{"Name": "spam", "Score":
 0.9999}, {"Name": "ham", "Score": 0.0001}]}
{"File": "SampleSMStext2.txt", "Line": "0", "Classes": [{"Name": "ham", "Score":
 0.9994}, {"Name": "spam", "Score": 0.0006}]}
{"File": "SampleSMSText3.txt", "Line": "0", "Classes": [{"Name": "spam", "Score":
 0.9999}, {"Name": "ham", "Score": 0.0001}]}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的自訂分類。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartDocumentClassificationJob。

start-dominant-language-detection-job

以下程式碼範例顯示如何使用 start-dominant-language-detection-job。

AWS CLI

啟動非同步語言偵測任務

下列start-dominant-language-detection-job範例會針對位於 --input-data-config標籤所指定地址的所有檔案，啟動非同步語言偵測任務。此範例中的 S3 儲存貯體包含 Sampletext1.txt。當任務完成時，資料夾 output會放置在 --output-data-config標籤指定的位置。資料夾包含output.txt每個文字檔案的主要語言，以及每個預測的預先訓練模型可信度分數。

```
aws comprehend start-dominant-language-detection-job \
    --job-name example_language_analysis_job \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
    --language-code en
```

Sampletext1.txt 的內容：

```
"Physics is the natural science that involves the study of matter and its motion and
 behavior through space and time, along with related concepts such as energy and
 force."
```

輸出：

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

output.txt 的內容：

```
{"File": "Sampletext1.txt", "Languages": [{"LanguageCode": "en", "Score":
 0.9913753867149353}], "Line": 0}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartDominantLanguageDetectionJob。

start-entities-detection-job

以下程式碼範例顯示如何使用 start-entities-detection-job。

AWS CLI

範例 1：使用預先訓練的模型啟動標準實體偵測任務

下列start-entities-detection-job範例會針對位於 --input-data-config標籤所指定地址的所有檔案，啟動非同步實體偵測任務。此範例中的 S3 儲存貯體包含 Sampletext1.txt、Sampletext2.txt和 Sampletext3.txt。當任務完成時，資料夾 output會放置在 --output-data-config標籤指定的位置。資料夾包含output.txt列出每個文字檔案中偵測到的所有具名實體，以及每個預測的預先訓練模型可信度分數。Json 輸出會列印在每個輸入檔案的一行上，但在此格式化為可讀性。

```
aws comprehend start-entities-detection-job \
    --job-name entitiestest \
```

```
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
    --language-code en
```

Sampletext1.txt 的內容：

```
"Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card
 account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July
 31st."
```

Sampletext2.txt 的內容：

```
"Dear Max, based on your autopay settings for your account example1.org account, we
 will withdraw your payment on the due date from your bank account number XXXXXX1111
 with the routing number XXXXX0000. "
```

Sampletext3.txt 的內容：

```
"Jane, please submit any customer feedback from this weekend to AnySpa, 123 Main St,
 Anywhere and send comments to Alice at AnySpa@example.com."
```

輸出：

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

output.txt 具有行縮排的內容可讀性：

```
{
"Entities": [
    {
    "BeginOffset": 6,
    "EndOffset": 15,
    "Score": 0.9994006636420306,
```

```
    "Text": "Zhang Wei",
    "Type": "PERSON"
    },
    {
    "BeginOffset": 22,
    "EndOffset": 26,
    "Score": 0.9976647915128143,
    "Text": "John",
    "Type": "PERSON"
    },
    {
    "BeginOffset": 33,
    "EndOffset": 67,
    "Score": 0.9984608700836206,
    "Text": "AnyCompany Financial Services, LLC",
    "Type": "ORGANIZATION"
    },
    {
    "BeginOffset": 88,
    "EndOffset": 107,
    "Score": 0.9868521019555556,
    "Text": "1111-XXXX-1111-XXXX",
    "Type": "OTHER"
    },
    {
    "BeginOffset": 133,
    "EndOffset": 139,
    "Score": 0.998242565709204,
    "Text": "$24.53",
    "Type": "QUANTITY"
    },
    {
    "BeginOffset": 155,
    "EndOffset": 164,
    "Score": 0.9993039263159287,
    "Text": "July 31st",
    "Type": "DATE"
    }
],
"File": "SampleText1.txt",
"Line": 0
}
{
"Entities": [
```

```
    {
    "BeginOffset": 5,
    "EndOffset": 8,
    "Score": 0.9866232147545232,
    "Text": "Max",
    "Type": "PERSON"
    },
    {
    "BeginOffset": 156,
    "EndOffset": 166,
    "Score": 0.9797723450933329,
    "Text": "XXXXXX1111",
    "Type": "OTHER"
    },
    {
    "BeginOffset": 191,
    "EndOffset": 200,
    "Score": 0.9247838572396843,
    "Text": "XXXXX0000",
    "Type": "OTHER"
    }
],
"File": "SampleText2.txt",
"Line": 0
}
{
 "Entities": [
    {
    "Score": 0.9990532994270325,
    "Type": "PERSON",
    "Text": "Jane",
    "BeginOffset": 0,
    "EndOffset": 4
    },
    {
    "Score": 0.9519651532173157,
    "Type": "DATE",
    "Text": "this weekend",
    "BeginOffset": 47,
    "EndOffset": 59
    },
    {
    "Score": 0.5566426515579224,
    "Type": "ORGANIZATION",
```

```
    "Text": "AnySpa",
    "BeginOffset": 63,
    "EndOffset": 69
    },
    {
    "Score": 0.8059805631637573,
    "Type": "LOCATION",
    "Text": "123 Main St, Anywhere",
    "BeginOffset": 71,
    "EndOffset": 92
    },
    {
    "Score": 0.998830258846283,
    "Type": "PERSON",
    "Text": "Alice",
    "BeginOffset": 114,
    "EndOffset": 119
    },
    {
    "Score": 0.997818112373352,
    "Type": "OTHER",
    "Text": "AnySpa@example.com",
    "BeginOffset": 123,
    "EndOffset": 138
    }
    ],
    "File": "SampleText3.txt",
    "Line": 0
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

範例 2：啟動自訂實體偵測任務

下列start-entities-detection-job範例會針對位於 --input-data-config標籤所指定地址的所有檔案，啟動非同步自訂實體偵測任務。在此範例中，此範例中的 S3 儲存貯體包含 SampleFeedback1.txt、 SampleFeedback2.txt和 SampleFeedback3.txt。實體辨識器模型已透過客戶支援意見回饋進行訓練，以辨識裝置名稱。當任務完成時， 資料夾 output會放在 --output-data-config標籤指定的位置。資料夾包含 output.txt，列出每個文字檔案中偵測到的所有具名實體，以及每個預測的預先訓練模型可信度分數。Json 輸出會列印在每個檔案的一行上，但在此格式化為可讀性。

```
aws comprehend start-entities-detection-job \
    --job-name customentitiestest \
    --entity-recognizer-arn "arn:aws:comprehend:us-west-2:111122223333:entity-
recognizer/entityrecognizer" \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/jobdata/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-IOrole"
```

SampleFeedback1.txt 的內容：

```
"I've been on the AnyPhone app have had issues for 24 hours when trying to pay bill.
 Cannot make payment. Sigh. | Oh man! Lets get that app up and running. DM me, and
 we can get to work!"
```

SampleFeedback2.txt 的內容：

```
"Hi, I have a discrepancy with my new bill. Could we get it sorted out? A rep added
 stuff I didnt sign up for when I did my AnyPhone 10 upgrade. | We can absolutely
 get this sorted!"
```

SampleFeedback3.txt 的內容：

```
"Is the by 1 get 1 free AnySmartPhone promo still going on? | Hi Christian! It ended
 yesterday, send us a DM if you have any questions and we can take a look at your
 options!"
```

輸出：

```
{
    "JobId": "019ea9edac758806850fa8a79ff83021",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/019ea9edac758806850fa8a79ff83021",
    "JobStatus": "SUBMITTED"
}
```

output.txt 具有行縮排的內容可讀性：

```
{
```

```
"Entities": [
    {
    "BeginOffset": 17,
    "EndOffset": 25,
    "Score": 0.9999728210205924,
    "Text": "AnyPhone",
    "Type": "DEVICE"
    }
],
"File": "SampleFeedback1.txt",
"Line": 0
}
{
"Entities": [
    {
    "BeginOffset": 123,
    "EndOffset": 133,
    "Score": 0.9999892116761524,
    "Text": "AnyPhone 10",
    "Type": "DEVICE"
    }
],
"File": "SampleFeedback2.txt",
"Line": 0
}
{
"Entities": [
    {
    "BeginOffset": 23,
    "EndOffset": 35,
    "Score": 0.9999971389852362,
    "Text": "AnySmartPhone",
    "Type": "DEVICE"
    }
],
"File": "SampleFeedback3.txt",
"Line": 0
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的自訂實體辨識。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartEntitiesDetectionJob。

start-events-detection-job

以下程式碼範例顯示如何使用 start-events-detection-job。

AWS CLI

啟動非同步事件偵測任務

下列start-events-detection-job範例會針對位於 --input-data-config標籤所指定地址的所有檔案，啟動非同步事件偵測任務。可能的目標事件類型包括 BANKRUPCTY、EMPLOYMENT、CORPORATE_ACQUISITION、INVESTMENT_GENERAL、CORPORATE_MERGER、IPO、RIGHTS_ISSUE、SECONDARY_OFFERING、SHELF_OFFERING、TENDER_OFFERING和 STOCK_SPLIT。此範例中的 S3 儲存貯體包含 SampleText1.txt、SampleText2.txt和 SampleText3.txt。當任務完成時，資料夾 output會放置在 --output-data-config標籤指定的位置。資料夾包含 SampleText1.txt.out、SampleText2.txt.out和 SampleText3.txt.out。JSON 輸出會列印在每個檔案的一行上，但在此格式化為可讀性。

```
aws comprehend start-events-detection-job \
    --job-name events-detection-1 \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/EventsData" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole \
    --language-code en \
    --target-event-
types "BANKRUPTCY" "EMPLOYMENT" "CORPORATE_ACQUISITION" "CORPORATE_MERGER" "INVESTMENT_GENER
```

SampleText1.txt 的內容：

```
"Company AnyCompany grew by increasing sales and through acquisitions. After
 purchasing competing firms in 2020, AnyBusiness, a part of the AnyBusinessGroup,
 gave Jane Does firm a going rate of one cent a gallon or forty-two cents a barrel."
```

SampleText2.txt 的內容：

```
"In 2021, AnyCompany officially purchased AnyBusiness for 100 billion dollars,
 surprising and exciting the shareholders."
```

SampleText3.txt 的內容：

```
"In 2022, AnyCompany stock crashed 50. Eventually later that year they filed for
 bankruptcy."
```

輸出：

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:events-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

SampleText1.txt.out 具有行縮排的內容可讀性：

```
{
    "Entities": [
        {
        "Mentions": [
            {
            "BeginOffset": 8,
            "EndOffset": 18,
            "Score": 0.99977,
            "Text": "AnyCompany",
            "Type": "ORGANIZATION",
            "GroupScore": 1
            },
            {
            "BeginOffset": 112,
            "EndOffset": 123,
            "Score": 0.999747,
            "Text": "AnyBusiness",
            "Type": "ORGANIZATION",
            "GroupScore": 0.979826
            },
            {
            "BeginOffset": 171,
            "EndOffset": 175,
            "Score": 0.999615,
            "Text": "firm",
            "Type": "ORGANIZATION",
            "GroupScore": 0.871647
            }
```

```
        ]
        },
        {
        "Mentions": [
            {
            "BeginOffset": 97,
            "EndOffset": 102,
            "Score": 0.987687,
            "Text": "firms",
            "Type": "ORGANIZATION",
            "GroupScore": 1
            }
        ]
        },
        {
        "Mentions": [
            {
            "BeginOffset": 103,
            "EndOffset": 110,
            "Score": 0.999458,
            "Text": "in 2020",
            "Type": "DATE",
            "GroupScore": 1
            }
        ]
        },
        {
        "Mentions": [
            {
            "BeginOffset": 160,
            "EndOffset": 168,
            "Score": 0.999649,
            "Text": "John Doe",
            "Type": "PERSON",
            "GroupScore": 1
            }
        ]
        }
    ],
    "Events": [
        {
        "Type": "CORPORATE_ACQUISITION",
        "Arguments": [
            {
```

```
            "EntityIndex": 0,
            "Role": "INVESTOR",
            "Score": 0.99977
            }
        ],
        "Triggers": [
            {
            "BeginOffset": 56,
            "EndOffset": 68,
            "Score": 0.999967,
            "Text": "acquisitions",
            "Type": "CORPORATE_ACQUISITION",
            "GroupScore": 1
            }
        ]
        },
        {
        "Type": "CORPORATE_ACQUISITION",
        "Arguments": [
            {
            "EntityIndex": 1,
            "Role": "INVESTEE",
            "Score": 0.987687
            },
            {
            "EntityIndex": 2,
            "Role": "DATE",
            "Score": 0.999458
            },
            {
            "EntityIndex": 3,
            "Role": "INVESTOR",
            "Score": 0.999649
            }
        ],
        "Triggers": [
            {
            "BeginOffset": 76,
            "EndOffset": 86,
            "Score": 0.999973,
            "Text": "purchasing",
            "Type": "CORPORATE_ACQUISITION",
            "GroupScore": 1
            }
```

```
        ]
        }
    ],
    "File": "SampleText1.txt",
    "Line": 0
}
```

SampleText2.txt.out 的內容：

```
{
    "Entities": [
        {
        "Mentions": [
            {
            "BeginOffset": 0,
            "EndOffset": 7,
            "Score": 0.999473,
            "Text": "In 2021",
            "Type": "DATE",
            "GroupScore": 1
            }
        ]
        },
        {
        "Mentions": [
            {
            "BeginOffset": 9,
            "EndOffset": 19,
            "Score": 0.999636,
            "Text": "AnyCompany",
            "Type": "ORGANIZATION",
            "GroupScore": 1
            }
        ]
        },
        {
        "Mentions": [
            {
            "BeginOffset": 45,
            "EndOffset": 56,
            "Score": 0.999712,
            "Text": "AnyBusiness",
            "Type": "ORGANIZATION",
```

```
            "GroupScore": 1
            }
        ]
        },
        {
        "Mentions": [
            {
            "BeginOffset": 61,
            "EndOffset": 80,
            "Score": 0.998886,
            "Text": "100 billion dollars",
            "Type": "MONETARY_VALUE",
            "GroupScore": 1
            }
        ]
        }
    ],
    "Events": [
        {
        "Type": "CORPORATE_ACQUISITION",
        "Arguments": [
            {
            "EntityIndex": 3,
            "Role": "AMOUNT",
            "Score": 0.998886
            },
            {
            "EntityIndex": 2,
            "Role": "INVESTEE",
            "Score": 0.999712
            },
            {
            "EntityIndex": 0,
            "Role": "DATE",
            "Score": 0.999473
            },
            {
            "EntityIndex": 1,
            "Role": "INVESTOR",
            "Score": 0.999636
            }
        ],
        "Triggers": [
            {
```

```
            "BeginOffset": 31,
            "EndOffset": 40,
            "Score": 0.99995,
            "Text": "purchased",
            "Type": "CORPORATE_ACQUISITION",
            "GroupScore": 1
            }
        ]
        }
    ],
    "File": "SampleText2.txt",
    "Line": 0
}
```

SampleText3.txt.out 的內容：

```
{
    "Entities": [
        {
        "Mentions": [
            {
            "BeginOffset": 9,
            "EndOffset": 19,
            "Score": 0.999774,
            "Text": "AnyCompany",
            "Type": "ORGANIZATION",
            "GroupScore": 1
            },
            {
            "BeginOffset": 66,
            "EndOffset": 70,
            "Score": 0.995717,
            "Text": "they",
            "Type": "ORGANIZATION",
            "GroupScore": 0.997626
            }
        ]
        },
        {
        "Mentions": [
            {
            "BeginOffset": 50,
            "EndOffset": 65,
```

```
            "Score": 0.999656,
            "Text": "later that year",
            "Type": "DATE",
            "GroupScore": 1
            }
        ]
        }
    ],
    "Events": [
        {
        "Type": "BANKRUPTCY",
        "Arguments": [
            {
            "EntityIndex": 1,
            "Role": "DATE",
            "Score": 0.999656
            },
            {
            "EntityIndex": 0,
            "Role": "FILER",
            "Score": 0.995717
            }
        ],
        "Triggers": [
            {
            "BeginOffset": 81,
            "EndOffset": 91,
            "Score": 0.999936,
            "Text": "bankruptcy",
            "Type": "BANKRUPTCY",
            "GroupScore": 1
            }
        ]
        }
    ],
    "File": "SampleText3.txt",
    "Line": 0
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartEventsDetectionJob。

start-flywheel-iteration

以下程式碼範例顯示如何使用 start-flywheel-iteration。

AWS CLI

啟動飛輪反覆運算

下列start-flywheel-iteration範例會啟動飛輪反覆運算。此操作使用飛輪中的任何新資料集來訓練新的模型版本。

```
aws comprehend start-flywheel-iteration \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel
```

輸出：

```
{
    "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel",
    "FlywheelIterationId": "12345123TEXAMPLE"
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的飛輪概觀。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartFlywheelIteration。

start-key-phrases-detection-job

以下程式碼範例顯示如何使用 start-key-phrases-detection-job。

AWS CLI

啟動金鑰片語偵測任務

下列start-key-phrases-detection-job範例會針對位於 --input-data-config標籤所指定地址的所有檔案，啟動非同步金鑰片語偵測任務。此範例中的 S3 儲存貯體包含 Sampletext1.txt、 Sampletext2.txt和 Sampletext3.txt。任務完成時，資料夾 output會放置在 --output-data-config標籤指定的位置。資料夾包含 檔案output.txt，其中包含每個文字檔案中偵測到的所有金鑰片語，以及每個預測的預先訓練模型可信度分數。Json 輸出會列印在每個檔案的一行上，但在此格式化為可讀性。

```
aws comprehend start-key-phrases-detection-job \
    --job-name keyphrasesanalysistest1 \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role" \
    --language-code en
```

Sampletext1.txt 的內容：

```
"Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card
 account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July
 31st."
```

Sampletext2.txt 的內容：

```
"Dear Max, based on your autopay settings for your account Internet.org account, we
 will withdraw your payment on the due date from your bank account number XXXXXX1111
 with the routing number XXXXX0000. "
```

Sampletext3.txt 的內容：

```
"Jane, please submit any customer feedback from this weekend to Sunshine Spa, 123
 Main St, Anywhere and send comments to Alice at AnySpa@example.com."
```

輸出：

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

output.txt 具有行縮排的內容以供讀取：

```
{
    "File": "SampleText1.txt",
    "KeyPhrases": [
        {
```

```
        "BeginOffset": 6,
        "EndOffset": 15,
        "Score": 0.9748965572679326,
        "Text": "Zhang Wei"
        },
        {
        "BeginOffset": 22,
        "EndOffset": 26,
        "Score": 0.9997344722354619,
        "Text": "John"
        },
        {
        "BeginOffset": 28,
        "EndOffset": 62,
        "Score": 0.9843791074032948,
        "Text": "Your AnyCompany Financial Services"
        },
        {
        "BeginOffset": 64,
        "EndOffset": 107,
        "Score": 0.8976122401721824,
        "Text": "LLC credit card account 1111-XXXX-1111-XXXX"
        },
        {
        "BeginOffset": 112,
        "EndOffset": 129,
        "Score": 0.9999612982629748,
        "Text": "a minimum payment"
        },
        {
        "BeginOffset": 133,
        "EndOffset": 139,
        "Score": 0.99975728947036,
        "Text": "$24.53"
        },
        {
        "BeginOffset": 155,
        "EndOffset": 164,
        "Score": 0.9940866241449973,
        "Text": "July 31st"
        }
    ],
    "Line": 0
    }
```

```
    {
    "File": "SampleText2.txt",
    "KeyPhrases": [
        {
        "BeginOffset": 0,
        "EndOffset": 8,
        "Score": 0.9974021100118472,
        "Text": "Dear Max"
        },
        {
        "BeginOffset": 19,
        "EndOffset": 40,
        "Score": 0.9961120519515884,
        "Text": "your autopay settings"
        },
        {
        "BeginOffset": 45,
        "EndOffset": 78,
        "Score": 0.9980620070116009,
        "Text": "your account Internet.org account"
        },
        {
        "BeginOffset": 97,
        "EndOffset": 109,
        "Score": 0.999919660140754,
        "Text": "your payment"
        },
        {
        "BeginOffset": 113,
        "EndOffset": 125,
        "Score": 0.9998370719754205,
        "Text": "the due date"
        },
        {
        "BeginOffset": 131,
        "EndOffset": 166,
        "Score": 0.9955068678502509,
        "Text": "your bank account number XXXXXX1111"
        },
        {
        "BeginOffset": 172,
        "EndOffset": 200,
        "Score": 0.8653433315829526,
        "Text": "the routing number XXXXX0000"
```

```
        }
    ],
    "Line": 0
    }
    {
    "File": "SampleText3.txt",
    "KeyPhrases": [
        {
        "BeginOffset": 0,
        "EndOffset": 4,
        "Score": 0.9142947833681668,
        "Text": "Jane"
        },
        {
        "BeginOffset": 20,
        "EndOffset": 41,
        "Score": 0.9984325676596763,
        "Text": "any customer feedback"
        },
        {
        "BeginOffset": 47,
        "EndOffset": 59,
        "Score": 0.9998782448150636,
        "Text": "this weekend"
        },
        {
        "BeginOffset": 63,
        "EndOffset": 75,
        "Score": 0.99866741830757,
        "Text": "Sunshine Spa"
        },
        {
        "BeginOffset": 77,
        "EndOffset": 88,
        "Score": 0.9695803485466054,
        "Text": "123 Main St"
        },
        {
        "BeginOffset": 108,
        "EndOffset": 116,
        "Score": 0.9997065928550928,
        "Text": "comments"
        },
        {
```

```
        "BeginOffset": 120,
        "EndOffset": 125,
        "Score": 0.9993466833825161,
        "Text": "Alice"
        },
        {
        "BeginOffset": 129,
        "EndOffset": 144,
        "Score": 0.9654563612885667,
        "Text": "AnySpa@example.com"
        }
    ],
    "Line": 0
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartKeyPhrasesDetectionJob。

start-pii-entities-detection-job

以下程式碼範例顯示如何使用 start-pii-entities-detection-job。

AWS CLI

啟動非同步 PII 偵測任務

下列start-pii-entities-detection-job範例會針對位於 --input-data-config標籤所指定地址的所有檔案，啟動非同步個人身分識別資訊 (PII) 實體偵測任務。此範例中的 S3 儲存貯體包含 Sampletext1.txt、 Sampletext2.txt和 Sampletext3.txt。當任務完成時，資料夾 output會放置在 --output-data-config標籤指定的位置。資料夾包含 SampleText1.txt.out、 SampleText2.txt.out和 SampleText3.txt.out，列出每個文字檔案中的具名實體。Json 輸出會列印在每個檔案的一行上，但在此格式化為可讀性。

```
aws comprehend start-pii-entities-detection-job \
    --job-name entities_test \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
```

```
    --language-code en \
    --mode ONLY_OFFSETS
```

Sampletext1.txt 的內容：

```
"Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card
 account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July
 31st."
```

Sampletext2.txt 的內容：

```
"Dear Max, based on your autopay settings for your account Internet.org account, we
 will withdraw your payment on the due date from your bank account number XXXXXX1111
 with the routing number XXXXX0000. "
```

Sampletext3.txt 的內容：

```
"Jane, please submit any customer feedback from this weekend to Sunshine Spa, 123
 Main St, Anywhere and send comments to Alice at AnySpa@example.com."
```

輸出：

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

SampleText1.txt.out 具有行縮排的內容可讀性：

```
{
    "Entities": [
        {
        "BeginOffset": 6,
        "EndOffset": 15,
        "Type": "NAME",
        "Score": 0.9998490510222595
        },
        {
```

```
        "BeginOffset": 22,
        "EndOffset": 26,
        "Type": "NAME",
        "Score": 0.9998937958019426
        },
        {
        "BeginOffset": 88,
        "EndOffset": 107,
        "Type": "CREDIT_DEBIT_NUMBER",
        "Score": 0.9554297245278491
        },
        {
        "BeginOffset": 155,
        "EndOffset": 164,
        "Type": "DATE_TIME",
        "Score": 0.9999720462925257
        }
    ],
    "File": "SampleText1.txt",
    "Line": 0
}
```

SampleText2.txt.out 具有行縮排的內容可讀性：

```
{
    "Entities": [
        {
        "BeginOffset": 5,
        "EndOffset": 8,
        "Type": "NAME",
        "Score": 0.9994390774924007
        },
        {
        "BeginOffset": 58,
        "EndOffset": 70,
        "Type": "URL",
        "Score": 0.9999958276922101
        },
        {
        "BeginOffset": 156,
        "EndOffset": 166,
        "Type": "BANK_ACCOUNT_NUMBER",
        "Score": 0.9999721058045592
```

```
        },
        {
        "BeginOffset": 191,
        "EndOffset": 200,
        "Type": "BANK_ROUTING",
        "Score": 0.9998968945989909
        }
    ],
    "File": "SampleText2.txt",
    "Line": 0
}
```

SampleText3.txt.out 具有行縮排的內容可讀性：

```
{
    "Entities": [
        {
        "BeginOffset": 0,
        "EndOffset": 4,
        "Type": "NAME",
        "Score": 0.999949934606805
        },
        {
        "BeginOffset": 77,
        "EndOffset": 88,
        "Type": "ADDRESS",
        "Score": 0.9999035300466904
        },
        {
        "BeginOffset": 120,
        "EndOffset": 125,
        "Type": "NAME",
        "Score": 0.9998203838716296
        },
        {
        "BeginOffset": 129,
        "EndOffset": 144,
        "Type": "EMAIL",
        "Score": 0.9998313473105228
        }
    ],
    "File": "SampleText3.txt",
    "Line": 0
```

```
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartPiiEntitiesDetectionJob。

start-sentiment-detection-job

以下程式碼範例顯示如何使用 start-sentiment-detection-job。

AWS CLI

啟動非同步情緒分析任務

下列start-sentiment-detection-job範例會針對位於 --input-data-config標籤所指定地址的所有檔案，啟動非同步情緒分析偵測任務。此範例中的 S3 儲存貯體資料夾包含 SampleMovieReview1.txt、 SampleMovieReview2.txt和 SampleMovieReview3.txt。當任務完成時，資料夾 output會放置在 --output-data-config標籤指定的位置。資料夾包含檔案 output.txt，其中包含每個文字檔案的慣用情緒，以及每個預測的預先訓練模型可信度分數。Json 輸出會列印在每個檔案的一行上，但在此格式化為可讀性。

```
aws comprehend start-sentiment-detection-job \
    --job-name example-sentiment-detection-job \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/MovieData" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role
```

SampleMovieReview1.txt 的內容：

```
"The film, AnyMovie2, is fairly predictable and just okay."
```

SampleMovieReview2.txt 的內容：

```
"AnyMovie2 is the essential sci-fi film that I grew up watching when I was a kid. I
 highly recommend this movie."
```

SampleMovieReview3.txt 的內容：

```
"Don't get fooled by the 'awards' for AnyMovie2. All parts of the film were poorly
 stolen from other modern directors."
```

輸出：

```
{
    "JobId": "0b5001e25f62ebb40631a9a1a7fde7b3",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-detection-
job/0b5001e25f62ebb40631a9a1a7fde7b3",
    "JobStatus": "SUBMITTED"
}
```

output.txt 具有可讀性縮排的 內容：

```
{
    "File": "SampleMovieReview1.txt",
        "Line": 0,
        "Sentiment": "MIXED",
        "SentimentScore": {
            "Mixed": 0.6591159105300903,
            "Negative": 0.26492202281951904,
            "Neutral": 0.035430654883384705,
            "Positive": 0.04053137078881264
            }
        }
    {
    "File": "SampleMovieReview2.txt",
        "Line": 0,
        "Sentiment": "POSITIVE",
        "SentimentScore": {
            "Mixed": 0.00000871846623340388,
            "Negative": 0.00006134175055194646,
            "Neutral": 0.0002941041602753103,
            "Positive": 0.9996358156204224
            }
        }
    {
    "File": "SampleMovieReview3.txt",
        "Line": 0,
        "Sentiment": "NEGATIVE",
        "SentimentScore": {
            "Mixed": 0.004146667663007975,
```

```
            "Negative": 0.9645107984542847,
            "Neutral": 0.016559595242142677,
            "Positive": 0.014782938174903393
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartSentimentDetectionJob。

start-targeted-sentiment-detection-job

以下程式碼範例顯示如何使用 start-targeted-sentiment-detection-job。

AWS CLI

啟動非同步目標情緒分析任務

下列start-targeted-sentiment-detection-job範例會針對位於 --input-data-config標籤所指定地址的所有檔案，啟動非同步目標情緒分析偵測任務。此範例中的 S3 儲存貯體資料夾包含 SampleMovieReview1.txt、 SampleMovieReview2.txt和 SampleMovieReview3.txt。當任務完成時， output.tar.gz 會放置在 --output-data-config標籤指定的位置。 output.tar.gz包含檔案 SampleMovieReview1.txt.out、SampleMovieReview2.txt.out和 SampleMovieReview3.txt.out，每個檔案都包含單一輸入文字檔案的所有具名實體和相關聯的情緒。

```
aws comprehend start-targeted-sentiment-detection-job \
    --job-name targeted_movie_review_analysis1 \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/MovieData" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role
```

SampleMovieReview1.txt 的內容：

```
"The film, AnyMovie, is fairly predictable and just okay."
```

SampleMovieReview2.txt 的內容：

```
"AnyMovie is the essential sci-fi film that I grew up watching when I was a kid. I
 highly recommend this movie."
```

SampleMovieReview3.txt 的內容：

```
"Don't get fooled by the 'awards' for AnyMovie. All parts of the film were poorly
 stolen from other modern directors."
```

輸出：

```
{
    "JobId": "0b5001e25f62ebb40631a9a1a7fde7b3",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-
detection-job/0b5001e25f62ebb40631a9a1a7fde7b3",
    "JobStatus": "SUBMITTED"
}
```

SampleMovieReview1.txt.out 具有行縮排的內容可讀性：

```
{
    "Entities": [
        {
        "DescriptiveMentionIndex": [
            0
        ],
        "Mentions": [
            {
            "BeginOffset": 4,
            "EndOffset": 8,
            "Score": 0.994972,
            "GroupScore": 1,
            "Text": "film",
            "Type": "MOVIE",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 1,
                "Positive": 0
                }
```

```
            }
            }
        ]
        },
        {
        "DescriptiveMentionIndex": [
            0
        ],
        "Mentions": [
            {
            "BeginOffset": 10,
            "EndOffset": 18,
            "Score": 0.631368,
            "GroupScore": 1,
            "Text": "AnyMovie",
            "Type": "ORGANIZATION",
            "MentionSentiment": {
                "Sentiment": "POSITIVE",
                "SentimentScore": {
                "Mixed": 0.001729,
                "Negative": 0.000001,
                "Neutral": 0.000318,
                "Positive": 0.997952
                }
            }
            }
        ]
        }
    ],
    "File": "SampleMovieReview1.txt",
    "Line": 0
}
```

可供讀取的換SampleMovieReview2.txt.out行內容：

```
{
    "Entities": [
        {
        "DescriptiveMentionIndex": [
            0
        ],
        "Mentions": [
            {
```

```
            "BeginOffset": 0,
            "EndOffset": 8,
            "Score": 0.854024,
            "GroupScore": 1,
            "Text": "AnyMovie",
            "Type": "MOVIE",
            "MentionSentiment": {
                "Sentiment": "POSITIVE",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 0.000007,
                "Positive": 0.999993
                }
            }
            },
            {
            "BeginOffset": 104,
            "EndOffset": 109,
            "Score": 0.999129,
            "GroupScore": 0.502937,
            "Text": "movie",
            "Type": "MOVIE",
            "MentionSentiment": {
                "Sentiment": "POSITIVE",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 0,
                "Positive": 1
                }
            }
            },
            {
            "BeginOffset": 33,
            "EndOffset": 37,
            "Score": 0.999823,
            "GroupScore": 0.999252,
            "Text": "film",
            "Type": "MOVIE",
            "MentionSentiment": {
                "Sentiment": "POSITIVE",
                "SentimentScore": {
                "Mixed": 0,
```

```
                "Negative": 0,
                "Neutral": 0.000001,
                "Positive": 0.999999
                }
            }
            }
        ]
        },
        {
        "DescriptiveMentionIndex": [
            0,
            1,
            2
        ],
        "Mentions": [
            {
            "BeginOffset": 43,
            "EndOffset": 44,
            "Score": 0.999997,
            "GroupScore": 1,
            "Text": "I",
            "Type": "PERSON",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 1,
                "Positive": 0
                }
            }
            },
            {
            "BeginOffset": 80,
            "EndOffset": 81,
            "Score": 0.999996,
            "GroupScore": 0.52523,
            "Text": "I",
            "Type": "PERSON",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
```

```
                "Neutral": 1,
                "Positive": 0
                }
            }
            },
            {
            "BeginOffset": 67,
            "EndOffset": 68,
            "Score": 0.999994,
            "GroupScore": 0.999499,
            "Text": "I",
            "Type": "PERSON",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 1,
                "Positive": 0
                }
            }
            }
        ]
        },
        {
        "DescriptiveMentionIndex": [
            0
        ],
        "Mentions": [
            {
            "BeginOffset": 75,
            "EndOffset": 78,
            "Score": 0.999978,
            "GroupScore": 1,
            "Text": "kid",
            "Type": "PERSON",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 1,
                "Positive": 0
                }
```

```
            }
            }
        ]
        }
    ],
    "File": "SampleMovieReview2.txt",
    "Line": 0
}
```

SampleMovieReview3.txt.out 具有行縮排的內容以供讀取：

```
{
    "Entities": [
        {
        "DescriptiveMentionIndex": [
            1
        ],
        "Mentions": [
            {
            "BeginOffset": 64,
            "EndOffset": 68,
            "Score": 0.992953,
            "GroupScore": 0.999814,
            "Text": "film",
            "Type": "MOVIE",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0.000004,
                "Negative": 0.010425,
                "Neutral": 0.989543,
                "Positive": 0.000027
                }
            }
            },
            {
            "BeginOffset": 37,
            "EndOffset": 45,
            "Score": 0.999782,
            "GroupScore": 1,
            "Text": "AnyMovie",
            "Type": "ORGANIZATION",
            "MentionSentiment": {
```

```
                "Sentiment": "POSITIVE",
                "SentimentScore": {
                "Mixed": 0.000095,
                "Negative": 0.039847,
                "Neutral": 0.000673,
                "Positive": 0.959384
                }
            }
            }
        ]
        },
        {
        "DescriptiveMentionIndex": [
            0
        ],
        "Mentions": [
            {
            "BeginOffset": 47,
            "EndOffset": 50,
            "Score": 0.999991,
            "GroupScore": 1,
            "Text": "All",
            "Type": "QUANTITY",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0.000001,
                "Negative": 0.000001,
                "Neutral": 0.999998,
                "Positive": 0
                }
            }
            }
        ]
        },
        {
        "DescriptiveMentionIndex": [
            0
        ],
        "Mentions": [
            {
            "BeginOffset": 106,
            "EndOffset": 115,
            "Score": 0.542083,
```

```
            "GroupScore": 1,
            "Text": "directors",
            "Type": "PERSON",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 1,
                "Positive": 0
                }
            }
            }
        ]
        }
    ],
    "File": "SampleMovieReview3.txt",
    "Line": 0
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartTargetedSentimentDetectionJob。

start-topics-detection-job

以下程式碼範例顯示如何使用 start-topics-detection-job。

AWS CLI

啟動主題偵測分析任務

下列start-topics-detection-job範例會針對位於 --input-data-config標籤所指定地址的所有檔案，啟動非同步主題偵測任務。當任務完成時， 資料夾 output會放置在 --ouput-data-config標籤指定的位置。 output包含 topic-terms.csv 和 doc-topics.csv。第一個輸出檔案 topic-terms.csv 是集合中的主題清單。根據預設，對於每個主題，清單會包含根據其權重按主題排列的熱門詞彙。第二個檔案 列出與主題相關聯的文件doc-topics.csv，以及與該主題相關的文件比例。

```
aws comprehend start-topics-detection-job \
```

```
    --job-name example_topics_detection_job \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
    --language-code en
```

輸出：

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的主題建模。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartTopicsDetectionJob。

stop-dominant-language-detection-job

以下程式碼範例顯示如何使用 stop-dominant-language-detection-job。

AWS CLI

停止非同步主要語言偵測任務

下列stop-dominant-language-detection-job範例會停止進行中的非同步主要語言偵測任務。如果目前的任務狀態為IN_PROGRESS任務標記為終止並進入 STOP_REQUESTED 狀態。如果任務在停止之前完成，則會進入 COMPLETED 狀態。

```
aws comprehend stop-dominant-language-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

輸出：

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
```

```
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopDominantLanguageDetectionJob。

stop-entities-detection-job

以下程式碼範例顯示如何使用 stop-entities-detection-job。

AWS CLI

停止非同步實體偵測任務

下列stop-entities-detection-job範例會停止進行中的非同步實體偵測任務。如果目前的任務狀態為IN_PROGRESS任務標記為終止並進入 STOP_REQUESTED 狀態。如果任務在停止之前完成，則會進入 COMPLETED 狀態。

```
aws comprehend stop-entities-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

輸出：

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopEntitiesDetectionJob。

stop-events-detection-job

以下程式碼範例顯示如何使用 stop-events-detection-job。

AWS CLI

停止非同步事件偵測任務

下列stop-events-detection-job範例會停止進行中的非同步事件偵測任務。如果目前的任務狀態為IN_PROGRESS任務標記為終止並進入 STOP_REQUESTED 狀態。如果任務在停止之前完成，則會進入 COMPLETED 狀態。

```
aws comprehend stop-events-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

輸出：

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopEventsDetectionJob。

stop-key-phrases-detection-job

以下程式碼範例顯示如何使用 stop-key-phrases-detection-job。

AWS CLI

停止非同步金鑰片語偵測任務

下列stop-key-phrases-detection-job範例會停止進行中的非同步金鑰片語偵測任務。如果目前的任務狀態為IN_PROGRESS任務標記為終止並進入 STOP_REQUESTED 狀態。如果任務在停止之前完成，則會進入 COMPLETED 狀態。

```
aws comprehend stop-key-phrases-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

輸出：

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
```

```
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopKeyPhrasesDetectionJob。

stop-pii-entities-detection-job

以下程式碼範例顯示如何使用 stop-pii-entities-detection-job。

AWS CLI

停止非同步 pii 實體偵測任務

下列stop-pii-entities-detection-job範例會停止進行中的非同步 pii 實體偵測任務。如果目前的任務狀態為IN_PROGRESS任務標記為終止並進入 STOP_REQUESTED 狀態。如果任務在停止之前完成，則會進入 COMPLETED 狀態。

```
aws comprehend stop-pii-entities-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

輸出：

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopPiiEntitiesDetectionJob。

stop-sentiment-detection-job

以下程式碼範例顯示如何使用 stop-sentiment-detection-job。

AWS CLI

停止非同步情緒偵測任務

下列stop-sentiment-detection-job範例會停止進行中的非同步情緒偵測任務。如果目前的任務狀態為IN_PROGRESS任務標記為終止並進入 STOP_REQUESTED 狀態。如果任務在停止之前完成，則會進入 COMPLETED 狀態。

```
aws comprehend stop-sentiment-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

輸出：

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopSentimentDetectionJob。

stop-targeted-sentiment-detection-job

以下程式碼範例顯示如何使用 stop-targeted-sentiment-detection-job。

AWS CLI

停止非同步目標情緒偵測任務

下列stop-targeted-sentiment-detection-job範例會停止進行中的非同步目標情緒偵測任務。如果目前的任務狀態為IN_PROGRESS任務標記為終止並進入 STOP_REQUESTED 狀態。如果任務在停止之前完成，則會進入 COMPLETED 狀態。

```
aws comprehend stop-targeted-sentiment-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

輸出：

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
```

```
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的 Amazon Comprehend 洞察的非同步分析。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopTargetedSentimentDetectionJob。

stop-training-document-classifier

以下程式碼範例顯示如何使用 stop-training-document-classifier。

AWS CLI

停止訓練文件分類器模型

下列stop-training-document-classifier範例會在進行中時停止訓練文件分類器模型。

```
aws comprehend stop-training-document-classifier
    --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的建立和管理自訂模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopTrainingDocumentClassifier。

stop-training-entity-recognizer

以下程式碼範例顯示如何使用 stop-training-entity-recognizer。

AWS CLI

停止實體辨識器模型的訓練

下列stop-training-entity-recognizer範例會在進行中時停止實體辨識器模型的訓練。

```
aws comprehend stop-training-entity-recognizer
    --entity-recognizer-arn "arn:aws:comprehend:us-west-2:111122223333:entity-recognizer/examplerecognizer1"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的建立和管理自訂模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopTrainingEntityRecognizer。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

範例 1：標記資源

下列tag-resource範例會將單一標籤新增至 Amazon Comprehend 資源。

```
aws comprehend tag-resource \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1 \
    --tags Key=Location,Value=Seattle
```

此命令沒有輸出。

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的標記您的 資源。

範例 2：將多個標籤新增至資源

下列tag-resource範例會將多個標籤新增至 Amazon Comprehend 資源。

```
aws comprehend tag-resource \
    --resource-arn "arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1" \
    --tags Key=location,Value=Seattle Key=Department,Value=Finance
```

此命令沒有輸出。

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的標記您的 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

範例 1：從資源中移除單一標籤

下列untag-resource範例會從 Amazon Comprehend 資源移除單一標籤。

```
aws comprehend untag-resource \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1
    --tag-keys Location
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的標記您的 資源。

範例 2：從資源中移除多個標籤

下列untag-resource範例會從 Amazon Comprehend 資源移除多個標籤。

```
aws comprehend untag-resource \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1
    --tag-keys Location Department
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的標記您的 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-endpoint

以下程式碼範例顯示如何使用 update-endpoint。

AWS CLI

範例 1：更新端點的推論單位

下列update-endpoint範例會更新 端點的相關資訊。在此範例中，推論單位的數量會增加。

```
aws comprehend update-endpoint \
```

```
    --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-endpoint/example-classifier-endpoint
    --desired-inference-units 2
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的管理 Amazon Comprehend 端點。 Amazon Comprehend

範例 2：更新端點的 動作模型

下列update-endpoint範例會更新 端點的相關資訊。在此範例中，作用中模型已變更。

```
aws comprehend update-endpoint \
    --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-endpoint/example-classifier-endpoint
    --active-model-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier-new
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的管理 Amazon Comprehend 端點。 Amazon Comprehend

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateEndpoint。

update-flywheel

以下程式碼範例顯示如何使用 update-flywheel。

AWS CLI

更新飛輪組態

下列update-flywheel範例會更新飛輪組態。在此範例中，飛輪的作用中模型會更新。

```
aws comprehend update-flywheel \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel-1 \
    --active-model-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier/version/new-example-classifier-model
```

輸出：

```
{
    "FlywheelProperties": {
        "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-
entity",
        "ActiveModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier/version/new-example-classifier-model",
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
        "TaskConfig": {
            "LanguageCode": "en",
            "DocumentClassificationConfig": {
                "Mode": "MULTI_CLASS"
            }
        },
        "DataLakeS3Uri": "s3://amzn-s3-demo-bucket/flywheel-entity/
schemaVersion=1/20230616T200543Z/",
        "DataSecurityConfig": {},
        "Status": "ACTIVE",
        "ModelType": "DOCUMENT_CLASSIFIER",
        "CreationTime": "2023-06-16T20:05:43.242000+00:00",
        "LastModifiedTime": "2023-06-19T04:00:43.027000+00:00",
        "LatestFlywheelIteration": "20230619T040032Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend 開發人員指南》中的飛輪概觀。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateFlywheel。

使用 的 Amazon Comprehend Medical 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon Comprehend Medical 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

describe-entities-detection-v2-job

以下程式碼範例顯示如何使用 describe-entities-detection-v2-job。

AWS CLI

描述實體偵測任務

下列describe-entities-detection-v2-job範例顯示與非同步實體偵測任務相關聯的屬性。

```
aws comprehendmedical describe-entities-detection-v2-job \
    --job-id "ab9887877365fe70299089371c043b96"
```

輸出：

```
{
    "ComprehendMedicalAsyncJobProperties": {
        "JobId": "ab9887877365fe70299089371c043b96",
        "JobStatus": "COMPLETED",
        "SubmitTime": "2020-03-18T21:20:15.614000+00:00",
        "EndTime": "2020-03-18T21:27:07.350000+00:00",
        "ExpirationTime": "2020-07-16T21:20:15+00:00",
        "InputDataConfig": {
            "S3Bucket": "comp-med-input",
            "S3Key": ""
        },
        "OutputDataConfig": {
            "S3Bucket": "comp-med-output",
            "S3Key": "867139942017-EntitiesDetection-
ab9887877365fe70299089371c043b96/"
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
        "ModelVersion": "DetectEntitiesModelV20190930"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的批次 APIs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEntitiesDetectionV2Job。

describe-icd10-cm-inference-job

以下程式碼範例顯示如何使用 describe-icd10-cm-inference-job。

AWS CLI

描述 ICD-10-CM 推論任務

下列describe-icd10-cm-inference-job範例說明具有指定 job-id 之請求推論任務的屬性。

```
aws comprehendmedical describe-icd10-cm-inference-job \
    --job-id "5780034166536cdb52ffa3295a1b00a7"
```

輸出：

```
{
    "ComprehendMedicalAsyncJobProperties": {
        "JobId": "5780034166536cdb52ffa3295a1b00a7",
        "JobStatus": "COMPLETED",
        "SubmitTime": "2020-05-18T21:20:15.614000+00:00",
        "EndTime": "2020-05-18T21:27:07.350000+00:00",
        "ExpirationTime": "2020-09-16T21:20:15+00:00",
        "InputDataConfig": {
            "S3Bucket": "comp-med-input",
            "S3Key": "AKIAIOSFODNN7EXAMPLE"
        },
        "OutputDataConfig": {
            "S3Bucket": "comp-med-output",
            "S3Key": "AKIAIOSFODNN7EXAMPLE"
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
        "ModelVersion":  "0.1.0"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的腫瘤科連結批次分析。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeIcd10CmInferenceJob。

describe-phi-detection-job

以下程式碼範例顯示如何使用 describe-phi-detection-job。

AWS CLI

描述 PHI 偵測任務

下列describe-phi-detection-job範例顯示與非同步受保護醫療資訊 (PHI) 偵測任務相關聯的屬性。

```
aws comprehendmedical describe-phi-detection-job \
    --job-id "4750034166536cdb52ffa3295a1b00a3"
```

輸出：

```
{
    "ComprehendMedicalAsyncJobProperties": {
        "JobId": "4750034166536cdb52ffa3295a1b00a3",
        "JobStatus": "COMPLETED",
        "SubmitTime": "2020-03-19T20:38:37.594000+00:00",
        "EndTime": "2020-03-19T20:45:07.894000+00:00",
        "ExpirationTime": "2020-07-17T20:38:37+00:00",
        "InputDataConfig": {
            "S3Bucket": "comp-med-input",
            "S3Key": ""
        },
        "OutputDataConfig": {
            "S3Bucket": "comp-med-output",
            "S3Key": "867139942017-PHIDetection-4750034166536cdb52ffa3295a1b00a3/"
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
        "ModelVersion": "PHIModelV20190903"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的批次 APIs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePhiDetectionJob。

describe-rx-norm-inference-job

以下程式碼範例顯示如何使用 describe-rx-norm-inference-job。

AWS CLI

描述 RxNorm 推論任務

下列describe-rx-norm-inference-job範例說明具有指定 job-id 之請求推論任務的屬性。

```
aws comprehendmedical describe-rx-norm-inference-job \
    --job-id "eg8199877365fc70299089371c043b96"
```

輸出：

```
{
    "ComprehendMedicalAsyncJobProperties": {
        "JobId": "g8199877365fc70299089371c043b96",
        "JobStatus": "COMPLETED",
        "SubmitTime": "2020-05-18T21:20:15.614000+00:00",
        "EndTime": "2020-05-18T21:27:07.350000+00:00",
        "ExpirationTime": "2020-09-16T21:20:15+00:00",
        "InputDataConfig": {
            "S3Bucket": "comp-med-input",
            "S3Key": "AKIAIOSFODNN7EXAMPLE"
        },
        "OutputDataConfig": {
            "S3Bucket": "comp-med-output",
            "S3Key": "AKIAIOSFODNN7EXAMPLE"
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
        "ModelVersion": "0.0.0"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的腫瘤科連結批次分析。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRxNormInferenceJob。

`describe-snomedct-inference-job`

以下程式碼範例顯示如何使用 `describe-snomedct-inference-job`。

AWS CLI

描述 SNOMED CT 推論任務

下列`describe-snomedct-inference-job`範例說明具有指定 job-id 之請求推論任務的屬性。

```
aws comprehendmedical describe-snomedct-inference-job \
    --job-id "2630034166536cdb52ffa3295a1b00a7"
```

輸出：

```
{
    "ComprehendMedicalAsyncJobProperties": {
        "JobId": "2630034166536cdb52ffa3295a1b00a7",
        "JobStatus": "COMPLETED",
        "SubmitTime": "2021-12-18T21:20:15.614000+00:00",
        "EndTime": "2021-12-18T21:27:07.350000+00:00",
        "ExpirationTime": "2022-05-16T21:20:15+00:00",
        "InputDataConfig": {
            "S3Bucket": "comp-med-input",
            "S3Key": "AKIAIOSFODNN7EXAMPLE"
        },
        "OutputDataConfig": {
            "S3Bucket": "comp-med-output",
            "S3Key": "AKIAIOSFODNN7EXAMPLE"
        },
        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
        "ModelVersion":  "0.1.0"
    }
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的腫瘤科連結批次分析。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSnomedctInferenceJob。

detect-entities-v2

以下程式碼範例顯示如何使用 detect-entities-v2。

AWS CLI

範例 1：直接從文字偵測實體

下列detect-entities-v2範例顯示偵測到的實體，並根據類型直接從輸入文字標記它們。

```
aws comprehendmedical detect-entities-v2 \
    --text "Sleeping trouble on present dosage of Clonidine. Severe rash on face and
 leg, slightly itchy."
```

輸出：

```
{
    "Id": 0,
    "BeginOffset": 38,
    "EndOffset": 47,
    "Score": 0.9942955374717712,
    "Text": "Clonidine",
    "Category": "MEDICATION",
    "Type": "GENERIC_NAME",
    "Traits": []
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的偵測實體第 2 版。

範例 2：從檔案路徑偵測實體

下列detect-entities-v2範例顯示偵測到的實體，並根據檔案路徑中的類型加以標記。

```
aws comprehendmedical detect-entities-v2 \
    --text file://medical_entities.txt
```

medical_entities.txt 的內容：

```
{
    "Sleeping trouble on present dosage of Clonidine. Severe rash on face and leg,
 slightly itchy."
}
```

輸出：

```
{
    "Id": 0,
    "BeginOffset": 38,
    "EndOffset": 47,
    "Score": 0.9942955374717712,
    "Text": "Clonidine",
    "Category": "MEDICATION",
    "Type": "GENERIC_NAME",
    "Traits": []
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的偵測實體第 2 版。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetectEntitiesV2。

detect-phi

以下程式碼範例顯示如何使用 detect-phi。

AWS CLI

範例 1：直接從文字偵測受保護的健康資訊 (PHI)

下列detect-phi範例會直接從輸入文字顯示偵測到的受保護醫療資訊 (PHI) 實體。

```
aws comprehendmedical detect-phi \
    --text "Patient Carlos Salazar presented with rash on his upper extremities and dry cough. He lives at 100 Main Street, Anytown, USA where he works from his home as a carpenter."
```

輸出：

```
{
    "Entities": [
        {
            "Id": 0,
            "BeginOffset": 8,
            "EndOffset": 21,
            "Score": 0.9914507269859314,
            "Text": "Carlos Salazar",
            "Category": "PROTECTED_HEALTH_INFORMATION",
```

```
            "Type": "NAME",
            "Traits": []
        },
        {
            "Id": 1,
            "BeginOffset": 94,
            "EndOffset": 109,
            "Score": 0.871849775314331,
            "Text": "100 Main Street, Anytown, USA",
            "Category": "PROTECTED_HEALTH_INFORMATION",
            "Type": "ADDRESS",
            "Traits": []
        },
        {
            "Id": 2,
            "BeginOffset": 145,
            "EndOffset": 154,
            "Score": 0.8302185535430908,
            "Text": "carpenter",
            "Category": "PROTECTED_HEALTH_INFORMATION",
            "Type": "PROFESSION",
            "Traits": []
        }
    ],
    "ModelVersion": "0.0.0"
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的偵測 PHI。

範例 2：直接從檔案路徑偵測受保護的運作狀態資訊 (PHI)

下列`detect-phi`範例顯示從檔案路徑偵測到的受保護醫療資訊 (PHI) 實體。

```
aws comprehendmedical detect-phi \
    --text file://phi.txt
```

`phi.txt` 的內容：

```
"Patient Carlos Salazar presented with a rash on his upper extremities and a dry
 cough. He lives at 100 Main Street, Anytown, USA, where he works from his home as a
 carpenter."
```

輸出：

```
{
    "Entities": [
        {
            "Id": 0,
            "BeginOffset": 8,
            "EndOffset": 21,
            "Score": 0.9914507269859314,
            "Text": "Carlos Salazar",
            "Category": "PROTECTED_HEALTH_INFORMATION",
            "Type": "NAME",
            "Traits": []
        },
        {
            "Id": 1,
            "BeginOffset": 94,
            "EndOffset": 109,
            "Score": 0.871849775314331,
            "Text": "100 Main Street, Anytown, USA",
            "Category": "PROTECTED_HEALTH_INFORMATION",
            "Type": "ADDRESS",
            "Traits": []
        },
        {
            "Id": 2,
            "BeginOffset": 145,
            "EndOffset": 154,
            "Score": 0.8302185535430908,
            "Text": "carpenter",
            "Category": "PROTECTED_HEALTH_INFORMATION",
            "Type": "PROFESSION",
            "Traits": []
        }
    ],
    "ModelVersion": "0.0.0"
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的偵測 PHI。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetectPhi。

`infer-icd10-cm`

以下程式碼範例顯示如何使用 `infer-icd10-cm`。

AWS CLI

範例 1：直接從文字偵測醫療情況實體和 ICD-10-CM Ontology 的連結

下列infer-icd10-cm範例會標記偵測到的醫療狀況實體，並將這些實體與 2019 年國際疾病分類臨床修改 (ICD-10-CM) 版中的代碼連結。

```
aws comprehendmedical infer-icd10-cm \
    --text "The patient complains of abdominal pain, has a long-standing history of
 diabetes treated with Micronase daily."
```

輸出：

```
{
    "Entities": [
        {
            "Id": 0,
            "Text": "abdominal pain",
            "Category": "MEDICAL_CONDITION",
            "Type": "DX_NAME",
            "Score": 0.9475538730621338,
            "BeginOffset": 28,
            "EndOffset": 42,
            "Attributes": [],
            "Traits": [
                {
                    "Name": "SYMPTOM",
                    "Score": 0.6724207401275635
                }
            ],
            "ICD10CMConcepts": [
                {
                    "Description": "Unspecified abdominal pain",
                    "Code": "R10.9",
                    "Score": 0.6904221177101135
                },
                {
                    "Description": "Epigastric pain",
                    "Code": "R10.13",
                    "Score": 0.1364113688468933
                },
                {
                    "Description": "Generalized abdominal pain",
```

```
                    "Code": "R10.84",
                    "Score": 0.12508003413677216
                },
                {
                    "Description": "Left lower quadrant pain",
                    "Code": "R10.32",
                    "Score": 0.10063883662223816
                },
                {
                    "Description": "Lower abdominal pain, unspecified",
                    "Code": "R10.30",
                    "Score": 0.09933677315711975
                }
            ]
        },
        {
            "Id": 1,
            "Text": "diabetes",
            "Category": "MEDICAL_CONDITION",
            "Type": "DX_NAME",
            "Score": 0.9899052977561951,
            "BeginOffset": 75,
            "EndOffset": 83,
            "Attributes": [],
            "Traits": [
                {
                    "Name": "DIAGNOSIS",
                    "Score": 0.9258432388305664
                }
            ],
            "ICD10CMConcepts": [
                {
                    "Description": "Type 2 diabetes mellitus without complications",
                    "Code": "E11.9",
                    "Score": 0.7158446311950684
                },
                {
                    "Description": "Family history of diabetes mellitus",
                    "Code": "Z83.3",
                    "Score": 0.5704703330993652
                },
                {
                    "Description": "Family history of other endocrine, nutritional
and metabolic diseases",
```

```
                    "Code": "Z83.49",
                    "Score": 0.19856023788452148
                },
                {
                    "Description": "Type 1 diabetes mellitus with ketoacidosis
 without coma",
                    "Code": "E10.10",
                    "Score": 0.13285516202449799
                },
                {
                    "Description": "Type 2 diabetes mellitus with hyperglycemia",
                    "Code": "E11.65",
                    "Score": 0.0993388369679451
                }
            ]
        }
    ],
    "ModelVersion": "0.1.0"
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的推論 ICD10-CM。

範例 2：從檔案路徑偵測醫療情況實體並連結至 ICD-10-CM 腫瘤學

下列infer-icd-10-cm範例會標記偵測到的醫療狀況實體，並將這些實體與 2019 年國際疾病分類臨床修改 (ICD-10-CM) 版中的代碼連結。

```
aws comprehendmedical infer-icd10-cm \
    --text file://icd10cm.txt
```

icd10cm.txt 的內容：

```
{
    "The patient complains of abdominal pain, has a long-standing history of
 diabetes treated with Micronase daily."
}
```

輸出：

```
{
    "Entities": [
        {
            "Id": 0,
```

```
            "Text": "abdominal pain",
            "Category": "MEDICAL_CONDITION",
            "Type": "DX_NAME",
            "Score": 0.9475538730621338,
            "BeginOffset": 28,
            "EndOffset": 42,
            "Attributes": [],
            "Traits": [
                {
                    "Name": "SYMPTOM",
                    "Score": 0.6724207401275635
                }
            ],
            "ICD10CMConcepts": [
                {
                    "Description": "Unspecified abdominal pain",
                    "Code": "R10.9",
                    "Score": 0.6904221177101135
                },
                {
                    "Description": "Epigastric pain",
                    "Code": "R10.13",
                    "Score": 0.1364113688468933
                },
                {
                    "Description": "Generalized abdominal pain",
                    "Code": "R10.84",
                    "Score": 0.12508003413677216
                },
                {
                    "Description": "Left lower quadrant pain",
                    "Code": "R10.32",
                    "Score": 0.10063883662223816
                },
                {
                    "Description": "Lower abdominal pain, unspecified",
                    "Code": "R10.30",
                    "Score": 0.09933677315711975
                }
            ]
        },
        {
            "Id": 1,
            "Text": "diabetes",
```

```
            "Category": "MEDICAL_CONDITION",
            "Type": "DX_NAME",
            "Score": 0.9899052977561951,
            "BeginOffset": 75,
            "EndOffset": 83,
            "Attributes": [],
            "Traits": [
                {
                    "Name": "DIAGNOSIS",
                    "Score": 0.9258432388305664
                }
            ],
            "ICD10CMConcepts": [
                {
                    "Description": "Type 2 diabetes mellitus without complications",
                    "Code": "E11.9",
                    "Score": 0.7158446311950684
                },
                {
                    "Description": "Family history of diabetes mellitus",
                    "Code": "Z83.3",
                    "Score": 0.5704703330993652
                },
                {
                    "Description": "Family history of other endocrine, nutritional
and metabolic diseases",
                    "Code": "Z83.49",
                    "Score": 0.19856023788452148
                },
                {
                    "Description": "Type 1 diabetes mellitus with ketoacidosis
without coma",
                    "Code": "E10.10",
                    "Score": 0.13285516202449799
                },
                {
                    "Description": "Type 2 diabetes mellitus with hyperglycemia",
                    "Code": "E11.65",
                    "Score": 0.0993388369679451
                }
            ]
        }
    ],
    "ModelVersion": "0.1.0"
```

```
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的 Infer-ICD10-CM。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 InferIcd10Cm。

`infer-rx-norm`

以下程式碼範例顯示如何使用 `infer-rx-norm`。

AWS CLI

範例 1：直接從文字偵測藥物實體和連結至 RxNorm

下列`infer-rx-norm`範例顯示並標記偵測到的藥品實體，並將這些實體連結至 National Library of Medicine RxNorm RxCUI)。

```
aws comprehendmedical infer-rx-norm \
    --text "Patient reports taking Levothyroxine 125 micrograms p.o. once daily, but denies taking Synthroid."
```

輸出：

```
{
    "Entities": [
        {
            "Id": 0,
            "Text": "Levothyroxine",
            "Category": "MEDICATION",
            "Type": "GENERIC_NAME",
            "Score": 0.9996285438537598,
            "BeginOffset": 23,
            "EndOffset": 36,
            "Attributes": [
                {
                    "Type": "DOSAGE",
                    "Score": 0.9892290830612183,
                    "RelationshipScore": 0.9997978806495667,
                    "Id": 1,
                    "BeginOffset": 37,
                    "EndOffset": 51,
                    "Text": "125 micrograms",
                    "Traits": []
```

```
                },
                {
                    "Type": "ROUTE_OR_MODE",
                    "Score": 0.9988924860954285,
                    "RelationshipScore": 0.998291552066803,
                    "Id": 2,
                    "BeginOffset": 52,
                    "EndOffset": 56,
                    "Text": "p.o.",
                    "Traits": []
                },
                {
                    "Type": "FREQUENCY",
                    "Score": 0.9953463673591614,
                    "RelationshipScore": 0.9999889135360718,
                    "Id": 3,
                    "BeginOffset": 57,
                    "EndOffset": 67,
                    "Text": "once daily",
                    "Traits": []
                }
            ],
            "Traits": [],
            "RxNormConcepts": [
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet",
                    "Code": "966224",
                    "Score": 0.9912070631980896
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Capsule",
                    "Code": "966405",
                    "Score": 0.8698278665542603
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Synthroid]",
                    "Code": "966191",
                    "Score": 0.7448257803916931
                },
                {
                    "Description": "levothyroxine",
                    "Code": "10582",
                    "Score": 0.7050482630729675
```

```
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Levoxyl]",
                    "Code": "966190",
                    "Score": 0.6921631693840027
                }
            ]
        },
        {
            "Id": 4,
            "Text": "Synthroid",
            "Category": "MEDICATION",
            "Type": "BRAND_NAME",
            "Score": 0.9946461319923401,
            "BeginOffset": 86,
            "EndOffset": 95,
            "Attributes": [],
            "Traits": [
                {
                    "Name": "NEGATION",
                    "Score": 0.5167351961135864
                }
            ],
            "RxNormConcepts": [
                {
                    "Description": "Synthroid",
                    "Code": "224920",
                    "Score": 0.9462039470672607
                },
                {
                    "Description": "Levothyroxine Sodium 0.088 MG Oral Tablet
[Synthroid]",
                    "Code": "966282",
                    "Score": 0.8309829235076904
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Synthroid]",
                    "Code": "966191",
                    "Score": 0.4945160448551178
                },
                {
```

```
                    "Description": "Levothyroxine Sodium 0.05 MG Oral Tablet
 [Synthroid]",
                    "Code": "966247",
                    "Score": 0.3674522042274475
                },
                {
                    "Description": "Levothyroxine Sodium 0.025 MG Oral Tablet
 [Synthroid]",
                    "Code": "966158",
                    "Score": 0.2588822841644287
                }
            ]
        }
    ],
    "ModelVersion": "0.0.0"
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的推論 RxNorm。

範例 2：從檔案路徑偵測藥物實體和 RxNorm 的連結。

下列infer-rx-norm範例顯示並標記偵測到的藥品實體，並將這些實體連結至 National Library of Medicine RxNorm RxCUI)。

```
aws comprehendmedical infer-rx-norm \
    --text file://rxnorm.txt
```

rxnorm.txt 的內容：

```
{
    "Patient reports taking Levothyroxine 125 micrograms p.o. once daily, but denies
 taking Synthroid."
}
```

輸出：

```
{
    "Entities": [
        {
            "Id": 0,
            "Text": "Levothyroxine",
            "Category": "MEDICATION",
            "Type": "GENERIC_NAME",
```

```
            "Score": 0.9996285438537598,
            "BeginOffset": 23,
            "EndOffset": 36,
            "Attributes": [
                {
                    "Type": "DOSAGE",
                    "Score": 0.9892290830612183,
                    "RelationshipScore": 0.9997978806495667,
                    "Id": 1,
                    "BeginOffset": 37,
                    "EndOffset": 51,
                    "Text": "125 micrograms",
                    "Traits": []
                },
                {
                    "Type": "ROUTE_OR_MODE",
                    "Score": 0.9988924860954285,
                    "RelationshipScore": 0.998291552066803,
                    "Id": 2,
                    "BeginOffset": 52,
                    "EndOffset": 56,
                    "Text": "p.o.",
                    "Traits": []
                },
                {
                    "Type": "FREQUENCY",
                    "Score": 0.9953463673591614,
                    "RelationshipScore": 0.9999889135360718,
                    "Id": 3,
                    "BeginOffset": 57,
                    "EndOffset": 67,
                    "Text": "once daily",
                    "Traits": []
                }
            ],
            "Traits": [],
            "RxNormConcepts": [
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet",
                    "Code": "966224",
                    "Score": 0.9912070631980896
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Capsule",
```

```
                    "Code": "966405",
                    "Score": 0.8698278665542603
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Synthroid]",
                    "Code": "966191",
                    "Score": 0.7448257803916931
                },
                {
                    "Description": "levothyroxine",
                    "Code": "10582",
                    "Score": 0.7050482630729675
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Levoxyl]",
                    "Code": "966190",
                    "Score": 0.6921631693840027
                }
            ]
        },
        {
            "Id": 4,
            "Text": "Synthroid",
            "Category": "MEDICATION",
            "Type": "BRAND_NAME",
            "Score": 0.9946461319923401,
            "BeginOffset": 86,
            "EndOffset": 95,
            "Attributes": [],
            "Traits": [
                {
                    "Name": "NEGATION",
                    "Score": 0.5167351961135864
                }
            ],
            "RxNormConcepts": [
                {
                    "Description": "Synthroid",
                    "Code": "224920",
                    "Score": 0.9462039470672607
                },
                {
```

```
                    "Description": "Levothyroxine Sodium 0.088 MG Oral Tablet
 [Synthroid]",
                    "Code": "966282",
                    "Score": 0.8309829235076904
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
 [Synthroid]",
                    "Code": "966191",
                    "Score": 0.4945160448551178
                },
                {
                    "Description": "Levothyroxine Sodium 0.05 MG Oral Tablet
 [Synthroid]",
                    "Code": "966247",
                    "Score": 0.3674522042274475
                },
                {
                    "Description": "Levothyroxine Sodium 0.025 MG Oral Tablet
 [Synthroid]",
                    "Code": "966158",
                    "Score": 0.2588822841644287
                }
            ]
        }
    ],
    "ModelVersion": "0.0.0"
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的推論 RxNorm。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 InferRxNorm。

`infer-snomedct`

以下程式碼範例顯示如何使用 `infer-snomedct`。

AWS CLI

範例：直接從文字偵測實體並連結至 SNOMED CT Ontology

下列`infer-snomedct`範例說明如何偵測醫療實體，並將其連結至 2021 年 3 年 Systematized Nomenclature of Medicine， clinical Terms (SNOMED CT) 版本的概念。

```
aws comprehendmedical infer-snomedct \
    --text "The patient complains of abdominal pain, has a long-standing history of
 diabetes treated with Micronase daily."
```

輸出：

```
{
    "Entities": [
        {
            "Id": 3,
            "BeginOffset": 26,
            "EndOffset": 40,
            "Score": 0.9598260521888733,
            "Text": "abdominal pain",
            "Category": "MEDICAL_CONDITION",
            "Type": "DX_NAME",
            "Traits": [
                {
                    "Name": "SYMPTOM",
                    "Score": 0.6819021701812744
                }
            ]
        },
        {
            "Id": 4,
            "BeginOffset": 73,
            "EndOffset": 81,
            "Score": 0.9905840158462524,
            "Text": "diabetes",
            "Category": "MEDICAL_CONDITION",
            "Type": "DX_NAME",
            "Traits": [
                {
                    "Name": "DIAGNOSIS",
                    "Score": 0.9255214333534241
                }
            ]
        },
        {
            "Id": 1,
            "BeginOffset": 95,
            "EndOffset": 104,
            "Score": 0.6371926665306091,
```

```
            "Text": "Micronase",
            "Category": "MEDICATION",
            "Type": "BRAND_NAME",
            "Traits": [],
            "Attributes": [
                {
                    "Type": "FREQUENCY",
                    "Score": 0.9761165380477905,
                    "RelationshipScore": 0.9984188079833984,
                    "RelationshipType": "FREQUENCY",
                    "Id": 2,
                    "BeginOffset": 105,
                    "EndOffset": 110,
                    "Text": "daily",
                    "Category": "MEDICATION",
                    "Traits": []
                }
            ]
        }
    ],
    "UnmappedAttributes": [],
    "ModelVersion": "1.0.0"
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的 InferSNOMEDCT。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 InferSnomedct。

list-entities-detection-v2-jobs

以下程式碼範例顯示如何使用 list-entities-detection-v2-jobs。

AWS CLI

列出實體偵測任務

下列list-entities-detection-v2-jobs範例列出目前的非同步偵測任務。

```
aws comprehendmedical list-entities-detection-v2-jobs
```

輸出：

```
{
```

```
    "ComprehendMedicalAsyncJobPropertiesList": [
        {
            "JobId": "ab9887877365fe70299089371c043b96",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2020-03-19T20:38:37.594000+00:00",
            "EndTime": "2020-03-19T20:45:07.894000+00:00",
            "ExpirationTime": "2020-07-17T20:38:37+00:00",
            "InputDataConfig": {
                "S3Bucket": "comp-med-input",
                "S3Key": ""
            },
            "OutputDataConfig": {
                "S3Bucket": "comp-med-output",
                "S3Key": "867139942017-EntitiesDetection-
ab9887877365fe70299089371c043b96/"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
            "ModelVersion": "DetectEntitiesModelV20190930"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的批次 APIs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEntitiesDetectionV2Jobs。

`list-icd10-cm-inference-jobs`

以下程式碼範例顯示如何使用 `list-icd10-cm-inference-jobs`。

AWS CLI

列出所有目前的 ICD-10-CM 推論任務

下列範例顯示 `list-icd10-cm-inference-jobs`操作如何傳回目前非同步 ICD-10-CM 批次推論任務的清單。

```
aws comprehendmedical list-icd10-cm-inference-jobs
```

輸出：

```
{
    "ComprehendMedicalAsyncJobPropertiesList": [
        {
            "JobId": "5780034166536cdb52ffa3295a1b00a7",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2020-05-19T20:38:37.594000+00:00",
            "EndTime": "2020-05-19T20:45:07.894000+00:00",
            "ExpirationTime": "2020-09-17T20:38:37+00:00",
            "InputDataConfig": {
                "S3Bucket": "comp-med-input",
                "S3Key": "AKIAIOSFODNN7EXAMPLE"
            },
            "OutputDataConfig": {
                "S3Bucket": "comp-med-output",
                "S3Key": "AKIAIOSFODNN7EXAMPLE"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
            "ModelVersion":  "0.1.0"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的腫瘤科連結批次分析。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListIcd10CmInferenceJobs。

list-phi-detection-jobs

以下程式碼範例顯示如何使用 `list-phi-detection-jobs`。

AWS CLI

列出受保護醫療資訊 (PHI) 偵測任務

下列`list-phi-detection-jobs`範例列出目前的受保護醫療資訊 (PHI) 偵測任務

```
aws comprehendmedical list-phi-detection-jobs
```

輸出：

```
{
    "ComprehendMedicalAsyncJobPropertiesList": [
        {
            "JobId": "4750034166536cdb52ffa3295a1b00a3",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2020-03-19T20:38:37.594000+00:00",
            "EndTime": "2020-03-19T20:45:07.894000+00:00",
            "ExpirationTime": "2020-07-17T20:38:37+00:00",
            "InputDataConfig": {
                "S3Bucket": "comp-med-input",
                "S3Key": ""
            },
            "OutputDataConfig": {
                "S3Bucket": "comp-med-output",
                "S3Key": "867139942017-
PHIDetection-4750034166536cdb52ffa3295a1b00a3/"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
            "ModelVersion": "PHIModelV20190903"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的批次 APIs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPhiDetectionJobs。

list-rx-norm-inference-jobs

以下程式碼範例顯示如何使用 `list-rx-norm-inference-jobs`。

AWS CLI

列出所有目前的 Rx-Norm 推論任務

下列範例顯示 如何`list-rx-norm-inference-jobs`傳回目前非同步 Rx-Norm 批次推論任務的清單。

```
aws comprehendmedical list-rx-norm-inference-jobs
```

輸出：

```
{
    "ComprehendMedicalAsyncJobPropertiesList": [
        {
            "JobId": "4980034166536cfb52gga3295a1b00a3",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2020-05-19T20:38:37.594000+00:00",
            "EndTime": "2020-05-19T20:45:07.894000+00:00",
            "ExpirationTime": "2020-09-17T20:38:37+00:00",
            "InputDataConfig": {
                "S3Bucket": "comp-med-input",
                "S3Key": "AKIAIOSFODNN7EXAMPLE"
            },
            "OutputDataConfig": {
                "S3Bucket": "comp-med-output",
                "S3Key": "AKIAIOSFODNN7EXAMPLE"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
            "ModelVersion": "0.0.0"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的腫瘤科連結批次分析。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRxNormInferenceJobs。

`list-snomedct-inference-jobs`

以下程式碼範例顯示如何使用 `list-snomedct-inference-jobs`。

AWS CLI

列出所有 SNOMED CT 推論任務

下列範例顯示 `list-snomedct-inference-jobs`操作如何傳回目前非同步 SNOMED CT 批次推論任務的清單。

```
aws comprehendmedical list-snomedct-inference-jobs
```

輸出：

```
{
    "ComprehendMedicalAsyncJobPropertiesList": [
        {
            "JobId": "5780034166536cdb52ffa3295a1b00a7",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2020-05-19T20:38:37.594000+00:00",
            "EndTime": "2020-05-19T20:45:07.894000+00:00",
            "ExpirationTime": "2020-09-17T20:38:37+00:00",
            "InputDataConfig": {
                "S3Bucket": "comp-med-input",
                "S3Key": "AKIAIOSFODNN7EXAMPLE"
            },
            "OutputDataConfig": {
                "S3Bucket": "comp-med-output",
                "S3Key": "AKIAIOSFODNN7EXAMPLE"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
            "ModelVersion":  "0.1.0"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的腫瘤科連結批次分析。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSnomedctInferenceJobs。

start-entities-detection-v2-job

以下程式碼範例顯示如何使用 start-entities-detection-v2-job。

AWS CLI

啟動實體偵測任務

下列start-entities-detection-v2-job範例會啟動非同步實體偵測任務。

```
aws comprehendmedical start-entities-detection-v2-job \
    --input-data-config "S3Bucket=comp-med-input" \
    --output-data-config "S3Bucket=comp-med-output" \
    --data-access-role-arn arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole \
    --language-code en
```

輸出：

```
{
    "JobId": "ab9887877365fe70299089371c043b96"
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的批次 APIs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartEntitiesDetectionV2Job。

start-icd10-cm-inference-job

以下程式碼範例顯示如何使用 start-icd10-cm-inference-job。

AWS CLI

啟動 ICD-10-CM 推論任務

下列start-icd10-cm-inference-job範例會啟動 ICD-10-CM 推論批次分析任務。

```
aws comprehendmedical start-icd10-cm-inference-job \
    --input-data-config "S3Bucket=comp-med-input" \
    --output-data-config "S3Bucket=comp-med-output" \
    --data-access-role-arn arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole \
    --language-code en
```

輸出：

```
{
    "JobId": "ef7289877365fc70299089371c043b96"
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的腫瘤科連結批次分析。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartIcd10CmInferenceJob。

start-phi-detection-job

以下程式碼範例顯示如何使用 start-phi-detection-job。

AWS CLI

啟動 PHI 偵測任務

下列start-phi-detection-job範例會啟動非同步 PHI 實體偵測任務。

```
aws comprehendmedical start-phi-detection-job \
    --input-data-config "S3Bucket=comp-med-input" \
    --output-data-config "S3Bucket=comp-med-output" \
    --data-access-role-arn arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole \
    --language-code en
```

輸出：

```
{
    "JobId": "ab9887877365fe70299089371c043b96"
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的批次 APIs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartPhiDetectionJob。

start-rx-norm-inference-job

以下程式碼範例顯示如何使用 start-rx-norm-inference-job。

AWS CLI

啟動 RxNorm 推論任務

下列start-rx-norm-inference-job範例會啟動 RxNorm 推論批次分析任務。

```
aws comprehendmedical start-rx-norm-inference-job \
    --input-data-config "S3Bucket=comp-med-input" \
```

```
    --output-data-config "S3Bucket=comp-med-output" \
    --data-access-role-arn arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole \
    --language-code en
```

輸出：

```
{
    "JobId": "eg8199877365fc70299089371c043b96"
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的腫瘤科連結批次分析。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartRxNormInferenceJob。

`start-snomedct-inference-job`

以下程式碼範例顯示如何使用 `start-snomedct-inference-job`。

AWS CLI

啟動 SNOMED CT 推論任務

下列`start-snomedct-inference-job`範例會啟動 SNOMED CT 推論批次分析任務。

```
aws comprehendmedical start-snomedct-inference-job \
    --input-data-config "S3Bucket=comp-med-input" \
    --output-data-config "S3Bucket=comp-med-output" \
    --data-access-role-arn arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole \
    --language-code en
```

輸出：

```
{
    "JobId": "dg7289877365fc70299089371c043b96"
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的腫瘤科連結批次分析。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartSnomedctInferenceJob。

stop-entities-detection-v2-job

以下程式碼範例顯示如何使用 stop-entities-detection-v2-job。

AWS CLI

停止實體偵測任務

下列stop-entities-detection-v2-job範例會停止非同步實體偵測任務。

```
aws comprehendmedical stop-entities-detection-v2-job \
    --job-id "ab9887877365fe70299089371c043b96"
```

輸出：

```
{
    "JobId": "ab9887877365fe70299089371c043b96"
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的批次 APIs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopEntitiesDetectionV2Job。

stop-icd10-cm-inference-job

以下程式碼範例顯示如何使用 stop-icd10-cm-inference-job。

AWS CLI

停止 ICD-10-CM 推論任務

下列stop-icd10-cm-inference-job範例會停止 ICD-10-CM 推論批次分析任務。

```
aws comprehendmedical stop-icd10-cm-inference-job \
    --job-id "4750034166536cdb52ffa3295a1b00a3"
```

輸出：

```
{
```

```
    "JobId": "ef7289877365fc70299089371c043b96",
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的腫瘤科連結批次分析。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopIcd10CmInferenceJob。

stop-phi-detection-job

以下程式碼範例顯示如何使用 stop-phi-detection-job。

AWS CLI

停止受保護醫療資訊 (PHI) 偵測任務

下列stop-phi-detection-job範例會停止非同步受保護醫療資訊 (PHI) 偵測任務。

```
aws comprehendmedical stop-phi-detection-job \
    --job-id "4750034166536cdb52ffa3295a1b00a3"
```

輸出：

```
{
    "JobId": "ab9887877365fe70299089371c043b96"
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的批次 APIs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopPhiDetectionJob。

stop-rx-norm-inference-job

以下程式碼範例顯示如何使用 stop-rx-norm-inference-job。

AWS CLI

停止 RxNorm 推論任務

下列stop-rx-norm-inference-job範例會停止 ICD-10-CM 推論批次分析任務。

```
aws comprehendmedical stop-rx-norm-inference-job \
```

```
    --job-id "eg8199877365fc70299089371c043b96"
```

輸出：

```
{
    "JobId": "eg8199877365fc70299089371c043b96",
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的腫瘤科連結批次分析。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopRxNormInferenceJob。

`stop-snomedct-inference-job`

以下程式碼範例顯示如何使用 `stop-snomedct-inference-job`。

AWS CLI

停止 SNOMED CT 推論任務

下列`stop-snomedct-inference-job`範例會停止 SNOMED CT 推論批次分析任務。

```
aws comprehendmedical stop-snomedct-inference-job \
    --job-id "8750034166436cdb52ffa3295a1b00a1"
```

輸出：

```
{
    "JobId": "8750034166436cdb52ffa3295a1b00a1",
}
```

如需詳細資訊，請參閱《Amazon Comprehend Medical 開發人員指南》中的腫瘤科連結批次分析。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopSnomedctInferenceJob。

AWS Config 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS Config。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`delete-config-rule`

以下程式碼範例顯示如何使用 `delete-config-rule`。

AWS CLI

刪除 Config AWS 規則

下列命令會刪除名為 的 AWS Config 規則`MyConfigRule`：

```
aws configservice delete-config-rule --config-rule-name MyConfigRule
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteConfigRule。

`delete-delivery-channel`

以下程式碼範例顯示如何使用 `delete-delivery-channel`。

AWS CLI

刪除交付管道

下列命令會刪除預設交付管道：

```
aws configservice delete-delivery-channel --delivery-channel-name default
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDeliveryChannel。

`delete-evaluation-results`

以下程式碼範例顯示如何使用 `delete-evaluation-results`。

AWS CLI

手動刪除評估結果

下列命令會刪除 AWS 受管規則 s3-bucket-versioning-enabled 的目前評估結果：

```
aws configservice delete-evaluation-results --config-rule-name s3-bucket-versioning-enabled
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteEvaluationResults。

deliver-config-snapshot

以下程式碼範例顯示如何使用 deliver-config-snapshot。

AWS CLI

交付組態快照

下列命令會將組態快照傳送至屬於預設交付通道的 Amazon S3 儲存貯體：

```
aws configservice deliver-config-snapshot --delivery-channel-name default
```

輸出：

```
{
    "configSnapshotId": "d0333b00-a683-44af-921e-examplefb794"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeliverConfigSnapshot。

describe-compliance-by-config-rule

以下程式碼範例顯示如何使用 describe-compliance-by-config-rule。

AWS CLI

取得 Config AWS 規則的合規資訊

下列命令會傳回一或多個 AWS 資源違反的每個 AWS Config 規則的合規資訊：

```
aws configservice describe-compliance-by-config-rule --compliance-
types NON_COMPLIANT
```

在輸出中，每個CappedCount屬性的值會指出有多少資源不符合相關規則。例如，下列輸出表示 3 個資源不符合名為 的規則InstanceTypesAreT2micro。

輸出：

```
{
    "ComplianceByConfigRules": [
        {
            "Compliance": {
                "ComplianceContributorCount": {
                    "CappedCount": 3,
                    "CapExceeded": false
                },
                "ComplianceType": "NON_COMPLIANT"
            },
            "ConfigRuleName": "InstanceTypesAreT2micro"
        },
        {
            "Compliance": {
                "ComplianceContributorCount": {
                    "CappedCount": 10,
                    "CapExceeded": false
                },
                "ComplianceType": "NON_COMPLIANT"
            },
            "ConfigRuleName": "RequiredTagsForVolumes"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeComplianceByConfigRule。

describe-compliance-by-resource

以下程式碼範例顯示如何使用 describe-compliance-by-resource。

AWS CLI

取得 AWS 資源的合規資訊

下列命令會傳回 Config 所記錄且違反一或多個規則的每個 EC2 AWS 執行個體的合規資訊：

```
aws configservice describe-compliance-by-resource --resource-type AWS::EC2::Instance
 --compliance-types NON_COMPLIANT
```

在輸出中，每個CappedCount屬性的值會指出資源違反多少規則。例如，下列輸出表示執行個體i-1a2b3c4d違反 2 個規則。

輸出：

```
{
    "ComplianceByResources": [
        {
            "ResourceType": "AWS::EC2::Instance",
            "ResourceId": "i-1a2b3c4d",
            "Compliance": {
                "ComplianceContributorCount": {
                    "CappedCount": 2,
                    "CapExceeded": false
                },
                "ComplianceType": "NON_COMPLIANT"
            }
        },
        {
            "ResourceType": "AWS::EC2::Instance",
            "ResourceId": "i-2a2b3c4d ",
            "Compliance": {
                "ComplianceContributorCount": {
                    "CappedCount": 3,
                    "CapExceeded": false
                },
                "ComplianceType": "NON_COMPLIANT"
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeComplianceByResource。

describe-config-rule-evaluation-status

以下程式碼範例顯示如何使用 describe-config-rule-evaluation-status。

AWS CLI

取得 Config AWS 規則的狀態資訊

下列命令會傳回名為 之 AWS Config 規則的狀態資訊MyConfigRule：

```
aws configservice describe-config-rule-evaluation-status --config-rule-
names MyConfigRule
```

輸出：

```
{
    "ConfigRulesEvaluationStatus": [
        {
            "ConfigRuleArn": "arn:aws:config:us-east-1:123456789012:config-rule/
config-rule-abcdef",
            "FirstActivatedTime": 1450311703.844,
            "ConfigRuleId": "config-rule-abcdef",
            "LastSuccessfulInvocationTime": 1450314643.156,
            "ConfigRuleName": "MyConfigRule"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeConfigRuleEvaluationStatus。

describe-config-rules

以下程式碼範例顯示如何使用 describe-config-rules。

AWS CLI

取得 Config AWS 規則的詳細資訊

下列命令會傳回名為 之 AWS Config 規則的詳細資訊InstanceTypesAreT2micro：

```
aws configservice describe-config-rules --config-rule-names InstanceTypesAreT2micro
```

輸出：

```
{
```

```
    "ConfigRules": [
        {
            "ConfigRuleState": "ACTIVE",
            "Description": "Evaluates whether EC2 instances are the t2.micro type.",
            "ConfigRuleName": "InstanceTypesAreT2micro",
            "ConfigRuleArn": "arn:aws:config:us-east-1:123456789012:config-rule/
config-rule-abcdef",
            "Source": {
                "Owner": "CUSTOM_LAMBDA",
                "SourceIdentifier": "arn:aws:lambda:us-
east-1:123456789012:function:InstanceTypeCheck",
                "SourceDetails": [
                    {
                        "EventSource": "aws.config",
                        "MessageType": "ConfigurationItemChangeNotification"
                    }
                ]
            },
            "InputParameters": "{\"desiredInstanceType\":\"t2.micro\"}",
            "Scope": {
                "ComplianceResourceTypes": [
                    "AWS::EC2::Instance"
                ]
            },
            "ConfigRuleId": "config-rule-abcdef"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeConfigRules。

describe-configuration-recorder-status

以下程式碼範例顯示如何使用 describe-configuration-recorder-status。

AWS CLI

取得組態記錄器的狀態資訊

下列命令會傳回預設組態記錄器的狀態：

```
aws configservice describe-configuration-recorder-status
```

輸出：

```
{
    "ConfigurationRecordersStatus": [
        {
            "name": "default",
            "lastStatus": "SUCCESS",
            "recording": true,
            "lastStatusChangeTime": 1452193834.344,
            "lastStartTime": 1441039997.819,
            "lastStopTime": 1441039992.835
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeConfigurationRecorderStatus。

describe-configuration-recorders

以下程式碼範例顯示如何使用 describe-configuration-recorders。

AWS CLI

取得組態記錄器的詳細資訊

下列命令會傳回預設組態記錄器的詳細資訊：

```
aws configservice describe-configuration-recorders
```

輸出：

```
{
    "ConfigurationRecorders": [
        {
            "recordingGroup": {
                "allSupported": true,
                "resourceTypes": [],
                "includeGlobalResourceTypes": true
            },
            "roleARN": "arn:aws:iam::123456789012:role/config-ConfigRole-
A1B2C3D4E5F6",
```

```
            "name": "default"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeConfigurationRecorders。

`describe-delivery-channel-status`

以下程式碼範例顯示如何使用 `describe-delivery-channel-status`。

AWS CLI

取得交付管道的狀態資訊

下列命令會傳回交付通道的狀態：

```
aws configservice describe-delivery-channel-status
```

輸出：

```
{
    "DeliveryChannelsStatus": [
        {
            "configStreamDeliveryInfo": {
                "lastStatusChangeTime": 1452193834.381,
                "lastStatus": "SUCCESS"
            },
            "configHistoryDeliveryInfo": {
                "lastSuccessfulTime": 1450317838.412,
                "lastStatus": "SUCCESS",
                "lastAttemptTime": 1450317838.412
            },
            "configSnapshotDeliveryInfo": {
                "lastSuccessfulTime": 1452185597.094,
                "lastStatus": "SUCCESS",
                "lastAttemptTime": 1452185597.094
            },
            "name": "default"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDeliveryChannelStatus。

describe-delivery-channels

以下程式碼範例顯示如何使用 describe-delivery-channels。

AWS CLI

取得交付管道的詳細資訊

下列命令會傳回有關交付管道的詳細資訊：

```
aws configservice describe-delivery-channels
```

輸出：

```
{
    "DeliveryChannels": [
        {
            "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",
            "name": "default",
            "s3BucketName": "config-bucket-123456789012"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDeliveryChannels。

get-compliance-details-by-config-rule

以下程式碼範例顯示如何使用 get-compliance-details-by-config-rule。

AWS CLI

取得 Config AWS 規則的評估結果

下列命令會傳回不符合名為 之 AWS Config 規則的所有資源的評估結果InstanceTypesAreT2micro：

```
aws configservice get-compliance-details-by-config-rule --config-rule-
name InstanceTypesAreT2micro --compliance-types NON_COMPLIANT
```

輸出：

```
{
    "EvaluationResults": [
        {
            "EvaluationResultIdentifier": {
                "OrderingTimestamp": 1450314635.065,
                "EvaluationResultQualifier": {
                    "ResourceType": "AWS::EC2::Instance",
                    "ResourceId": "i-1a2b3c4d",
                    "ConfigRuleName": "InstanceTypesAreT2micro"
                }
            },
            "ResultRecordedTime": 1450314645.261,
            "ConfigRuleInvokedTime": 1450314642.948,
            "ComplianceType": "NON_COMPLIANT"
        },
        {
            "EvaluationResultIdentifier": {
                "OrderingTimestamp": 1450314635.065,
                "EvaluationResultQualifier": {
                    "ResourceType": "AWS::EC2::Instance",
                    "ResourceId": "i-2a2b3c4d",
                    "ConfigRuleName": "InstanceTypesAreT2micro"
                }
            },
            "ResultRecordedTime": 1450314645.18,
            "ConfigRuleInvokedTime": 1450314642.902,
            "ComplianceType": "NON_COMPLIANT"
        },
        {
            "EvaluationResultIdentifier": {
                "OrderingTimestamp": 1450314635.065,
                "EvaluationResultQualifier": {
                    "ResourceType": "AWS::EC2::Instance",
                    "ResourceId": "i-3a2b3c4d",
                    "ConfigRuleName": "InstanceTypesAreT2micro"
                }
            },
            "ResultRecordedTime": 1450314643.346,
            "ConfigRuleInvokedTime": 1450314643.124,
            "ComplianceType": "NON_COMPLIANT"
        }
    ]
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetComplianceDetailsByConfigRule。

get-compliance-details-by-resource

以下程式碼範例顯示如何使用 get-compliance-details-by-resource。

AWS CLI

取得 AWS 資源的評估結果

下列命令會傳回 EC2 執行個體i-1a2b3c4d未遵守之每個規則的評估結果：

```
aws configservice get-compliance-details-by-resource --resource-
type AWS::EC2::Instance --resource-id i-1a2b3c4d --compliance-types NON_COMPLIANT
```

輸出：

```
{
    "EvaluationResults": [
        {
            "EvaluationResultIdentifier": {
                "OrderingTimestamp": 1450314635.065,
                "EvaluationResultQualifier": {
                    "ResourceType": "AWS::EC2::Instance",
                    "ResourceId": "i-1a2b3c4d",
                    "ConfigRuleName": "InstanceTypesAreT2micro"
                }
            },
            "ResultRecordedTime": 1450314643.288,
            "ConfigRuleInvokedTime": 1450314643.034,
            "ComplianceType": "NON_COMPLIANT"
        },
        {
            "EvaluationResultIdentifier": {
                "OrderingTimestamp": 1450314635.065,
                "EvaluationResultQualifier": {
                    "ResourceType": "AWS::EC2::Instance",
                    "ResourceId": "i-1a2b3c4d",
                    "ConfigRuleName": "RequiredTagForEC2Instances"
                }
```

```
            },
            "ResultRecordedTime": 1450314645.261,
            "ConfigRuleInvokedTime": 1450314642.948,
            "ComplianceType": "NON_COMPLIANT"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetComplianceDetailsByResource。

get-compliance-summary-by-config-rule

以下程式碼範例顯示如何使用 get-compliance-summary-by-config-rule。

AWS CLI

取得 Config AWS 規則的合規摘要

下列命令會傳回合規的規則數目和不合規的數目：

```
aws configservice get-compliance-summary-by-config-rule
```

在輸出中，每個CappedCount屬性的值會指出合規或不合規的規則數量。

輸出：

```
{
    "ComplianceSummary": {
        "NonCompliantResourceCount": {
            "CappedCount": 3,
            "CapExceeded": false
        },
        "ComplianceSummaryTimestamp": 1452204131.493,
        "CompliantResourceCount": {
            "CappedCount": 2,
            "CapExceeded": false
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetComplianceSummaryByConfigRule。

get-compliance-summary-by-resource-type

以下程式碼範例顯示如何使用 get-compliance-summary-by-resource-type。

AWS CLI

取得所有資源類型的合規摘要

下列命令會傳回不合規 AWS 的資源數目，以及合規的數目：

```
aws configservice get-compliance-summary-by-resource-type
```

在輸出中，每個CappedCount屬性的值會指出有多少資源合規或不合規。

輸出：

```
{
    "ComplianceSummariesByResourceType": [
        {
            "ComplianceSummary": {
                "NonCompliantResourceCount": {
                    "CappedCount": 16,
                    "CapExceeded": false
                },
                "ComplianceSummaryTimestamp": 1453237464.543,
                "CompliantResourceCount": {
                    "CappedCount": 10,
                    "CapExceeded": false
                }
            }
        }
    ]
}
```

取得特定資源類型的合規摘要

下列命令會傳回不合規的 EC2 執行個體數目，以及合規的數目：

```
aws configservice get-compliance-summary-by-resource-type --resource-
types AWS::EC2::Instance
```

在輸出中，每個CappedCount屬性的值會指出有多少資源合規或不合規。

輸出：

```
{
    "ComplianceSummariesByResourceType": [
        {
            "ResourceType": "AWS::EC2::Instance",
            "ComplianceSummary": {
                "NonCompliantResourceCount": {
                    "CappedCount": 3,
                    "CapExceeded": false
                },
                "ComplianceSummaryTimestamp": 1452204923.518,
                "CompliantResourceCount": {
                    "CappedCount": 7,
                    "CapExceeded": false
                }
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetComplianceSummaryByResourceType。

get-resource-config-history

以下程式碼範例顯示如何使用 get-resource-config-history。

AWS CLI

取得 AWS 資源的組態歷史記錄

下列命令會傳回 ID 為 之 EC2 執行個體的組態項目清單i-1a2b3c4d：

```
aws configservice get-resource-config-history --resource-type AWS::EC2::Instance --
resource-id i-1a2b3c4d
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResourceConfigHistory。

get-status

以下程式碼範例顯示如何使用 get-status。

AWS CLI

取得 AWS Config 的狀態

下列命令會傳回交付管道和組態記錄器的狀態：

```
aws configservice get-status
```

輸出：

```
Configuration Recorders:

name: default
recorder: ON
last status: SUCCESS

Delivery Channels:

name: default
last stream delivery status: SUCCESS
last history delivery status: SUCCESS
last snapshot delivery status: SUCCESS
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetStatus。

`list-discovered-resources`

以下程式碼範例顯示如何使用 `list-discovered-resources`。

AWS CLI

列出 AWS Config 已探索的資源

下列命令列出 AWS Config 已探索的 EC2 執行個體：

```
aws configservice list-discovered-resources --resource-type AWS::EC2::Instance
```

輸出：

```
{
    "resourceIdentifiers": [
        {
```

```
            "resourceType": "AWS::EC2::Instance",
            "resourceId": "i-1a2b3c4d"
        },
        {
            "resourceType": "AWS::EC2::Instance",
            "resourceId": "i-2a2b3c4d"
        },
        {
            "resourceType": "AWS::EC2::Instance",
            "resourceId": "i-3a2b3c4d"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDiscoveredResources。

put-config-rule

以下程式碼範例顯示如何使用 put-config-rule。

AWS CLI

新增 AWS 受管 Config 規則

下列命令提供 JSON 程式碼來新增 AWS 受管 Config 規則：

```
aws configservice put-config-rule --config-rule file://
RequiredTagsForEC2Instances.json
```

RequiredTagsForEC2Instances.json 是包含規則組態的 JSON 檔案：

```
{
  "ConfigRuleName": "RequiredTagsForEC2Instances",
  "Description": "Checks whether the CostCenter and Owner tags are applied to EC2
 instances.",
  "Scope": {
    "ComplianceResourceTypes": [
      "AWS::EC2::Instance"
    ]
  },
  "Source": {
    "Owner": "AWS",
    "SourceIdentifier": "REQUIRED_TAGS"
```

```
  },
  "InputParameters": "{\"tag1Key\":\"CostCenter\",\"tag2Key\":\"Owner\"}"
}
```

對於 ComplianceResourceTypes 屬性，此 JSON 程式碼會將範圍限制為 AWS::EC2::Instance類型的資源，因此 AWS Config 只會根據規則評估 EC2 執行個體。由於規則是受管規則，Owner屬性會設為 AWS，而SourceIdentifier屬性會設為規則識別符 REQUIRED_TAGS。針對 InputParameters 屬性，會指定規則所需的標籤索引鍵OwnerCostCenter和 。

如果命令成功， AWS Config 不會傳回任何輸出。若要驗證規則組態，請執行 describe-config-rules 命令，並指定規則名稱。

新增客戶受管 Config 規則

下列命令提供 JSON 程式碼來新增客戶受管 Config 規則：

```
aws configservice put-config-rule --config-rule file://InstanceTypesAreT2micro.json
```

InstanceTypesAreT2micro.json 是包含規則組態的 JSON 檔案：

```
{
  "ConfigRuleName": "InstanceTypesAreT2micro",
  "Description": "Evaluates whether EC2 instances are the t2.micro type.",
  "Scope": {
    "ComplianceResourceTypes": [
      "AWS::EC2::Instance"
    ]
  },
  "Source": {
    "Owner": "CUSTOM_LAMBDA",
    "SourceIdentifier": "arn:aws:lambda:us-
east-1:123456789012:function:InstanceTypeCheck",
    "SourceDetails": [
      {
        "EventSource": "aws.config",
        "MessageType": "ConfigurationItemChangeNotification"
      }
    ]
  },
  "InputParameters": "{\"desiredInstanceType\":\"t2.micro\"}"
}
```

對於 ComplianceResourceTypes 屬性，此 JSON 程式碼會將範圍限制為 AWS::EC2::Instance類型的資源，因此 AWS Config 只會根據規則評估 EC2 執行個體。由於此規則是客戶受管規則，Owner屬性設定為 CUSTOM_LAMBDA，SourceIdentifier屬性設定為 AWS Lambda 函數的 ARN。SourceDetails 物件為必要項目。當 Config 調用屬性來根據規則評估資源時，為 InputParameters 屬性指定的參數會傳遞至 AWS Lambda AWS 函數。

如果命令成功， AWS Config 不會傳回任何輸出。若要驗證規則組態，請執行 describe-config-rules 命令，並指定規則名稱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutConfigRule。

put-configuration-recorder

以下程式碼範例顯示如何使用 put-configuration-recorder。

AWS CLI

範例 1：記錄所有支援的資源

下列命令會建立組態記錄器，追蹤所有支援資源類型的變更，包括全域資源類型：

```
aws configservice put-configuration-recorder \
    --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/config-role \
    --recording-group allSupported=true,includeGlobalResourceTypes=true
```

如果命令成功， AWS Config 不會傳回任何輸出。若要驗證組態記錄器的設定，請執行 describe-configuration-recorders 命令。

範例 2：記錄特定類型的資源

下列命令會建立組態記錄器，僅追蹤 --recording-group 選項的 JSON 檔案中指定的資源類型變更：

```
aws configservice put-configuration-recorder \
    --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/config-role \
    --recording-group file://recordingGroup.json
```

recordingGroup.json 是 JSON 檔案，指定 AWS Config 將記錄的資源類型：

```
{
    "allSupported": false,
```

```
    "includeGlobalResourceTypes": false,
    "resourceTypes": [
        "AWS::EC2::EIP",
        "AWS::EC2::Instance",
        "AWS::EC2::NetworkAcl",
        "AWS::EC2::SecurityGroup",
        "AWS::CloudTrail::Trail",
        "AWS::EC2::Volume",
        "AWS::EC2::VPC",
        "AWS::IAM::User",
        "AWS::IAM::Policy"
    ]
}
```

您必須先將 allSupported 和 includeGlobalResourceTypes 選項設定為 false 或省略它們，才能指定 resourceTypes 金鑰的資源類型。 includeGlobalResourceTypes

如果命令成功， AWS Config 不會傳回任何輸出。若要驗證組態記錄器的設定，請執行 describe-configuration-recorders 命令。

範例 3：選取所有支援的資源，不包括特定類型的資源

下列命令會建立組態記錄器，追蹤所有目前和未來支援的資源類型變更，但不包括 --recording-group 選項的 JSON 檔案中指定的這些資源類型：

```
aws configservice put-configuration-recorder \
    --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/
config-role \
    --recording-group file://recordingGroup.json
```

recordingGroup.json 是一種 JSON 檔案，可指定 AWS Config 將記錄的資源類型：

```
{
    "allSupported": false,
    "exclusionByResourceTypes": {
        "resourceTypes": [
        "AWS::Redshift::ClusterSnapshot",
        "AWS::RDS::DBClusterSnapshot",
        "AWS::CloudFront::StreamingDistribution"
    ]
    },
        "includeGlobalResourceTypes": false,
        "recordingStrategy": {
```

```
        "useOnly": "EXCLUSION_BY_RESOURCE_TYPES"
    },
}
```

您必須先將 allSupported 和 includeGlobalResourceTypes 選項設定為 false 或省略它們，以及 2) 您必須將 RecordingStrategy 的 useOnly 欄位設定為 EXCLUSION_BY_RESOURCE_TYPES，才能指定要排除錄製的資源類型：1) 您必須將 allSupported 和 includeGlobalResourceTypes 選項設定為 false 或省略它們。

如果命令成功， AWS Config 不會傳回任何輸出。若要驗證組態記錄器的設定，請執行 describe-configuration-recorders 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutConfigurationRecorder。

put-delivery-channel

以下程式碼範例顯示如何使用 put-delivery-channel。

AWS CLI

建立交付管道

下列命令會以 JSON 程式碼的形式提供交付管道的設定：

```
aws configservice put-delivery-channel --delivery-channel file://
deliveryChannel.json
```

deliveryChannel.json 檔案會指定交付管道屬性：

```
{
    "name": "default",
    "s3BucketName": "config-bucket-123456789012",
    "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",
    "configSnapshotDeliveryProperties": {
        "deliveryFrequency": "Twelve_Hours"
    }
}
```

此範例會設定下列屬性：

name - 交付管道的名稱。根據預設， AWS Config 會將名稱指派給default新的交付管道。您無法使用 put-delivery-channel命令更新交付管道名稱。如需變更名稱的步驟，請參閱重新命名

交付管道。s3BucketName- Config AWS 交付組態快照和組態歷史記錄檔案的 Amazon S3 儲存貯體名稱。如果您指定屬於另一個 AWS 帳戶的儲存貯體，該儲存貯體必須具有將存取許可授予 AWS Config 的政策。如需詳細資訊，請參閱《Amazon S3 儲存貯體許可》。

snsTopicARN - 組態傳送組態變更通知的 Amazon SNS 主題的 Amazon Resource Name (ARN)。如果您從另一個帳戶選擇主題，該主題必須具有授予 Config AWS 存取許可的政策。 AWS 如需詳細資訊，請參閱 Amazon SNS 主題的許可。

configSnapshotDeliveryProperties - 包含 deliveryFrequency 屬性，這會設定 Config AWS 交付組態快照的頻率，以及叫用定期 Config 規則評估的頻率。

如果命令成功， AWS Config 不會傳回任何輸出。若要驗證交付管道的設定，請執行 describe-delivery-channels 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutDeliveryChannel。

start-config-rules-evaluation

以下程式碼範例顯示如何使用 start-config-rules-evaluation。

AWS CLI

執行 Config AWS 規則的隨需評估

下列命令會啟動兩個 AWS 受管規則的評估：

```
aws configservice start-config-rules-evaluation --config-rule-names s3-bucket-versioning-enabled cloudtrail-enabled
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartConfigRulesEvaluation。

start-configuration-recorder

以下程式碼範例顯示如何使用 start-configuration-recorder。

AWS CLI

啟動組態記錄器

下列命令會啟動預設組態記錄器：

```
aws configservice start-configuration-recorder --configuration-recorder-name default
```

如果命令成功，AWS Config 不會傳回任何輸出。若要驗證 AWS Config 正在記錄您的資源，請執行 get-status 命令。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 StartConfigurationRecorder。

stop-configuration-recorder

以下程式碼範例顯示如何使用 stop-configuration-recorder。

AWS CLI

停止組態記錄器

下列命令會停止預設組態記錄器：

```
aws configservice stop-configuration-recorder --configuration-recorder-name default
```

如果命令成功，AWS Config 不會傳回任何輸出。若要驗證 AWS Config 是否未記錄您的資源，請執行 get-status 命令。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 StopConfigurationRecorder。

subscribe

以下程式碼範例顯示如何使用 subscribe。

AWS CLI

訂閱 AWS Config

下列命令會建立預設交付管道和組態記錄器。命令也會指定 Config 將交付組態資訊的 Amazon S3 儲存貯體和 Amazon SNS AWS 主題：

```
aws configservice subscribe --s3-bucket config-bucket-123456789012
 --sns-topic arn:aws:sns:us-east-1:123456789012:config-topic --iam-
role arn:aws:iam::123456789012:role/ConfigRole-A1B2C3D4E5F6
```

輸出：

```
Using existing S3 bucket: config-bucket-123456789012
Using existing SNS topic: arn:aws:sns:us-east-1:123456789012:config-topic
Subscribe succeeded:
```

```
Configuration Recorders: [
    {
        "recordingGroup": {
            "allSupported": true,
            "resourceTypes": [],
            "includeGlobalResourceTypes": false
        },
        "roleARN": "arn:aws:iam::123456789012:role/ConfigRole-A1B2C3D4E5F6",
        "name": "default"
    }
]

Delivery Channels: [
    {
        "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",
        "name": "default",
        "s3BucketName": "config-bucket-123456789012"
    }
]
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 Subscribe。

使用 的 Amazon Connect 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon Connect 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-user`

以下程式碼範例顯示如何使用 `create-user`。

AWS CLI

建立使用者

下列create-user範例會將具有指定屬性的使用者新增至指定的 Amazon Connect 執行個體。

```
aws connect create-user \
    --username Mary \
    --password Pass@Word1 \
    --identity-info FirstName=Mary,LastName=Major \
    --phone-
config PhoneType=DESK_PHONE,AutoAccept=true,AfterContactWorkTimeLimit=60,DeskPhoneNumber=
+15555551212 \
    --security-profile-id 12345678-1111-2222-aaaa-a1b2c3d4f5g7 \
    --routing-profile-id 87654321-9999-3434-abcd-x1y2z3a1b2c3 \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "UserId": "87654321-2222-1234-1234-111234567891",
    "UserArn": "arn:aws:connect:us-west-2:123456789012:instance/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111/agent/87654321-2222-1234-1234-111234567891"
}
```

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的新增使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateUser。

delete-user

以下程式碼範例顯示如何使用 delete-user。

AWS CLI

若要刪除使用者

下列delete-user範例會從指定的 Amazon Connect 執行個體刪除指定的使用者。

```
aws connect delete-user \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --user-id 87654321-2222-1234-1234-111234567891
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的管理使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteUser。

describe-user-hierarchy-group

以下程式碼範例顯示如何使用 describe-user-hierarchy-group。

AWS CLI

顯示階層群組的詳細資訊

下列describe-user-hierarchy-group範例顯示指定 Amazon Connect 階層群組的詳細資訊。

```
aws connect describe-user-hierarchy-group \
    --hierarchy-group-id 12345678-1111-2222-800e-aaabbb555gg \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
     "HierarchyGroup": {
         "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",
         "Arn": "arn:aws:connect:us-west-2:123456789012:instance/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111/agent-group/12345678-1111-2222-800e-a2b3c4d5f6g7",
         "Name": "Example Corporation",
         "LevelId": "1",
         "HierarchyPath": {
             "LevelOne": {
                 "Id": "abcdefgh-3333-4444-8af3-201123456789",
                 "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/agent-group/
abcdefgh-3333-4444-8af3-201123456789",
                 "Name": "Example Corporation"
             }
         }
     }
 }
```

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的設定客服人員階層。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeUserHierarchyGroup。

describe-user-hierarchy-structure

以下程式碼範例顯示如何使用 describe-user-hierarchy-structure。

AWS CLI

顯示階層結構的詳細資訊

下列describe-user-hierarchy-structure範例顯示指定 Amazon Connect 執行個體的階層結構詳細資訊。

```
aws connect describe-user-hierarchy-group \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
     "HierarchyStructure": {
         "LevelOne": {
             "Id": "12345678-1111-2222-800e-aaabbb555gg",
             "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/agent-group-level/1",
             "Name": "Corporation"
         },
         "LevelTwo": {
             "Id": "87654321-2222-3333-ac99-123456789102",
             "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/agent-group-level/2",
             "Name": "Services Division"
         },
         "LevelThree": {
             "Id": "abcdefgh-3333-4444-8af3-201123456789",
             "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/agent-group-level/3",
             "Name": "EU Site"
         }
     }
 }
```

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的設定客服人員階層。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeUserHierarchyStructure。

describe-user

以下程式碼範例顯示如何使用 describe-user。

AWS CLI

顯示使用者的詳細資訊

下列describe-user範例顯示指定 Amazon Connect 使用者的詳細資訊。

```
aws connect describe-user \
    --user-id 0c245dc0-0cf5-4e37-800e-2a7481cc8a60
    --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

輸出：

```
{
    "User": {
        "Id": "0c245dc0-0cf5-4e37-800e-2a7481cc8a60",
        "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/0c245dc0-0cf5-4e37-800e-2a7481cc8a60",
        "Username": "Jane",
        "IdentityInfo": {
            "FirstName": "Jane",
            "LastName": "Doe",
            "Email": "example.com"
        },
        "PhoneConfig": {
            "PhoneType": "SOFT_PHONE",
            "AutoAccept": false,
            "AfterContactWorkTimeLimit": 0,
            "DeskPhoneNumber": ""
        },
        "DirectoryUserId": "8b444cf6-b368-4f29-ba18-07af27405658",
        "SecurityProfileIds": [
            "b6f85a42-1dc5-443b-b621-de0abf70c9cf"
        ],
        "RoutingProfileId": "0be36ee9-2b5f-4ef4-bcf7-87738e5be0e5",
        "Tags": {}
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的管理使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeUser。

`get-contact-attributes`

以下程式碼範例顯示如何使用 `get-contact-attributes`。

AWS CLI

擷取聯絡人的屬性

下列`get-contact-attributes`範例會擷取為指定 Amazon Connect 聯絡人設定的屬性。

```
aws connect get-contact-attributes \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --initial-contact-id 12345678-1111-2222-800e-a2b3c4d5f6g7
```

輸出：

```
{
    "Attributes": {
        "greetingPlayed": "true"
    }
}
```

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的使用 Amazon Connect 聯絡屬性。
Amazon Connect

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetContactAttributes。

`list-contact-flows`

以下程式碼範例顯示如何使用 `list-contact-flows`。

AWS CLI

列出執行個體中的聯絡流程

下列`list-contact-flows`範例列出指定 Amazon Connect 執行個體中的聯絡流程。

```
aws connect list-contact-flows \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
     "ContactFlowSummaryList": [
         {
             "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",
             "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/contact-flow/12345678-1111-2222-800e-
a2b3c4d5f6g7",
             "Name": "Default queue transfer",
             "ContactFlowType": "QUEUE_TRANSFER"
         },
         {
             "Id": "87654321-2222-3333-ac99-123456789102",
             "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/contact-flow/87654321-2222-3333-
ac99-123456789102",
             "Name": "Default agent hold",
             "ContactFlowType": "AGENT_HOLD"
         },
         {
             "Id": "abcdefgh-3333-4444-8af3-201123456789",
                 "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/contact-flow/
abcdefgh-3333-4444-8af3-201123456789",
             "Name": "Default customer hold",
             "ContactFlowType": "CUSTOMER_HOLD"
         },
     ]
 }
```

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的建立 Amazon Connect 聯絡流程。Amazon Connect

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListContactFlows。

list-hours-of-operations

以下程式碼範例顯示如何使用 list-hours-of-operations。

AWS CLI

列出執行個體中的操作時數

下列list-hours-of-operations範例列出指定 Amazon Connect 執行個體的操作時數。

```
aws connect list-hours-of-operations \
    --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

輸出：

```
{
    "HoursOfOperationSummaryList": [
        {
            "Id": "d69f1f84-7457-4924-8fbe-e64875546259",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/operating-hours/d69f1f84-7457-4924-8fbe-e64875546259",
            "Name": "Basic Hours"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的設定佇列的操作時數。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListHoursOfOperations。

list-phone-numbers

以下程式碼範例顯示如何使用 list-phone-numbers。

AWS CLI

列出執行個體中的電話號碼

下列list-phone-numbers範例列出指定 Amazon Connect 執行個體中的電話號碼。

```
aws connect list-phone-numbers \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
```

```
    "PhoneNumberSummaryList": [
        {
            "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/phone-number/xyz80zxy-xyz1-80zx-
zx80-11111EXAMPLE",
            "PhoneNumber": "+17065551212",
            "PhoneNumberType": "DID",
            "PhoneNumberCountryCode": "US"
        },
        {
            "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/phone-number/ccc0ccc-xyz1-80zx-
zx80-22222EXAMPLE",
            "PhoneNumber": "+18555551212",
            "PhoneNumberType": "TOLL_FREE",
            "PhoneNumberCountryCode": "US"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的為您的聯絡中心設定電話號碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPhoneNumbers。

`list-queues`

以下程式碼範例顯示如何使用 `list-queues`。

AWS CLI

列出執行個體中的佇列

下列`list-queues`範例列出指定 Amazon Connect 執行個體中的佇列。

```
aws connect list-queues \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
```

```
    "QueueSummaryList": [
        {
            "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/queue/agent/12345678-1111-2222-800e-
a2b3c4d5f6g7",
            "QueueType": "AGENT"
        },
        {
            "Id": "87654321-2222-3333-ac99-123456789102",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/queue/agent/87654321-2222-3333-
ac99-123456789102",
            "QueueType": "AGENT"
        },
        {
            "Id": "abcdefgh-3333-4444-8af3-201123456789",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/queue/agent/
abcdefgh-3333-4444-8af3-201123456789",
            "QueueType": "AGENT"
        },
        {
            "Id": "hgfedcba-4444-5555-a31f-123456789102",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/queue/hgfedcba-4444-5555-a31f-123456789102",
            "Name": "BasicQueue",
            "QueueType": "STANDARD"
        },
    ]
}
```

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的建立佇列。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListQueues。

`list-routing-profiles`

以下程式碼範例顯示如何使用 `list-routing-profiles`。

AWS CLI

列出執行個體中的路由描述檔

下列list-routing-profiles範例列出指定 Amazon Connect 執行個體中的路由設定檔。

```
aws connect list-routing-profiles \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "RoutingProfileSummaryList": [
        {
            "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/routing-profile/12345678-1111-2222-800e-
a2b3c4d5f6g7",
            "Name": "Basic Routing Profile"
        },
    ]
}
```

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的建立路由設定檔。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRoutingProfiles。

list-security-profiles

以下程式碼範例顯示如何使用 list-security-profiles。

AWS CLI

列出執行個體中的安全性描述檔

下列list-security-profiles範例列出指定 Amazon Connect 執行個體中的安全性設定檔。

```
aws connect list-security-profiles \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "SecurityProfileSummaryList": [
        {
            "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",
```

```
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/security-profile/12345678-1111-2222-800e-
a2b3c4d5f6g7",
            "Name": "CallCenterManager"
        },
        {
            "Id": "87654321-2222-3333-ac99-123456789102",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/security-profile/87654321-2222-3333-
ac99-123456789102",
            "Name": "QualityAnalyst"
        },
        {
            "Id": "abcdefgh-3333-4444-8af3-201123456789",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/security-profile/
abcdefgh-3333-4444-8af3-201123456789",
            "Name": "Agent"
        },
        {
            "Id": "12345678-1111-2222-800e-x2y3c4d5fzzzz",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/security-profile/12345678-1111-2222-800e-
x2y3c4d5fzzzz",
            "Name": "Admin"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的指派許可：安全性設定檔。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSecurityProfiles。

`list-user-hierarchy-groups`

以下程式碼範例顯示如何使用 `list-user-hierarchy-groups`。

AWS CLI

列出執行個體中的使用者階層群組

下列`list-user-hierarchy-groups`範例列出指定 Amazon Connect 執行個體中的使用者階層群組。

```
aws connect list-user-hierarchy-groups \
    --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

輸出：

```
{
    "UserHierarchyGroupSummaryList": [
        {
            "Id": "0e2f6d1d-b3ca-494b-8dbc-ba81d9f8182a",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent-group/0e2f6d1d-b3ca-494b-8dbc-ba81d9f8182a",
            "Name": "Example Corporation"
        },
    ]
}
```

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的設定客服人員階層。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListUserHierarchyGroups。

list-users

以下程式碼範例顯示如何使用 `list-users`。

AWS CLI

列出執行個體中的使用者階層群組

下列`list-users`範例列出指定 Amazon Connect 執行個體中的使用者。

```
aws connect list-users \
    --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

輸出：

```
{
    "UserSummaryList": [
        {
            "Id": "0c245dc0-0cf5-4e37-800e-2a7481cc8a60",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/0c245dc0-0cf5-4e37-800e-2a7481cc8a60",
```

```
            "Username": "Jane"
        },
        {
            "Id": "46f0c67c-3fc7-4806-ac99-403798788c14",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/46f0c67c-3fc7-4806-ac99-403798788c14",
            "Username": "Paulo"
        },
        {
            "Id": "55a83578-95e1-4710-8af3-2b7afe310e48",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/55a83578-95e1-4710-8af3-2b7afe310e48",
            "Username": "JohnD"
        },
        {
            "Id": "703e27b5-c9f0-4f1f-a239-64ccbb160125",
            "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/703e27b5-c9f0-4f1f-a239-64ccbb160125",
            "Username": "JohnS"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的新增使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListUsers。

update-contact-attributes

以下程式碼範例顯示如何使用 update-contact-attributes。

AWS CLI

更新聯絡人的屬性

下列update-contact-attributes範例會更新指定 Amazon Connect 使用者的 greetingPlayed 屬性。

```
aws connect update-contact-attributes \
    --initial-contact-id 11111111-2222-3333-4444-12345678910 \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --attributes greetingPlayed=false
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的使用 Amazon Connect 聯絡屬性。Amazon Connect

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateContactAttributes。

update-user-hierarchy

以下程式碼範例顯示如何使用 update-user-hierarchy。

AWS CLI

更新使用者的階層

下列update-user-hierarchy範例會更新指定 Amazon Connect 使用者的代理程式階層。

```
aws connect update-user-hierarchy \
    --hierarchy-group-id 12345678-a1b2-c3d4-e5f6-123456789abc \
    --user-id 87654321-2222-1234-1234-111234567891 \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的設定客服人員設定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateUserHierarchy。

update-user-identity-info

以下程式碼範例顯示如何使用 update-user-identity-info。

AWS CLI

更新使用者的身分資訊

下列update-user-identity-info範例會更新指定 Amazon Connect 使用者的身分資訊。

```
aws connect update-user-identity-info \
    --identity-info FirstName=Mary,LastName=Major,Email=marym@example.com \
    --user-id 87654321-2222-1234-1234-111234567891 \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的設定客服人員設定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateUserIdentityInfo。

update-user-phone-config

以下程式碼範例顯示如何使用 update-user-phone-config。

AWS CLI

更新使用者的電話組態

下列update-user-phone-config範例會更新指定使用者的電話組態。

```
aws connect update-user-phone-config \
    --phone-
config PhoneType=SOFT_PHONE,AutoAccept=false,AfterContactWorkTimeLimit=60,DeskPhoneNumber=
+18005551212 \
    --user-id 12345678-4444-3333-2222-111122223333 \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的設定客服人員設定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateUserPhoneConfig。

update-user-routing-profile

以下程式碼範例顯示如何使用 update-user-routing-profile。

AWS CLI

更新使用者的路由描述檔

下列update-user-routing-profile範例會更新指定 Amazon Connect 使用者的路由描述檔。

```
aws connect update-user-routing-profile \
    --routing-profile-id 12345678-1111-3333-2222-4444EXAMPLE \
    --user-id 87654321-2222-1234-1234-111234567891 \
```

```
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的設定客服人員設定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateUserRoutingProfile。

update-user-security-profiles

以下程式碼範例顯示如何使用 update-user-security-profiles。

AWS CLI

更新使用者的安全性描述檔

下列update-user-security-profiles範例會更新指定 Amazon Connect 使用者的安全性描述檔。

```
aws connect update-user-security-profiles \
    --security-profile-ids 12345678-1234-1234-1234-1234567892111 \
    --user-id 87654321-2222-1234-1234-111234567891 \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Connect 管理員指南》中的指派許可：安全性設定檔。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateUserSecurityProfiles。

AWS Cost and Usage Report 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS Cost and Usage Report。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

delete-report-definition

以下程式碼範例顯示如何使用 delete-report-definition。

AWS CLI

刪除 AWS 成本和用量報告

此範例會刪除 AWS 成本和用量報告。

命令：

```
aws cur --region us-east-1 delete-report-definition --report-name "ExampleReport"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteReportDefinition。

describe-report-definitions

以下程式碼範例顯示如何使用 describe-report-definitions。

AWS CLI

擷取 AWS 成本和用量報告清單

此範例說明 帳戶擁有 AWS 的成本和用量報告清單。

命令：

```
aws cur --region us-east-1 describe-report-definitions --max-items 5
```

輸出：

```
  {
"ReportDefinitions": [
  {
      "ReportName": "ExampleReport",
```

```
      "Compression": "ZIP",
      "S3Region": "us-east-1",
      "Format": "textORcsv",
      "S3Prefix": "exampleprefix",
      "S3Bucket": "example-s3-bucket",
      "TimeUnit": "DAILY",
      "AdditionalArtifacts": [
          "REDSHIFT",
          "QUICKSIGHT"
      ],
      "AdditionalSchemaElements": [
          "RESOURCES"
      ]
  }
]
  }
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReportDefinitions。

put-report-definition

以下程式碼範例顯示如何使用 put-report-definition。

AWS CLI

建立 AWS 成本和用量報告

下列put-report-definition範例會建立每日 AWS 成本和用量報告，您可以將其上傳至 Amazon Redshift 或 Amazon QuickSight。

```
aws cur put-report-definition --report-definition file://report-definition.json
```

report-definition.json 的內容：

```
{
    "ReportName": "ExampleReport",
    "TimeUnit": "DAILY",
    "Format": "textORcsv",
    "Compression": "ZIP",
    "AdditionalSchemaElements": [
        "RESOURCES"
    ],
```

```
    "S3Bucket": "example-s3-bucket",
    "S3Prefix": "exampleprefix",
    "S3Region": "us-east-1",
    "AdditionalArtifacts": [
        "REDSHIFT",
        "QUICKSIGHT"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutReportDefinition。

使用 的 Cost Explorer Service 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Cost Explorer Service 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`get-cost-and-usage`

以下程式碼範例顯示如何使用 `get-cost-and-usage`。

AWS CLI

擷取 2017 年 9 月帳戶的 S3 用量

下列`get-cost-and-usage`範例會擷取 2017 年 9 月帳戶的 S3 用量。

```
aws ce get-cost-and-usage \
    --time-period Start=2017-09-01,End=2017-10-01 \
    --granularity MONTHLY \
    --metrics "BlendedCost" "UnblendedCost" "UsageQuantity" \
```

```
    --group-by Type=DIMENSION,Key=SERVICE Type=TAG,Key=Environment \
    --filter file://filters.json
```

filters.json 的內容：

```
{
    "Dimensions": {
        "Key": "SERVICE",
        "Values": [
            "Amazon Simple Storage Service"
        ]
    }
}
```

輸出：

```
{
    "GroupDefinitions": [
        {
            "Type": "DIMENSION",
            "Key": "SERVICE"
        },
        {
            "Type": "TAG",
            "Key": "Environment"
        }
    ],
    "ResultsByTime": [
        {
            "Estimated": false,
            "TimePeriod": {
                "Start": "2017-09-01",
                "End": "2017-10-01"
            },
            "Total": {},
            "Groups": [
                {
                    "Keys": [
                        "Amazon Simple Storage Service",
                        "Environment$"
                    ],
                    "Metrics": {
                        "BlendedCost": {
```

```
                            "Amount": "40.3527508453",
                            "Unit": "USD"
                        },
                        "UnblendedCost": {
                            "Amount": "40.3543773134",
                            "Unit": "USD"
                        },
                        "UsageQuantity": {
                            "Amount": "9312771.098461578",
                            "Unit": "N/A"
                        }
                    }
                },
                {
                    "Keys": [
                        "Amazon Simple Storage Service",
                        "Environment$Dev"
                    ],
                    "Metrics": {
                        "BlendedCost": {
                            "Amount": "0.2682364644",
                            "Unit": "USD"
                        },
                        "UnblendedCost": {
                            "Amount": "0.2682364644",
                            "Unit": "USD"
                        },
                        "UsageQuantity": {
                            "Amount": "22403.4395271182",
                            "Unit": "N/A"
                        }
                    }
                }
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCostAndUsage。

get-dimension-values

以下程式碼範例顯示如何使用 get-dimension-values。

AWS CLI

若要擷取維度 SERVICE 的標籤，值為 "Elastic"

此範例會擷取維度 SERVICE 的標籤，其值為「彈性」，適用於 2017 年 1 月 1 日至 2017 年 5 月 18 日。

命令：

```
aws ce get-dimension-values --search-string Elastic --time-
period Start=2017-01-01,End=2017-05-18 --dimension SERVICE
```

輸出：

```
{
   "TotalSize": 6,
   "DimensionValues": [
       {
           "Attributes": {},
           "Value": "Amazon ElastiCache"
       },
       {
           "Attributes": {},
           "Value": "EC2 - Other"
       },
       {
           "Attributes": {},
           "Value": "Amazon Elastic Compute Cloud - Compute"
       },
       {
           "Attributes": {},
           "Value": "Amazon Elastic Load Balancing"
       },
       {
           "Attributes": {},
           "Value": "Amazon Elastic MapReduce"
       },
       {
           "Attributes": {},
           "Value": "Amazon Elasticsearch Service"
       }
   ],
   "ReturnSize": 6
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDimensionValues。

get-reservation-coverage

以下程式碼範例顯示如何使用 get-reservation-coverage。

AWS CLI

擷取 us-east-1 區域中 EC2 t2.nano 執行個體的保留涵蓋範圍

此範例會擷取 2017 年 7 月至 9 月 us-east-1 區域中 EC2 t2.nano 執行個體的保留涵蓋範圍。

命令：

```
aws ce get-reservation-coverage --time-period Start=2017-07-01,End=2017-10-01 --
group-by Type=Dimension,Key=REGION --filter file://filters.json
```

filter.json：

```
{
   "And": [
     {
       "Dimensions": {
         "Key": "INSTANCE_TYPE",
         "Values": [
           "t2.nano"
         ]
       },
       "Dimensions": {
         "Key": "REGION",
         "Values": [
           "us-east-1"
         ]
       }
     }
   ]
 }
```

輸出：

```
{
   "TotalSize": 6,
   "DimensionValues": [
       {
           "Attributes": {},
           "Value": "Amazon ElastiCache"
       },
       {
           "Attributes": {},
           "Value": "EC2 - Other"
       },
       {
           "Attributes": {},
           "Value": "Amazon Elastic Compute Cloud - Compute"
       },
       {
           "Attributes": {},
           "Value": "Amazon Elastic Load Balancing"
       },
       {
           "Attributes": {},
           "Value": "Amazon Elastic MapReduce"
       },
       {
           "Attributes": {},
           "Value": "Amazon Elasticsearch Service"
       }
   ],
   "ReturnSize": 6
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetReservationCoverage。

get-reservation-purchase-recommendation

以下程式碼範例顯示如何使用 get-reservation-purchase-recommendation。

AWS CLI

擷取三年期間的部分預付 EC2 RIs 的保留建議

下列get-reservation-purchase-recommendation範例會根據過去 60 天的 EC2 用量，擷取三年期間的部分預付 EC2 執行個體建議。

```
aws ce get-reservation-purchase-recommendation \
    --service "Amazon Redshift" \
    --lookback-period-in-days SIXTY_DAYS \
    --term-in-years THREE_YEARS \
    --payment-option PARTIAL_UPFRONT
```

輸出：

```
{
    "Recommendations": [],
    "Metadata": {
        "GenerationTimestamp": "2018-08-08T15:20:57Z",
        "RecommendationId": "00d59dde-a1ad-473f-8ff2-iexample3330b"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetReservationPurchaseRecommendation。

get-reservation-utilization

以下程式碼範例顯示如何使用 get-reservation-utilization。

AWS CLI

擷取您帳戶的保留使用率

下列get-reservation-utilization範例會擷取帳戶從 2018-03-01 到 2018-08-01 的所有 t2.nano 執行個體類型的 RI 使用率。

```
aws ce get-reservation-utilization \
    --time-period Start=2018-03-01,End=2018-08-01 \
    --filter file://filters.json
```

filters.json 的內容：

```
{
```

```
    "Dimensions": {
        "Key": "INSTANCE_TYPE",
        "Values": [
            "t2.nano"
        ]
    }
}
```

輸出：

```
{
    "Total": {
        "TotalAmortizedFee": "0",
        "UtilizationPercentage": "0",
        "PurchasedHours": "0",
        "NetRISavings": "0",
        "TotalActualHours": "0",
        "AmortizedRecurringFee": "0",
        "UnusedHours": "0",
        "TotalPotentialRISavings": "0",
        "OnDemandCostOfRIHoursUsed": "0",
        "AmortizedUpfrontFee": "0"
    },
    "UtilizationsByTime": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetReservationUtilization。

get-tags

以下程式碼範例顯示如何使用 get-tags。

AWS CLI

擷取成本分配標籤的索引鍵和值

此範例會擷取鍵為 "Project" 且值包含 "secretProject" 的所有成本分配標籤。

命令：

```
aws ce get-tags --search-string secretProject --time-
period Start=2017-01-01,End=2017-05-18 --tag-key Project
```

輸出：

```
{
  "ReturnSize": 2,
  "Tags": [
    "secretProject1",
    "secretProject2"
  ],
  "TotalSize": 2
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTags。

使用 的 Firehose 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Firehose 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`list-delivery-streams`

以下程式碼範例顯示如何使用 `list-delivery-streams`。

AWS CLI

列出可用的交付串流

下列`list-delivery-streams`範例列出您 AWS 帳戶中可用的交付串流。

```
aws firehose list-delivery-streams
```

輸出：

```
{
    "DeliveryStreamNames": [
        "my-stream"
    ],
    "HasMoreDeliveryStreams": false
}
```

如需詳細資訊，請參閱《Amazon Kinesis Data Firehose 開發人員指南》中的建立 Amazon Kinesis Data Firehose 交付串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDeliveryStreams。

put-record-batch

以下程式碼範例顯示如何使用 put-record-batch。

AWS CLI

將多筆記錄寫入串流

下列put-record-batch範例會將三個記錄寫入串流。資料以 Base64 格式編碼。

```
aws firehose put-record-batch \
    --delivery-stream-name my-stream \
    --records file://records.json
```

myfile.json 的內容：

```
[
    {"Data": "Rmlyc3QgdGhpbmc="},
    {"Data": "U2Vjb25kIHRoaW5n"},
    {"Data": "VGhpcmQgdGhpbmc="}
]
```

輸出：

```
{
    "FailedPutCount": 0,
    "Encrypted": false,
    "RequestResponses": [
        {
```

```
            "RecordId": "9D2OJ6t2EqCTZTXwGzeSv/EVHxRoRCw89xd+o3+sXg8DhYOaWKPSmZy/
CGlRVEys1u1xbeKh6VofEYKkoeiDrcjrxhQp9iF7sUW7pujiMEQ5LzlrzCkGosxQn
+3boDnURDEaD42V7GiixpOyLJkYZcae1i7HzlCEoy9LJhMr8EjDSi4Om/9Vc2uhwwuAtGE0XKpxJ2WD7ZRWtAnYlKAnv
        },
        {
            "RecordId": "jFirejqxCLlK5xjH/UNmlMVcjktEN76I7916X9PaZ
+PVaOSXDfU1WGOqEZhxq2js7xcZ552eoeDxsuTU1MSq9nZTbVfb6cQTIXnm/GsuF37Uhg67GKmR5z90l6XKJ
+/+pDloFv7Hh9a3oUS6wYm3DcNRLTHHAimANp1PhkQvWpvLRfzbuCUkBphR2QVzhP9OiHLbzGwy8/
DfH8sqWEUYASNJKS8GXP5s"
        },
        {
            "RecordId":
 "oy0amQ40o5Y2YV4vxzufdcMOOw6n3EPr3tpPJGoYVNKH4APPVqNcbUgefo1stEFRg4hTLrf2k6eliHu/9+YJ5R3iie
DTBt3qBlmTj7Xq8SKVbO1S7YvMTpWkMKA86f8JfmT8BMKoMb4XZS/sOkQLe+qh0sYKXWl"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Kinesis Data Firehose 開發人員指南》中的將資料傳送至 Amazon Kinesis Data Firehose 交付串流。 Amazon Kinesis Data Firehose

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutRecordBatch。

put-record

以下程式碼範例顯示如何使用 put-record。

AWS CLI

將記錄寫入串流

下列put-record範例會將資料寫入串流。資料以 Base64 格式編碼。

```
aws firehose put-record \
    --delivery-stream-name my-stream \
    --record '{"Data":"SGVsbG8gd29ybGQ="}'
```

輸出：

```
{
    "RecordId": "RjB5K/nnoGFHqwTsZlNd/
TTqvjE8V5dsyXZTQn2JXrdpMTOwssyEb6nfC8fwf1whhwnItt4mvrn+gsqeK5jB7QjuLg283+Ps4Sz/
j1Xujv31iDhnPdaLw4BOyM9Amv7PcCuB2079RuM0NhoakbyUymlwY8yt20G8X2420wu1jlFafhci4erAt7QhDEvpwuK8
```

```
    "Encrypted": false
}
```

如需詳細資訊，請參閱《Amazon Kinesis Data Firehose 開發人員指南》中的將資料傳送至 Amazon Kinesis Data Firehose 交付串流。 Amazon Kinesis Data Firehose

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutRecord。

使用 的 Amazon Data Lifecycle Manager 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon Data Lifecycle Manager 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-default-role`

以下程式碼範例顯示如何使用 `create-default-role`。

AWS CLI

建立 Amazon DLM 所需的 IAM 角色

下列`dlm create-default-role`範例會建立用於管理快照的 AWS DataLifecycleManagerDefaultRole 預設角色。

```
aws dlm create-default-role \
    --resource-type snapshot
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的 Amazon Data Lifecycle Manager 的預設服務角色。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDefaultRole。

create-lifecycle-policy

以下程式碼範例顯示如何使用 create-lifecycle-policy。

AWS CLI

建立生命週期政策

下列create-lifecycle-policy範例會建立生命週期政策，以在指定時間建立磁碟區的每日快照。指定的標籤會新增至快照，而標籤也會從磁碟區複製並新增至快照。如果建立新的快照超過指定的計數上限，則會刪除最舊的快照。

```
aws dlm create-lifecycle-policy \
    --description "My first policy" \
    --state ENABLED \
    --execution-role-arn arn:aws:iam::12345678910:role/
AWSDataLifecycleManagerDefaultRole \
    --policy-details file://policyDetails.json
```

policyDetails.json 的內容：

```
{
    "ResourceTypes": [
        "VOLUME"
    ],
    "TargetTags": [
        {
            "Key": "costCenter",
            "Value": "115"
        }
    ],
    "Schedules":[
        {
            "Name": "DailySnapshots",
            "CopyTags": true,
            "TagsToAdd": [
                {
                    "Key": "type",
                     "Value": "myDailySnapshot"
                }
```

```
            ],
            "CreateRule": {
                "Interval": 24,
                "IntervalUnit": "HOURS",
                "Times": [
                    "03:00"
                ]
            },
            "RetainRule": {
                "Count":5
            }
        }
    ]
}
```

輸出：

```
{
    "PolicyId": "policy-0123456789abcdef0"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLifecyclePolicy。

delete-lifecycle-policy

以下程式碼範例顯示如何使用 delete-lifecycle-policy。

AWS CLI

刪除生命週期政策

下列範例會刪除指定的生命週期政策：

```
aws dlm delete-lifecycle-policy --policy-id policy-0123456789abcdef0
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLifecyclePolicy。

get-lifecycle-policies

以下程式碼範例顯示如何使用 get-lifecycle-policies。

AWS CLI

取得生命週期政策的摘要

下列get-lifecycle-policies範例列出所有生命週期政策。

```
aws dlm get-lifecycle-policies
```

輸出：

```
{
    "Policies": [
        {
            "PolicyId": "policy-0123456789abcdef0",
            "Description": "My first policy",
            "State": "ENABLED"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLifecyclePolicies。

get-lifecycle-policy

以下程式碼範例顯示如何使用 get-lifecycle-policy。

AWS CLI

描述生命週期政策

下列get-lifecycle-policy範例顯示指定生命週期政策的詳細資訊。

```
aws dlm get-lifecycle-policy \
    --policy-id policy-0123456789abcdef0
```

輸出：

```
{
    "Policy": {
        "PolicyId": "policy-0123456789abcdef0",
```

```
        "Description": "My policy",
        "State": "ENABLED",
        "ExecutionRoleArn": "arn:aws:iam::123456789012:role/
AWSDataLifecycleManagerDefaultRole",
        "DateCreated": "2019-08-08T17:45:42Z",
        "DateModified": "2019-08-08T17:45:42Z",
        "PolicyDetails": {
            "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
            "ResourceTypes": [
                "VOLUME"
            ],
            "TargetTags": [
              {
                  "Key": "costCenter",
                  "Value": "115"
              }
            ],
            "Schedules": [
              {
                  "Name": "DailySnapshots",
                  "CopyTags": true,
                  "TagsToAdd": [
                    {
                        "Key": "type",
                        "Value": "myDailySnapshot"
                    }
                  ],
                  "CreateRule": {
                    "Interval": 24,
                    "IntervalUnit": "HOURS",
                    "Times": [
                        "03:00"
                    ]
                  },
                  "RetainRule": {
                    "Count": 5
                  }
              }
            ]
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLifecyclePolicy。

update-lifecycle-policy

以下程式碼範例顯示如何使用 update-lifecycle-policy。

AWS CLI

範例 1：啟用生命週期政策

下列update-lifecycle-policy範例會啟用指定的生命週期政策。

```
aws dlm update-lifecycle-policy \
    --policy-id policy-0123456789abcdef0 \
    --state ENABLED
```

範例 2：停用生命週期政策

下列update-lifecycle-policy範例會停用指定的生命週期政策。

```
aws dlm update-lifecycle-policy \
    --policy-id policy-0123456789abcdef0 \
    --state DISABLED
```

範例 3：更新生命週期政策的詳細資訊

下列update-lifecycle-policy範例會更新指定生命週期政策的目標標籤。

```
aws dlm update-lifecycle-policy \
    --policy-id policy-0123456789abcdef0
    --policy-details file://policyDetails.json
```

policyDetails.json 的內容。命令不會變更此檔案中未參考的其他詳細資訊。

```
{
    "TargetTags": [
        {
            "Key": "costCenter",
            "Value": "120"
        },
        {
            "Key": "project",
            "Value": "lima"
```

```
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateLifecyclePolicy。

AWS Data Pipeline 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS Data Pipeline。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`activate-pipeline`

以下程式碼範例顯示如何使用 `activate-pipeline`。

AWS CLI

啟用管道

此範例會啟用指定的管道：

```
aws datapipeline activate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

若要在特定日期和時間啟用管道，請使用下列命令：

```
aws datapipeline activate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE --start-
timestamp 2015-04-07T00:00:00Z
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ActivatePipeline。

add-tags

以下程式碼範例顯示如何使用 add-tags。

AWS CLI

將標籤新增至管道

此範例會將指定的標籤新增至指定的管道：

```
aws datapipeline add-tags --pipeline-id df-00627471SOVYZEXAMPLE --
tags key=environment,value=production key=owner,value=sales
```

若要檢視標籤，請使用 describe-pipelines 命令。例如，範例命令中新增的標籤會顯示在 describe-pipelines 的輸出中，如下所示：

```
{
    ...
        "tags": [
            {
                "value": "production",
                "key": "environment"
            },
            {
                "value": "sales",
                "key": "owner"
            }
        ]
    ...
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddTags。

create-pipeline

以下程式碼範例顯示如何使用 create-pipeline。

AWS CLI

建立管道

此範例會建立管道：

```
aws datapipeline create-pipeline --name my-pipeline --unique-id my-pipeline-token
```

下列為範例輸出：

```
{
    "pipelineId": "df-00627471SOVYZEXAMPLE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePipeline。

deactivate-pipeline

以下程式碼範例顯示如何使用 deactivate-pipeline。

AWS CLI

停用管道

此範例會停用指定的管道：

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

若要在所有執行中的活動完成後停用管道，請使用下列命令：

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE --no-
cancel-active
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeactivatePipeline。

delete-pipeline

以下程式碼範例顯示如何使用 delete-pipeline。

AWS CLI

刪除管道

此範例會刪除指定的管道：

```
aws datapipeline delete-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePipeline。

describe-pipelines

以下程式碼範例顯示如何使用 describe-pipelines。

AWS CLI

描述您的管道

此範例說明指定的管道：

```
aws datapipeline describe-pipelines --pipeline-ids df-00627471SOVYZEXAMPLE
```

下列為範例輸出：

```
{
  "pipelineDescriptionList": [
      {
          "fields": [
              {
                  "stringValue": "PENDING",
                  "key": "@pipelineState"
              },
              {
                  "stringValue": "my-pipeline",
                  "key": "name"
              },
              {
                  "stringValue": "2015-04-07T16:05:58",
                  "key": "@creationTime"
              },
              {
                  "stringValue": "df-00627471SOVYZEXAMPLE",
                  "key": "@id"
              },
              {
                  "stringValue": "123456789012",
                  "key": "pipelineCreator"
              },
              {
                  "stringValue": "PIPELINE",
                  "key": "@sphere"
```

```
            },
            {
                "stringValue": "123456789012",
                "key": "@userId"
            },
            {
                "stringValue": "123456789012",
                "key": "@accountId"
            },
            {
                "stringValue": "my-pipeline-token",
                "key": "uniqueId"
            }
        ],
        "pipelineId": "df-00627471SOVYZEXAMPLE",
        "name": "my-pipeline",
        "tags": []
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePipelines。

get-pipeline-definition

以下程式碼範例顯示如何使用 get-pipeline-definition。

AWS CLI

取得管道定義

此範例會取得指定管道的管道定義：

```
aws datapipeline get-pipeline-definition --pipeline-id df-00627471SOVYZEXAMPLE
```

下列為範例輸出：

```
{
  "parameters": [
      {
          "type": "AWS::S3::ObjectKey",
          "id": "myS3OutputLoc",
          "description": "S3 output folder"
```

```
      },
      {
          "default": "s3://us-east-1.elasticmapreduce.samples/pig-apache-logs/data",
          "type": "AWS::S3::ObjectKey",
          "id": "myS3InputLoc",
          "description": "S3 input folder"
      },
      {
          "default": "grep -rc \"GET\" ${INPUT1_STAGING_DIR}/* >
 ${OUTPUT1_STAGING_DIR}/output.txt",
          "type": "String",
          "id": "myShellCmd",
          "description": "Shell command to run"
      }
  ],
  "objects": [
      {
          "type": "Ec2Resource",
          "terminateAfter": "20 Minutes",
          "instanceType": "t1.micro",
          "id": "EC2ResourceObj",
          "name": "EC2ResourceObj"
      },
      {
          "name": "Default",
          "failureAndRerunMode": "CASCADE",
          "resourceRole": "DataPipelineDefaultResourceRole",
          "schedule": {
              "ref": "DefaultSchedule"
          },
          "role": "DataPipelineDefaultRole",
          "scheduleType": "cron",
          "id": "Default"
      },
      {
          "directoryPath": "#{myS3OutputLoc}/#{format(@scheduledStartTime, 'YYYY-MM-
dd-HH-mm-ss')}",
          "type": "S3DataNode",
          "id": "S3OutputLocation",
          "name": "S3OutputLocation"
      },
      {
          "directoryPath": "#{myS3InputLoc}",
          "type": "S3DataNode",
```

```
            "id": "S3InputLocation",
            "name": "S3InputLocation"
        },
        {
            "startAt": "FIRST_ACTIVATION_DATE_TIME",
            "name": "Every 15 minutes",
            "period": "15 minutes",
            "occurrences": "4",
            "type": "Schedule",
            "id": "DefaultSchedule"
        },
        {
            "name": "ShellCommandActivityObj",
            "command": "#{myShellCmd}",
            "output": {
                "ref": "S3OutputLocation"
            },
            "input": {
                "ref": "S3InputLocation"
            },
            "stage": "true",
            "type": "ShellCommandActivity",
            "id": "ShellCommandActivityObj",
            "runsOn": {
                "ref": "EC2ResourceObj"
            }
        }
    ],
    "values": {
        "myS3OutputLoc": "s3://amzn-s3-demo-bucket/",
        "myS3InputLoc": "s3://us-east-1.elasticmapreduce.samples/pig-apache-logs/
data",
        "myShellCmd": "grep -rc \"GET\" ${INPUT1_STAGING_DIR}/* >
 ${OUTPUT1_STAGING_DIR}/output.txt"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPipelineDefinition。

list-pipelines

以下程式碼範例顯示如何使用 list-pipelines。

AWS CLI

列出您的管道

此範例列出您的管道：

```
aws datapipeline list-pipelines
```

下列為範例輸出：

```
{
  "pipelineIdList": [
      {
          "id": "df-00627471SOVYZEXAMPLE",
          "name": "my-pipeline"
      },
      {
          "id": "df-09028963KNVMREXAMPLE",
          "name": "ImportDDB"
      },
      {
          "id": "df-0870198233ZYVEXAMPLE",
          "name": "CrossRegionDDB"
      },
      {
          "id": "df-00189603TB4MZEXAMPLE",
          "name": "CopyRedshift"
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPipelines。

list-runs

以下程式碼範例顯示如何使用 `list-runs`。

AWS CLI

範例 1：列出您的管道執行

下列`list-runs`範例列出指定管道的執行。

```
aws datapipeline list-runs --pipeline-id df-00627471SOVYZEXAMPLE
```

輸出：

```
    Name                        Scheduled Start        Status                      ID
                                              Started                Ended

 ---------------------------------------------------------------------------------------
1.  EC2ResourceObj              2015-04-12T17:33:02    CREATING
 @EC2ResourceObj_2015-04-12T17:33:02             2015-04-12T17:33:10
2.  S3InputLocation             2015-04-12T17:33:02    FINISHED
      @S3InputLocation_2015-04-12T17:33:02            2015-04-12T17:33:09
 2015-04-12T17:33:09
3.  S3OutputLocation            2015-04-12T17:33:02    WAITING_ON_DEPENDENCIES
 @S3OutputLocation_2015-04-12T17:33:02           2015-04-12T17:33:09
4.  ShellCommandActivityObj     2015-04-12T17:33:02    WAITING_FOR_RUNNER
 @ShellCommandActivityObj_2015-04-12T17:33:02    2015-04-12T17:33:09
```

範例 2：列出指定日期之間的管道執行

下列`list-runs`範例使用 `--start-interval`來指定要包含在輸出中的日期。

```
aws datapipeline list-runs --pipeline-id df-01434553B58A2SHZUKO5 --start-
interval 2017-10-07T00:00:00,2017-10-08T00:00:00
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRuns。

put-pipeline-definition

以下程式碼範例顯示如何使用 `put-pipeline-definition`。

AWS CLI

上傳管道定義

此範例會將指定的管道定義上傳至指定的管道：

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471SOVYZEXAMPLE --
pipeline-definition file://my-pipeline-definition.json
```

下列為範例輸出：

```
{
  "validationErrors": [],
  "errored": false,
  "validationWarnings": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutPipelineDefinition。

`remove-tags`

以下程式碼範例顯示如何使用 `remove-tags`。

AWS CLI

從管道移除標籤

此範例會從指定的管道移除指定的標籤：

```
aws datapipeline remove-tags --pipeline-id df-00627471SOVYZEXAMPLE --tag-
keys environment
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveTags。

使用 的 DataSync 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 DataSync 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

update-location-azure-blob

以下程式碼範例顯示如何使用 update-location-azure-blob。

AWS CLI

使用新的客服人員更新您的轉接位置

下列update-location-object-storage範例會使用新的代理程式更新 Microsoft Azure Blob Storage 的 DataSync 位置。

```
aws datasync update-location-azure-blob \
    --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-
abcdef01234567890 \
    --agent-arns arn:aws:datasync:us-west-2:123456789012:agent/
agent-1234567890abcdef0 \
    --sas-configuration '{ \
        "Token": "sas-token-for-azure-blob-storage-access" \
    }'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS DataSync 使用者指南中的取代您的代理程式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateLocationAzureBlob。

update-location-hdfs

以下程式碼範例顯示如何使用 update-location-hdfs。

AWS CLI

使用新的客服人員更新您的轉接位置

下列update-location-hdfs範例會使用新的代理程式更新您的 DataSync HDFS 位置。只有在 HDFS 叢集使用 Kerberos 身分驗證時，才需要 --kerberos-keytab和 --kerberos-krb5-conf選項。

```
aws datasync update-location-hdfs \
    --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-
abcdef01234567890 \
```

```
    --agent-arns arn:aws:datasync:us-west-2:123456789012:agent/
agent-1234567890abcdef0 \
    --kerberos-keytab file://hdfs.keytab
    --kerberos-krb5-conf file://krb5.conf
```

hdfs.keytab 的內容：

```
N/A. The content of this file is encrypted and not human readable.
```

krb5.conf 的內容：

```
[libdefaults]
    default_realm = EXAMPLE.COM
    dns_lookup_realm = false
    dns_lookup_kdc = false
    rdns = true
    ticket_lifetime = 24h
    forwardable = true
    udp_preference_limit = 1000000
    default_tkt_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 des3-cbc-
sha1
    default_tgs_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 des3-cbc-
sha1
    permitted_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 des3-cbc-
sha1

[realms]
    EXAMPLE.COM = {
      kdc = kdc1.example.com
      admin_server = krbadmin.example.com
      default_domain = example.com
    }

[domain_realm]
    .example.com = EXAMPLE.COM
    example.com = EXAMPLE.COM

[logging]
    kdc = FILE:/var/log/krb5kdc.log
    admin_server = FILE:/var/log/kerberos/kadmin.log
    default = FILE:/var/log/krb5libs.log
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS DataSync 使用者指南中的取代您的代理程式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateLocationHdfs。

update-location-nfs

以下程式碼範例顯示如何使用 update-location-nfs。

AWS CLI

使用新的客服人員更新您的轉接位置

下列update-location-nfs範例會使用新的代理程式更新您的 DataSync NFS 位置。

```
aws datasync update-location-nfs \
    --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-
abcdef01234567890 \
    --on-prem-config AgentArns=arn:aws:datasync:us-west-2:123456789012:agent/
agent-1234567890abcdef0
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS DataSync 使用者指南中的取代您的代理程式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateLocationNfs。

update-location-object-storage

以下程式碼範例顯示如何使用 update-location-object-storage。

AWS CLI

使用新的客服人員更新您的轉接位置

下列update-location-object-storage範例會使用新的代理程式更新 DataSync 物件儲存位置。

```
aws datasync update-location-object-storage \
    --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-
abcdef01234567890 \
    --agent-arns arn:aws:datasync:us-west-2:123456789012:agent/
agent-1234567890abcdef0 \
```

```
    --secret-key secret-key-for-object-storage
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS DataSync 使用者指南中的取代您的代理程式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateLocationObjectStorage。

update-location-smb

以下程式碼範例顯示如何使用 update-location-smb。

AWS CLI

使用新的客服人員更新您的轉接位置

下列update-location-smb範例會使用新的代理程式更新您的 DataSync SMB 位置。

```
aws datasync update-location-smb \
    --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-
abcdef01234567890 \
    --agent-arns arn:aws:datasync:us-west-2:123456789012:agent/
agent-1234567890abcdef0 \
    --password smb-file-server-password
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS DataSync 使用者指南中的取代您的代理程式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateLocationSmb。

使用 的 DAX 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 DAX 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-cluster`

以下程式碼範例顯示如何使用 `create-cluster`。

AWS CLI

建立 DAX 叢集

下列`create-cluster`範例會建立具有指定設定的 DAX 叢集。

```
aws dax create-cluster \
    --cluster-name daxcluster \
    --node-type dax.r4.large \
    --replication-factor 3 \
    --iam-role-arn roleARN  \
    --sse-specification Enabled=true
```

輸出：

```
{
    "Cluster": {
        "ClusterName": "daxcluster",
        "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",
        "TotalNodes": 3,
        "ActiveNodes": 0,
        "NodeType": "dax.r4.large",
        "Status": "creating",
        "ClusterDiscoveryEndpoint": {
            "Port": 8111
        },
        "PreferredMaintenanceWindow": "thu:13:00-thu:14:00",
        "SubnetGroup": "default",
        "SecurityGroups": [
            {
                "SecurityGroupIdentifier": "sg-1af6e36e",
                "Status": "active"
            }
        ],
        "IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
        "ParameterGroup": {
            "ParameterGroupName": "default.dax1.0",
```

```
            "ParameterApplyStatus": "in-sync",
            "NodeIdsToReboot": []
        },
        "SSEDescription": {
            "Status": "ENABLED"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的步驟 3：建立 DAX 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCluster。

create-parameter-group

以下程式碼範例顯示如何使用 create-parameter-group。

AWS CLI

建立參數群組

下列 `` create-parameter-group`` 範例會建立具有指定設定的參數群組。

```
aws dax create-parameter-group \
    --parameter-group-name daxparametergroup \
    --description "A new parameter group"
```

輸出：

```
{
    "ParameterGroup": {
        "ParameterGroupName": "daxparametergroup",
        "Description": "A new parameter group"
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的管理 DAX 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateParameterGroup。

create-subnet-group

以下程式碼範例顯示如何使用 create-subnet-group。

AWS CLI

建立 DAX 子網路群組

下列create-subnet-group範例會建立具有指定設定的子網路群組。

```
aws dax create-subnet-group \
    --subnet-group-name daxSubnetGroup \
    --subnet-ids subnet-11111111 subnet-22222222
```

輸出：

```
{
    "SubnetGroup": {
        "SubnetGroupName": "daxSubnetGroup",
        "VpcId": "vpc-05a1fa8e00c325226",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-11111111",
                "SubnetAvailabilityZone": "us-west-2b"
            },
            {
                "SubnetIdentifier": "subnet-22222222",
                "SubnetAvailabilityZone": "us-west-2c"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的步驟 2：建立子網路群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSubnetGroup。

decrease-replication-factor

以下程式碼範例顯示如何使用 decrease-replication-factor。

AWS CLI

從叢集中移除一或多個節點

下列decrease-replication-factor範例會將指定 DAX 叢集中的節點數目減少為一個。

```
aws dax decrease-replication-factor \
    --cluster-name daxcluster \
    --new-replication-factor 1
```

輸出：

```
{
    "Cluster": {
        "ClusterName": "daxcluster",
        "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",
        "TotalNodes": 3,
        "ActiveNodes": 3,
        "NodeType": "dax.r4.large",
        "Status": "modifying",
        "ClusterDiscoveryEndpoint": {
            "Address": "daxcluster.ey3o9d.clustercfg.dax.usw2.cache.amazonaws.com",
            "Port": 8111
        },
        "Nodes": [
            {
                "NodeId": "daxcluster-a",
                "Endpoint": {
                    "Address": "daxcluster-
a.ey3o9d.0001.dax.usw2.cache.amazonaws.com",
                    "Port": 8111
                },
                "NodeCreateTime": 1576625059.509,
                "AvailabilityZone": "us-west-2c",
                "NodeStatus": "available",
                "ParameterGroupStatus": "in-sync"
            },
            {
                "NodeId": "daxcluster-b",
                "Endpoint": {
                    "Address": "daxcluster-
b.ey3o9d.0001.dax.usw2.cache.amazonaws.com",
                    "Port": 8111
                },
                "NodeCreateTime": 1576625059.509,
                "AvailabilityZone": "us-west-2a",
                "NodeStatus": "available",
                "ParameterGroupStatus": "in-sync"
            },
```

```
            {
                "NodeId": "daxcluster-c",
                "Endpoint": {
                    "Address": "daxcluster-
c.ey3o9d.0001.dax.usw2.cache.amazonaws.com",
                    "Port": 8111
                },
                "NodeCreateTime": 1576625059.509,
                "AvailabilityZone": "us-west-2b",
                "NodeStatus": "available",
                "ParameterGroupStatus": "in-sync"
            }
        ],
        "PreferredMaintenanceWindow": "thu:13:00-thu:14:00",
        "SubnetGroup": "default",
        "SecurityGroups": [
            {
                "SecurityGroupIdentifier": "sg-1af6e36e",
                "Status": "active"
            }
        ],
        "IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
        "ParameterGroup": {
            "ParameterGroupName": "default.dax1.0",
            "ParameterApplyStatus": "in-sync",
            "NodeIdsToReboot": []
        },
        "SSEDescription": {
            "Status": "ENABLED"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的管理 DAX 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DecreaseReplicationFactor。

delete-cluster

以下程式碼範例顯示如何使用 delete-cluster。

AWS CLI

刪除 DAX 叢集

下列delete-cluster範例會刪除指定的 DAX 叢集。

```
aws dax delete-cluster \
    --cluster-name daxcluster
```

輸出：

```
{
    "Cluster": {
        "ClusterName": "daxcluster",
        "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",
        "TotalNodes": 3,
        "ActiveNodes": 0,
        "NodeType": "dax.r4.large",
        "Status": "deleting",
        "ClusterDiscoveryEndpoint": {
            "Address": "dd.ey3o9d.clustercfg.dax.usw2.cache.amazonaws.com",
            "Port": 8111
        },
        "PreferredMaintenanceWindow": "fri:06:00-fri:07:00",
        "SubnetGroup": "default",
        "SecurityGroups": [
            {
                "SecurityGroupIdentifier": "sg-1af6e36e",
                "Status": "active"
            }
        ],
        "IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
        "ParameterGroup": {
            "ParameterGroupName": "default.dax1.0",
            "ParameterApplyStatus": "in-sync",
            "NodeIdsToReboot": []
        },
        "SSEDescription": {
            "Status": "ENABLED"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的管理 DAX 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCluster。

delete-parameter-group

以下程式碼範例顯示如何使用 delete-parameter-group。

AWS CLI

刪除參數群組

下列delete-parameter-group範例會刪除指定的 DAX 參數群組。

```
aws dax  delete-parameter-group \
    --parameter-group-name daxparametergroup
```

輸出：

```
{
    "DeletionMessage": "Parameter group daxparametergroup has been deleted."
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的管理 DAX 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteParameterGroup。

delete-subnet-group

以下程式碼範例顯示如何使用 delete-subnet-group。

AWS CLI

刪除子網路群組

下列delete-subnet-group範例會刪除指定的 DAX 子網路群組。

```
aws dax delete-subnet-group \
    --subnet-group-name daxSubnetGroup
```

輸出：

```
{
```

```
    "DeletionMessage": "Subnet group daxSubnetGroup has been deleted."
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的管理 DAX 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSubnetGroup。

describe-clusters

以下程式碼範例顯示如何使用 `describe-clusters`。

AWS CLI

傳回所有佈建 DAX 叢集的相關資訊

下列`describe-clusters`範例顯示所有佈建 DAX 叢集的詳細資訊。

```
aws dax describe-clusters
```

輸出：

```
{
    "Clusters": [
        {
            "ClusterName": "daxcluster",
            "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",
            "TotalNodes": 1,
            "ActiveNodes": 1,
            "NodeType": "dax.r4.large",
            "Status": "available",
            "ClusterDiscoveryEndpoint": {
                "Address":
 "daxcluster.ey3o9d.clustercfg.dax.usw2.cache.amazonaws.com",
                "Port": 8111
            },
            "Nodes": [
                {
                    "NodeId": "daxcluster-a",
                    "Endpoint": {
                        "Address": "daxcluster-
a.ey3o9d.0001.dax.usw2.cache.amazonaws.com",
                        "Port": 8111
                    },
```

```
                    "NodeCreateTime": 1576625059.509,
                    "AvailabilityZone": "us-west-2c",
                    "NodeStatus": "available",
                    "ParameterGroupStatus": "in-sync"
                }
            ],
            "PreferredMaintenanceWindow": "thu:13:00-thu:14:00",
            "SubnetGroup": "default",
            "SecurityGroups": [
                {
                    "SecurityGroupIdentifier": "sg-1af6e36e",
                    "Status": "active"
                }
            ],
            "IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
            "ParameterGroup": {
                "ParameterGroupName": "default.dax1.0",
                "ParameterApplyStatus": "in-sync",
                "NodeIdsToReboot": []
            },
            "SSEDescription": {
                "Status": "ENABLED"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的管理 DAX 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClusters。

describe-default-parameters

以下程式碼範例顯示如何使用 describe-default-parameters。

AWS CLI

傳回 DAX 的預設系統參數資訊

下列describe-default-parameters範例顯示 DAX 的預設系統參數資訊。

```
aws dax describe-default-parameters
```

輸出：

```
{
    "Parameters": [
        {
            "ParameterName": "query-ttl-millis",
            "ParameterType": "DEFAULT",
            "ParameterValue": "300000",
            "NodeTypeSpecificValues": [],
            "Description": "Duration in milliseconds for queries to remain cached",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": "TRUE",
            "ChangeType": "IMMEDIATE"
        },
        {
            "ParameterName": "record-ttl-millis",
            "ParameterType": "DEFAULT",
            "ParameterValue": "300000",
            "NodeTypeSpecificValues": [],
            "Description": "Duration in milliseconds for records to remain valid in
 cache (Default: 0 = infinite)",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": "TRUE",
            "ChangeType": "IMMEDIATE"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的管理 DAX 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDefaultParameters。

describe-events

以下程式碼範例顯示如何使用 describe-events。

AWS CLI

傳回與 DAX 叢集和參數群組相關的所有事件

下列describe-events範例顯示與 DAX 叢集和參數群組相關的事件詳細資訊。

```
aws dax describe-events
```

輸出：

```
{
    "Events": [
        {
            "SourceName": "daxcluster",
            "SourceType": "CLUSTER",
            "Message": "Cluster deleted.",
            "Date": 1576702736.706
        },
        {
            "SourceName": "daxcluster",
            "SourceType": "CLUSTER",
            "Message": "Removed node daxcluster-b.",
            "Date": 1576702691.738
        },
        {
            "SourceName": "daxcluster",
            "SourceType": "CLUSTER",
            "Message": "Removed node daxcluster-a.",
            "Date": 1576702633.498
        },
        {
            "SourceName": "daxcluster",
            "SourceType": "CLUSTER",
            "Message": "Removed node daxcluster-c.",
            "Date": 1576702631.329
        },
        {
            "SourceName": "daxcluster",
            "SourceType": "CLUSTER",
            "Message": "Cluster created.",
            "Date": 1576626560.057
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的管理 DAX 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEvents。

describe-parameter-groups

以下程式碼範例顯示如何使用 describe-parameter-groups。

AWS CLI

描述 DAX 中定義的參數群組

下列describe-parameter-groups範例會擷取 DAX 中定義之參數群組的詳細資訊。

```
aws dax describe-parameter-groups
```

輸出：

```
{
    "ParameterGroups": [
        {
            "ParameterGroupName": "default.dax1.0",
            "Description": "Default parameter group for dax1.0"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的管理 DAX 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeParameterGroups。

describe-parameters

以下程式碼範例顯示如何使用 describe-parameters。

AWS CLI

描述 DAX 參數群組中定義的參數

下列describe-parameters範例會擷取指定 DAX 參數群組中定義之參數的詳細資訊。

```
aws dax describe-parameters \
    --parameter-group-name default.dax1.0
```

輸出：

```
{
    "Parameters": [
        {
            "ParameterName": "query-ttl-millis",
            "ParameterType": "DEFAULT",
            "ParameterValue": "300000",
            "NodeTypeSpecificValues": [],
            "Description": "Duration in milliseconds for queries to remain cached",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": "TRUE",
            "ChangeType": "IMMEDIATE"
        },
        {
            "ParameterName": "record-ttl-millis",
            "ParameterType": "DEFAULT",
            "ParameterValue": "300000",
            "NodeTypeSpecificValues": [],
            "Description": "Duration in milliseconds for records to remain valid in
 cache (Default: 0 = infinite)",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": "TRUE",
            "ChangeType": "IMMEDIATE"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的管理 DAX 叢集。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeParameter。

describe-subnet-groups

以下程式碼範例顯示如何使用 describe-subnet-groups。

AWS CLI

描述 DAX 中定義的子網路群組

下列describe-subnet-groups範例會擷取 DAX 中定義之子網路群組的詳細資訊。

```
aws dax describe-subnet-groups
```

輸出：

```
{
    "SubnetGroups": [
        {
            "SubnetGroupName": "default",
            "Description": "Default CacheSubnetGroup",
            "VpcId": "vpc-ee70a196",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-874953af",
                    "SubnetAvailabilityZone": "us-west-2d"
                },
                {
                    "SubnetIdentifier": "subnet-bd3d1fc4",
                    "SubnetAvailabilityZone": "us-west-2a"
                },
                {
                    "SubnetIdentifier": "subnet-72c2ff28",
                    "SubnetAvailabilityZone": "us-west-2c"
                },
                {
                    "SubnetIdentifier": "subnet-09e6aa42",
                    "SubnetAvailabilityZone": "us-west-2b"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的管理 DAX 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSubnetGroups。

increase-replication-factor

以下程式碼範例顯示如何使用 increase-replication-factor。

AWS CLI

增加 DAX 叢集的複寫係數

下列increase-replication-factor範例會將指定的 DAX 叢集複寫係數增加為 3。

```
aws dax increase-replication-factor \
    --cluster-name daxcluster \
    --new-replication-factor 3
```

輸出：

```
{
    "Cluster": {
        "ClusterName": "daxcluster",
        "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",
        "TotalNodes": 3,
        "ActiveNodes": 1,
        "NodeType": "dax.r4.large",
        "Status": "modifying",
        "ClusterDiscoveryEndpoint": {
            "Address": "daxcluster.ey3o9d.clustercfg.dax.usw2.cache.amazonaws.com",
            "Port": 8111
        },
        "Nodes": [
            {
                "NodeId": "daxcluster-a",
                "Endpoint": {
                    "Address": "daxcluster-
a.ey3o9d.0001.dax.usw2.cache.amazonaws.com",
                    "Port": 8111
                },
                "NodeCreateTime": 1576625059.509,
                "AvailabilityZone": "us-west-2c",
                "NodeStatus": "available",
                "ParameterGroupStatus": "in-sync"
            },
            {
                "NodeId": "daxcluster-b",
                "NodeStatus": "creating"
            },
            {
                "NodeId": "daxcluster-c",
```

```
                "NodeStatus": "creating"
            }
        ],
        "PreferredMaintenanceWindow": "thu:13:00-thu:14:00",
        "SubnetGroup": "default",
        "SecurityGroups": [
            {
                "SecurityGroupIdentifier": "sg-1af6e36e",
                "Status": "active"
            }
        ],
        "IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
        "ParameterGroup": {
            "ParameterGroupName": "default.dax1.0",
            "ParameterApplyStatus": "in-sync",
            "NodeIdsToReboot": []
        },
        "SSEDescription": {
            "Status": "ENABLED"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的管理 DAX 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 IncreaseReplicationFactor。

`list-tags`

以下程式碼範例顯示如何使用 `list-tags`。

AWS CLI

列出 DAX 資源上的標籤

下列`list-tags`範例列出連接至指定 DAX 叢集的標籤索引鍵和值。

```
aws dax list-tags \
    --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster
```

輸出：

```
{
    "Tags": [
        {
            "Key": "ClusterUsage",
            "Value": "prod"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的管理 DAX 叢集。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTags。

`tag-resource`

以下程式碼範例顯示如何使用 `tag-resource`。

AWS CLI

標記 DAX 資源

下列`tag-resource`範例會將指定的標籤金鑰名稱和相關聯的值連接到指定的 DAX 叢集，以描述叢集用量。

```
aws dax tag-resource \
    --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster \
    --tags="Key=ClusterUsage,Value=prod"
```

輸出：

```
{
    "Tags": [
        {
            "Key": "ClusterUsage",
            "Value": "prod"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的管理 DAX 叢集。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

`untag-resource`

以下程式碼範例顯示如何使用 `untag-resource`。

AWS CLI

從 DAX 資源移除標籤

下列`untag-resource`範例會從 DAX 叢集移除具有指定金鑰名稱的標籤。

```
aws dax untag-resource  \
    --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster \
    --tag-keys="ClusterUsage"
```

輸出：

```
{
    "Tags": []
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的管理 DAX 叢集。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

使用 的偵測範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Detective 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`accept-invitation`

以下程式碼範例顯示如何使用 `accept-invitation`。

AWS CLI

接受在行為圖表中成為成員帳戶的邀請

下列`accept-invitation`範例接受邀請，以在行為圖表 arn：aws：detective：us-east-1：111122223333：graph：123412341234 中成為成員帳戶。

```
aws detective accept-invitation \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Detective 管理指南》中的回應行為圖表邀請。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptInvitation。

`create-graph`

以下程式碼範例顯示如何使用 `create-graph`。

AWS CLI

啟用 Amazon Detective 並建立新的行為圖表

下列`create-graph`範例為在執行命令的 區域中執行命令 AWS 的帳戶啟用 Detective。建立新的行為圖表，以該帳戶做為其管理員帳戶。命令也會將 Finance 值指派給 Department 標籤。

```
aws detective create-graph \
    --tags '{"Department": "Finance"}'
```

輸出：

```
{
    "GraphArn": "arn:aws:detective:us-
east-1:111122223333:graph:027c7c4610ea4aacaf0b883093cab899"
```

```
}
```

如需詳細資訊，請參閱《Amazon Detective 管理指南》中的啟用 Amazon Detective。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateGraph。

create-members

以下程式碼範例顯示如何使用 create-members。

AWS CLI

邀請成員帳戶加入行為圖表

下列create-members範例會邀請兩個 AWS 帳戶成為行為圖表 arn：aws：detective：us-east-1：111122223333：graph：123412341234 中的成員帳戶。對於每個帳戶，請求會提供 AWS 帳戶 ID 和帳戶根使用者電子郵件地址。請求包含要插入邀請電子郵件的自訂訊息。

```
aws detective create-members \
    --
accounts AccountId=444455556666,EmailAddress=mmajor@example.com AccountId=123456789012,Email
 \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \
    --message "This is Paul Santos. I need to add your account to the data we use
 for security investigation in Amazon Detective. If you have any questions, contact
 me at psantos@example.com."
```

輸出：

```
{
    "Members": [
    {
        "AccountId": "444455556666",
        "AdministratorId": "111122223333",
        "EmailAddress": "mmajor@example.com",
        "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
        "InvitedTime": 1579826107000,
        "MasterId": "111122223333",
        "Status": "INVITED",
        "UpdatedTime": 1579826107000
   },
   {
```

```
        "AccountId": "123456789012",
        "AdministratorId": "111122223333",
        "EmailAddress": "jstiles@example.com",
        "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
        "InvitedTime": 1579826107000,
        "MasterId": "111122223333",
        "Status": "VERIFICATION_IN_PROGRESS",
        "UpdatedTime": 1579826107000
     }
    ],
    "UnprocessedAccounts": [ ]
}
```

如需詳細資訊，請參閱《Amazon Detective 管理指南》中的將成員帳戶邀請至行為圖表<https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-add-member-accounts.html>。

在不傳送邀請電子郵件的情況下邀請成員帳戶

下列create-members範例會邀請兩個 AWS 帳戶成為行為圖表 arn：aws：detective：us-east-1：111122223333：graph：123412341234 中的成員帳戶。對於每個帳戶，請求會提供 AWS 帳戶 ID 和帳戶根使用者電子郵件地址。成員帳戶不會收到邀請電子郵件。

```
aws detective create-members \
    --
accounts AccountId=444455556666,EmailAddress=mmajor@example.com AccountId=123456789012,Email
 \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \
    --disable-email-notification
```

輸出：

```
{
    "Members": [
    {
        "AccountId": "444455556666",
        "AdministratorId": "111122223333",
        "EmailAddress": "mmajor@example.com",
        "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
        "InvitedTime": 1579826107000,
        "MasterId": "111122223333",
        "Status": "INVITED",
        "UpdatedTime": 1579826107000
```

```
    },
    {
         "AccountId": "123456789012",
         "AdministratorId": "111122223333",
         "EmailAddress": "jstiles@example.com",
         "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
         "InvitedTime": 1579826107000,
         "MasterId": "111122223333",
         "Status": "VERIFICATION_IN_PROGRESS",
         "UpdatedTime": 1579826107000
      }
     ],
     "UnprocessedAccounts": [ ]
}
```

如需詳細資訊，請參閱《Amazon Detective 管理指南》中的將成員帳戶邀請至行為圖表<https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-add-member-accounts.html>。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateMembers。

delete-graph

以下程式碼範例顯示如何使用 delete-graph。

AWS CLI

停用 Detective 並刪除行為圖表

下列delete-graph範例會停用 Detective 並刪除指定的行為圖表。

```
aws detective delete-graph \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Detective 管理指南》中的停用 Amazon Detective。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteGraph。

delete-members

以下程式碼範例顯示如何使用 delete-members。

AWS CLI

從行為圖表中移除成員帳戶

下列delete-members範例會從行為圖表 arn：aws：detective：us-east-1：111122223333：graph：123412341234 中移除兩個成員帳戶。若要識別帳戶，請求會提供 AWS 帳戶 IDs。

```
aws detective delete-members \
    --account-ids 444455556666 123456789012 \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

輸出：

```
 {
    "AccountIds": [ "444455556666", "123456789012" ],
    "UnprocessedAccounts": [ ]
}
```

如需詳細資訊，請參閱《Amazon Detective 管理指南》中的從行為圖表移除成員帳戶<https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-remove-member-accounts.html>。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteMembers。

disassociate-membership

以下程式碼範例顯示如何使用 disassociate-membership。

AWS CLI

從行為圖表中重新簽署成員資格

下列 disassociate-membership 範例會從行為圖表 arn：aws：detective：us-east-1：111122223333：graph：123412341234 中移除執行命令 AWS 的帳戶。

```
aws detective disassociate-membership \
     --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

如需詳細資訊，請參閱《Amazon Detective 管理指南》中的從行為圖表中移除您的帳戶<https://docs.aws.amazon.com/detective/latest/adminguide/member-remove-self-from-graph.html>。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateMembership。

get-members

以下程式碼範例顯示如何使用 get-members。

AWS CLI

擷取所選行為圖表成員帳戶的相關資訊

下列get-members範例會擷取行為圖表 arn：aws：detective：us-east-1：111122223333：graph：123412341234 中兩個成員帳戶的相關資訊。對於這兩個帳戶，請求會提供 AWS 帳戶 IDs。

```
aws detective get-members \
    --account-ids 444455556666 123456789012 \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

輸出：

```
{
    "MemberDetails": [
    {
        "AccountId": "444455556666",
        "AdministratorId": "111122223333",
        "EmailAddress": "mmajor@example.com",
        "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
        "InvitedTime": 1579826107000,
        "MasterId": "111122223333",
        "Status": "INVITED",
        "UpdatedTime": 1579826107000
    }
    {
        "AccountId": "123456789012",
        "AdministratorId": "111122223333",
        "EmailAddress": "jstiles@example.com",
        "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
        "InvitedTime": 1579826107000,
        "MasterId": "111122223333",
        "Status": "INVITED",
        "UpdatedTime": 1579826107000
    }
],
    "UnprocessedAccounts": [ ]
```

```
}
```

如需詳細資訊，請參閱《Amazon Detective 管理指南》中的在行為圖表中檢視帳戶清單<https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-view-accounts.html>。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMembers。

`list-graphs`

以下程式碼範例顯示如何使用 `list-graphs`。

AWS CLI

檢視您的帳戶是 管理員的行為圖表清單

下列`list-graphs`範例會擷取呼叫帳戶是目前區域中 管理員的行為圖表。

```
aws detective list-graphs
```

輸出：

```
{
    "GraphList": [
        {
            "Arn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
            "CreatedTime": 1579736111000
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListGraphs。

`list-invitations`

以下程式碼範例顯示如何使用 `list-invitations`。

AWS CLI

檢視帳戶是 成員或受邀加入的行為圖表清單

下列`list-invitations`範例會擷取呼叫帳戶受邀使用的行為圖表。結果僅包含開啟和接受的邀請。它們不包含拒絕的邀請或移除的成員資格。

```
aws detective list-invitations
```

輸出：

```
{
    "Invitations": [
    {
        "AccountId": "444455556666",
        "AdministratorId": "111122223333",
        "EmailAddress": "mmajor@example.com",
        "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
        "InvitedTime": 1579826107000,
        "MasterId": "111122223333",
        "Status": "INVITED",
        "UpdatedTime": 1579826107000
    }
]
}
```

如需詳細資訊，請參閱《Amazon Detective 管理指南》中的檢視行為圖表邀請清單<https://docs.aws.amazon.com/detective/latest/adminguide/member-view-graph-invitations.html>。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListInvitations。

`list-members`

以下程式碼範例顯示如何使用 `list-members`。

AWS CLI

在行為圖表中列出成員帳戶

下列`list-members`範例會擷取行為圖表 的受邀和啟用成員帳戶`arn:aws:detective:us-east-1:111122223333:graph:123412341234`。結果不包含已移除的成員帳戶。

```
aws detective list-members \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

輸出：

```
{
```

```
    "MemberDetails": [
        {
            "AccountId": "444455556666",
            "AdministratorId": "111122223333",
            "EmailAddress": "mmajor@example.com",
            "GraphArn": "arn:aws:detective:us-
east-1:111122223333:graph:123412341234",
            "InvitedTime": 1579826107000,
            "MasterId": "111122223333",
            "Status": "INVITED",
            "UpdatedTime": 1579826107000
        },
        {
            "AccountId": "123456789012",
            "AdministratorId": "111122223333",
            "EmailAddress": "jstiles@example.com",
            "GraphArn": "arn:aws:detective:us-
east-1:111122223333:graph:123412341234",
            "InvitedTime": 1579826107000,
            "MasterId": "111122223333",
            "PercentOfGraphUtilization": 2,
            "PercentOfGraphUtilizationUpdatedTime": 1586287843,
            "Status": "ENABLED",
            "UpdatedTime": 1579973711000,
            "VolumeUsageInBytes": 200,
            "VolumeUsageUpdatedTime": 1586287843
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Detective 管理指南》中的在行為圖表中檢視帳戶清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMembers。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

擷取指派給行為圖表的標籤

下列`list-tags-for-resource`範例會傳回指派給指定行為圖表的標籤。

```
aws detective list-tags-for-resource \
    --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

輸出：

```
{
    "Tags": {
        "Department" : "Finance"
    }
}
```

如需詳細資訊，請參閱《Amazon Detective 管理指南》中的管理行為圖表的標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

reject-invitation

以下程式碼範例顯示如何使用 reject-invitation。

AWS CLI

拒絕在行為圖表中成為成員帳戶的邀請

下列reject-invitation範例拒絕在行為圖表 arn：aws：detective：us-east-1：111122223333：graph：123412341234 中成為成員帳戶的邀請。

```
aws detective reject-invitation \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Detective 管理指南》中的回應行為圖表邀請<https://docs.aws.amazon.com/detective/latest/adminguide/member-invitation-response.html>。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RejectInvitation。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤指派給資源

下列tag-resource範例會將 Department 標籤的值指派給指定的行為圖表。

```
aws detective tag-resource \
    --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \
    --tags '{"Department":"Finance"}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Detective 管理指南》中的管理行為圖表的標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源移除標籤值

下列untag-resource範例會從指定的行為圖表中移除部門標籤。

```
aws detective untag-resource \
    --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \
    --tag-keys "Department"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Detective 管理指南》中的管理行為圖表的標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

使用 的 Device Farm 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Device Farm 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-device-pool`

以下程式碼範例顯示如何使用 `create-device-pool`。

AWS CLI

建立裝置集區

下列命令會為專案建立 Android 裝置集區：

```
aws devicefarm create-device-pool --name pool1 --rules file://
device-pool-rules.json --project-arn "arn:aws:devicefarm:us-
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506"
```

您可以從 `create-project`或 的輸出取得專案 ARN`list-projects`。檔案`device-pool-rules.json`是目前資料夾中的 JSON 文件，可指定裝置平台：

```
[
    {
        "attribute": "PLATFORM",
        "operator": "EQUALS",
        "value": "\"ANDROID\""
    }
]
```

輸出：

```
{
    "devicePool": {
        "rules": [
            {
                "operator": "EQUALS",
                "attribute": "PLATFORM",
```

```
                "value": "\"ANDROID\""
            }
        ],
        "type": "PRIVATE",
        "name": "pool1",
        "arn": "arn:aws:devicefarm:us-
west-2:123456789012:devicepool:070fc3ca-7ec1-4741-9c1f-
d3e044efc506/2aa8d2a9-5e73-47ca-b929-659cb34b7dcd"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDevicePool。

create-project

以下程式碼範例顯示如何使用 create-project。

AWS CLI

建立專案

下列命令會建立新的專案，名為 my-project：

```
aws devicefarm create-project --name my-project
```

輸出：

```
{
    "project": {
        "name": "myproject",
        "arn": "arn:aws:devicefarm:us-
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506",
        "created": 1503612890.057
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateProject。

create-upload

以下程式碼範例顯示如何使用 create-upload。

AWS CLI

建立上傳

下列命令會為 Android 應用程式建立上傳：

```
aws devicefarm create-upload --project-arn "arn:aws:devicefarm:us-
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506" --name app.apk --
type ANDROID_APP
```

您可以從 create-project 或 list-projects 的輸出取得專案 ARN。

輸出：

```
{
    "upload": {
        "status": "INITIALIZED",
        "name": "app.apk",
        "created": 1503614408.769,
        "url": "https://prod-us-west-2-uploads.s3-us-west-2.amazonaws.com/
arn%3Aaws%3Adevicefarm%3Aus-west-2%3A123456789012%3Aproject%3A070fc3ca-
c7e1-4471-91cf-d3e4efc50604/uploads/arn%3Aaws%3Adevicefarm%3Aus-
west-2%3A123456789012%3Aupload%3A070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514/app.apk?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20170824T224008Z&X-Amz-SignedHeaders=host&X-Amz-Expires=86400&X-Amz-
Credential=AKIAEXAMPLEPBUMBC3GA%2F20170824%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-
Signature=05050370c38894ef5bd09f5d009f36fc8f96fa4bb04e1bba9aca71b8dbe49a0f",
        "type": "ANDROID_APP",
        "arn": "arn:aws:devicefarm:us-
west-2:123456789012:upload:070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514"
    }
}
```

在輸出中使用簽章的 URL 將檔案上傳至 Device Farm：

```
curl -T app.apk "https://prod-us-west-2-uploads.s3-us-west-2.amazonaws.com/
arn%3Aaws%3Adevicefarm%3Aus-west-2%3A123456789012%3Aproject%3A070fc3ca-
c7e1-4471-91cf-d3e4efc50604/uploads/arn%3Aaws%3Adevicefarm%3Aus-
west-2%3A123456789012%3Aupload%3A070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514/app.apk?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20170824T224008Z&X-Amz-SignedHeaders=host&X-Amz-Expires=86400&X-Amz-
```

```
Credential=AKIAEXAMPLEPBUMBC3GA%2F20170824%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-
Signature=05050370c38894ef5bd09f5d009f36fc8f96fa4bb04e1bba9aca71b8dbe49a0f"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateUpload。

get-upload

以下程式碼範例顯示如何使用 get-upload。

AWS CLI

檢視上傳

下列命令會擷取上傳的相關資訊：

```
aws devicefarm get-upload --arn "arn:aws:devicefarm:us-
west-2:123456789012:upload:070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514"
```

您可以從 的輸出取得上傳 ARNcreate-upload。

輸出：

```
{
    "upload": {
        "status": "SUCCEEDED",
        "name": "app.apk",
        "created": 1505262773.186,
        "type": "ANDROID_APP",
        "arn": "arn:aws:devicefarm:us-
west-2:123456789012:upload:070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514",
        "metadata": "{\"device_admin\":false,\"activity_name\":
\"ccom.example.client.LauncherActivity\",\"version_name\":\"1.0.2.94\",\"screens
\":[\"small\",\"normal\",\"large\",\"xlarge\"],\"error_type\":null,\"sdk_version
\":\"16\",\"package_name\":\"com.example.client\",\"version_code\":\"20994\",
\"native_code\":[\"armeabi-v7a\"],\"target_sdk_version\":\"25\"}"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetUpload。

`list-projects`

以下程式碼範例顯示如何使用 `list-projects`。

AWS CLI

列出專案

以下內容會擷取專案清單：

```
aws devicefarm list-projects
```

輸出：

```
{
    "projects": [
        {
            "name": "myproject",
            "arn": "arn:aws:devicefarm:us-
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506",
            "created": 1503612890.057
        },
        {
            "name": "otherproject",
            "arn": "arn:aws:devicefarm:us-
west-2:123456789012:project:a5f5b752-8098-49d1-86bf-5f7682c1c77e",
            "created": 1505257519.337
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListProjects。

AWS Direct Connect 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS Direct Connect。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

accept-direct-connect-gateway-association-proposal

以下程式碼範例顯示如何使用 accept-direct-connect-gateway-association-proposal。

AWS CLI

接受閘道關聯提案

以下accept-direct-connect-gateway-association-proposal接受指定的提案。

```
aws directconnect  accept-direct-connect-gateway-association-proposal \
    --direct-connect-gateway-id 11460968-4ac1-4fd3-bdb2-00599EXAMPLE \
    --proposal-id cb7f41cb-8128-43a5-93b1-dcaedEXAMPLE \
    --associated-gateway-owner-account 111122223333

{
    "directConnectGatewayAssociation": {
        "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
        "directConnectGatewayOwnerAccount": "111122223333",
        "associationState": "associating",
        "associatedGateway": {
            "id": "tgw-02f776b1a7EXAMPLE",
            "type": "transitGateway",
            "ownerAccount": "111122223333",
            "region": "us-east-1"
        },
        "associationId": "6441f8bf-5917-4279-ade1-9708bEXAMPLE",
        "allowedPrefixesToDirectConnectGateway": [
            {
                "cidr": "192.168.1.0/30"
            }
        ]

  }
}
```

如需詳細資訊，請參閱 AWS Direct Connect 使用者指南中的接受或拒絕傳輸閘道關聯提案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptDirectConnectGatewayAssociationProposal。

allocate-connection-on-interconnect

以下程式碼範例顯示如何使用 allocate-connection-on-interconnect。

AWS CLI

在 互連上建立託管連線

下列allocate-connection-on-interconnect命令會在 互連上建立託管連線：

```
aws directconnect allocate-connection-on-interconnect --bandwidth 500Mbps --
connection-name mydcinterconnect --owner-account 123456789012 --interconnect-
id dxcon-fgktov66 --vlan 101
```

輸出：

```
{
    "partnerName": "TIVIT",
    "vlan": 101,
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffzc51m1",
    "connectionState": "ordering",
    "bandwidth": "500Mbps",
    "location": "TIVIT",
    "connectionName": "mydcinterconnect",
    "region": "sa-east-1"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AllocateConnectionOnInterconnect。

allocate-hosted-connection

以下程式碼範例顯示如何使用 allocate-hosted-connection。

AWS CLI

在互連上建立託管連線

下列allocate-hosted-connection範例會在指定的互連上建立託管連線。

```
aws directconnect allocate-hosted-connection \
    --bandwidth 500Mbps \
    --connection-name mydcinterconnect \
    --owner-account 123456789012
    -connection-id dxcon-fgktov66
    -vlan 101
```

輸出：

```
{
    "partnerName": "TIVIT",
    "vlan": 101,
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffzc51m1",
    "connectionState": "ordering",
    "bandwidth": "500Mbps",
    "location": "TIVIT",
    "connectionName": "mydcinterconnect",
    "region": "sa-east-1"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AllocateHostedConnection。

allocate-private-virtual-interface

以下程式碼範例顯示如何使用 allocate-private-virtual-interface。

AWS CLI

佈建私有虛擬介面

下列allocate-private-virtual-interface命令會佈建由不同客戶擁有的私有虛擬介面：

```
aws directconnect allocate-private-virtual-interface --connection-id dxcon-
ffjrkx17 --owner-account 123456789012 --new-private-virtual-interface-
allocation virtualInterfaceName=PrivateVirtualInterface,vlan=1000,asn=65000,authKey=asdf34ex
```

輸出：

```
{
    "virtualInterfaceState": "confirming",
```

```
    "asn": 65000,
    "vlan": 1000,
    "customerAddress": "192.168.1.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffjrkx17",
    "virtualInterfaceId": "dxvif-fgy8orxu",
    "authKey": "asdf34example",
    "routeFilterPrefixes": [],
    "location": "TIVIT",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>\n  <logical_connection id=\"dxvif-fgy8orxu\">\n  <vlan>1000</
vlan>\n  <customer_address>192.168.1.2/30</customer_address>\n
 <amazon_address>192.168.1.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
 <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\n  <connection_type>private</connection_type>\n</logical_connection>\n",
    "amazonAddress": "192.168.1.1/30",
    "virtualInterfaceType": "private",
    "virtualInterfaceName": "PrivateVirtualInterface"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AllocatePrivateVirtualInterface。

allocate-public-virtual-interface

以下程式碼範例顯示如何使用 allocate-public-virtual-interface。

AWS CLI

佈建公有虛擬介面

下列allocate-public-virtual-interface命令會佈建由不同客戶擁有的公有虛擬介面：

```
aws directconnect allocate-public-virtual-interface --connection-id dxcon-
ffjrkx17 --owner-account 123456789012 --new-public-virtual-interface-
allocation virtualInterfaceName=PublicVirtualInterface,vlan=2000,asn=65000,authKey=asdf34exa
{cidr=203.0.113.4/30}]
```

輸出：

```
{
    "virtualInterfaceState": "confirming",
    "asn": 65000,
    "vlan": 2000,
```

```
    "customerAddress": "203.0.113.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffjrkx17",
    "virtualInterfaceId": "dxvif-fg9xo9vp",
    "authKey": "asdf34example",
    "routeFilterPrefixes": [
        {
            "cidr": "203.0.113.0/30"
        },
        {
            "cidr": "203.0.113.4/30"
        }
    ],
    "location": "TIVIT",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>\n<logical_connection id=\"dxvif-fg9xo9vp\">\n  <vlan>2000</
vlan>\n  <customer_address>203.0.113.2/30</customer_address>\n
 <amazon_address>203.0.113.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
 <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\n  <connection_type>public</connection_type>\n</logical_connection>\n",
    "amazonAddress": "203.0.113.1/30",
    "virtualInterfaceType": "public",
    "virtualInterfaceName": "PublicVirtualInterface"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AllocatePublicVirtualInterface。

`allocate-transit-virtual-interface`

以下程式碼範例顯示如何使用 `allocate-transit-virtual-interface`。

AWS CLI

佈建要由指定 AWS 帳戶擁有的傳輸虛擬介面

下列`allocate-transit-virtual-interface`範例會為指定的帳戶佈建傳輸虛擬介面。

```
aws directconnect allocate-transit-virtual-interface \
    --connection-id dxlag-fEXAMPLE \
    --owner-account 123456789012 \
    --new-transit-virtual-interface-allocation "virtualInterfaceName=Example Transit
 Virtual
 Interface,vlan=126,asn=65110,mtu=1500,authKey=0xzxgA9YoW9h58u8SEXAMPLE,amazonAddress=192.16
```

輸出：


```
{
    "virtualInterface": {
        "ownerAccount": "123456789012",
        "virtualInterfaceId": "dxvif-fEXAMPLE",
        "location": "loc1",
        "connectionId": "dxlag-fEXAMPLE",
        "virtualInterfaceType": "transit",
        "virtualInterfaceName": "Example Transit Virtual Interface",
        "vlan": 126,
        "asn": 65110,
        "amazonSideAsn": 7224,
        "authKey": "0xzxgA9YoW9h58u8SEXAMPLE",
        "amazonAddress": "192.168.1.1/30",
        "customerAddress": "192.168.1.2/30",
        "addressFamily": "ipv4",
        "virtualInterfaceState": "confirming",
        "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-fEXAMPLE\">\n  <vlan>126</
vlan>\n  <customer_address>192.168.1.2/30</customer_address>\n
 <amazon_address>192.168.1.1/30</amazon_address>\n  <bgp_asn>65110</bgp_asn>\n
 <bgp_auth_key>0xzxgA9YoW9h58u8SEXAMPLE</bgp_auth_key>\n  <amazon_bgp_asn>7224</
amazon_bgp_asn>\n  <connection_type>transit</connection_type>\n</logical_connection>
\n",
        "mtu": 1500,
        "jumboFrameCapable": true,
        "virtualGatewayId": "",
        "directConnectGatewayId": "",
        "routeFilterPrefixes": [],
        "bgpPeers": [
            {
                "bgpPeerId": "dxpeer-fEXAMPLE",
                "asn": 65110,
                "authKey": "0xzxgA9YoW9h58u8EXAMPLE",
                "addressFamily": "ipv4",
                "amazonAddress": "192.168.1.1/30",
                "customerAddress": "192.168.1.2/30",
                "bgpPeerState": "pending",
                "bgpStatus": "down",
                "awsDeviceV2": "loc1-26wz6vEXAMPLE"
            }
        ],
        "region": "sa-east-1",
```

```
        "awsDeviceV2": "loc1-26wz6vEXAMPLE",
        "tags": [
            {
                "key": "Tag",
                "value": "Example"
            }
        ]
    }
}
```

如需詳細資訊，請參閱 AWS Direct Connect 使用者指南中的建立託管傳輸虛擬介面。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AllocateTransitVirtualInterface。

associate-connection-with-lag

以下程式碼範例顯示如何使用 associate-connection-with-lag。

AWS CLI

將連線與 LAG 建立關聯

下列範例會將指定的連線與指定的 LAG 建立關聯。

命令：

```
aws directconnect associate-connection-with-lag --lag-id dxlag-fhccu14t --
connection-id dxcon-fg9607vm
```

輸出：

```
{
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg9607vm",
  "lagId": "dxlag-fhccu14t",
  "connectionState": "requested",
  "bandwidth": "1Gbps",
  "location": "EqDC2",
  "connectionName": "Con2ForLag",
  "region": "us-east-1"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateConnectionWithLag。

associate-hosted-connection

以下程式碼範例顯示如何使用 associate-hosted-connection。

AWS CLI

將託管連線與 LAG 建立關聯

下列範例會將指定的託管連線與指定的 LAG 建立關聯。

命令：

```
aws directconnect associate-hosted-connection --parent-connection-id dxlag-fhccu14t
 --connection-id dxcon-fg9607vm
```

輸出：

```
{
  "partnerName": "TIVIT",
  "vlan": 101,
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg9607vm",
  "lagId": "dxlag-fhccu14t",
  "connectionState": "ordering",
  "bandwidth": "500Mbps",
  "location": "TIVIT",
  "connectionName": "mydcinterconnect",
  "region": "sa-east-1"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateHostedConnection。

associate-virtual-interface

以下程式碼範例顯示如何使用 associate-virtual-interface。

AWS CLI

將虛擬介面與連線建立關聯

下列範例會將指定的虛擬介面與指定的 LAG 建立關聯。或者，若要將虛擬介面與連線建立關聯，請指定 的 AWS Direct Connect 連線 ID--connection-id；例如 dxcon-ffnikghc。

命令：

```
aws directconnect associate-virtual-interface --connection-id dxlag-ffjhj9lx --
virtual-interface-id dxvif-fgputw0j
```

輸出：

```
{
  "virtualInterfaceState": "pending",
  "asn": 65000,
  "vlan": 123,
  "customerAddress": "169.254.255.2/30",
  "ownerAccount": "123456789012",
  "connectionId": "dxlag-ffjhj9lx",
  "addressFamily": "ipv4",
  "virtualGatewayId": "vgw-38e90b51",
  "virtualInterfaceId": "dxvif-fgputw0j",
  "authKey": "0x123pK5_VBqv.UQ3kJ4123_",
  "routeFilterPrefixes": [],
  "location": "CSVA1",
  "bgpPeers": [
      {
          "bgpStatus": "down",
          "customerAddress": "169.254.255.2/30",
          "addressFamily": "ipv4",
          "authKey": "0x123pK5_VBqv.UQ3kJ4123_",
          "bgpPeerState": "deleting",
          "amazonAddress": "169.254.255.1/30",
          "asn": 65000
      },
      {
          "bgpStatus": "down",
          "customerAddress": "169.254.255.2/30",
          "addressFamily": "ipv4",
          "authKey": "0x123pK5_VBqv.UQ3kJ4123_",
          "bgpPeerState": "pending",
          "amazonAddress": "169.254.255.1/30",
          "asn": 65000
      }
  ],
  "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>\n<logical_connection id=\"dxvif-fgputw0j\">\n  <vlan>123</vlan>
\n  <customer_address>169.254.255.2/30</customer_address>\n
```

```
 <amazon_address>169.254.255.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
  <bgp_auth_key>0x123pK5_VBqv.UQ3kJ4123_</bgp_auth_key>\n  <amazon_bgp_asn>7224</
amazon_bgp_asn>\n  <connection_type>private</connection_type>\n</logical_connection>
\n",
  "amazonAddress": "169.254.255.1/30",
  "virtualInterfaceType": "private",
  "virtualInterfaceName": "VIF1A"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateVirtualInterface。

confirm-connection

以下程式碼範例顯示如何使用 confirm-connection。

AWS CLI

確認在互連上建立託管連線

下列confirm-connection命令會確認在 互連上建立託管連線：

```
aws directconnect confirm-connection --connection-id dxcon-fg2wi7hy
```

輸出：

```
{
    "connectionState": "pending"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ConfirmConnection。

confirm-private-virtual-interface

以下程式碼範例顯示如何使用 confirm-private-virtual-interface。

AWS CLI

接受私有虛擬介面的擁有權

下列confirm-private-virtual-interface命令接受另一個客戶所建立私有虛擬介面的擁有權：

```
aws directconnect confirm-private-virtual-interface --virtual-interface-id dxvif-fgy8orxu --virtual-gateway-id vgw-e4a47df9
```

輸出：

```
{
    "virtualInterfaceState": "pending"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ConfirmPrivateVirtualInterface。

confirm-public-virtual-interface

以下程式碼範例顯示如何使用 confirm-public-virtual-interface。

AWS CLI

接受公有虛擬界面的擁有權

下列confirm-public-virtual-interface命令接受另一個客戶建立的公有虛擬介面的擁有權：

```
aws directconnect confirm-public-virtual-interface --virtual-interface-id dxvif-fg9xo9vp
```

輸出：

```
{
    "virtualInterfaceState": "verifying"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ConfirmPublicVirtualInterface。

confirm-transit-virtual-interface

以下程式碼範例顯示如何使用 confirm-transit-virtual-interface。

AWS CLI

接受傳輸虛擬界面的擁有權

以下confirm-transit-virtual-interface接受另一個客戶所建立傳輸虛擬介面的擁有權。

```
aws directconnect confirm-transit-virtual-interface \
    --virtual-interface-id dxvif-fEXAMPLE \
    --direct-connect-gateway-id 4112ccf9-25e9-4111-8237-b6c5dEXAMPLE
```

輸出：

```
{
    "virtualInterfaceState": "pending"
}
```

如需詳細資訊，請參閱 AWS Direct Connect 使用者指南中的接受託管虛擬介面。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ConfirmTransitVirtualInterface。

create-bgp-peer

以下程式碼範例顯示如何使用 create-bgp-peer。

AWS CLI

建立 IPv6 BGP 對等互連工作階段

下列範例會在私有虛擬介面 上建立 IPv6 BGP 對等互連工作階段dxvif-fg1vuj3d。Amazon 會自動配置對等 IPv6 地址。

命令：

```
aws directconnect create-bgp-peer --virtual-interface-id dxvif-fg1vuj3d --new-bgp-peer asn=64600,addressFamily=ipv6
```

輸出：

```
{
  "virtualInterface": {
      "virtualInterfaceState": "available",
      "asn": 65000,
      "vlan": 125,
      "customerAddress": "169.254.255.2/30",
      "ownerAccount": "123456789012",
```

```
      "connectionId": "dxcon-fguhmqlc",
      "addressFamily": "ipv4",
      "virtualGatewayId": "vgw-f9eb0c90",
      "virtualInterfaceId": "dxvif-fg1vuj3d",
      "authKey": "0xC_ukbCerl6EYA0example",
      "routeFilterPrefixes": [],
      "location": "EqDC2",
      "bgpPeers": [
          {
              "bgpStatus": "down",
              "customerAddress": "169.254.255.2/30",
              "addressFamily": "ipv4",
              "authKey": "0xC_ukbCerl6EYA0uexample",
              "bgpPeerState": "available",
              "amazonAddress": "169.254.255.1/30",
              "asn": 65000
          },
          {
              "bgpStatus": "down",
              "customerAddress": "2001:db8:1100:2f0:0:1:9cb4:4216/125",
              "addressFamily": "ipv6",
              "authKey": "0xS27kAIU_VHPjjAexample",
              "bgpPeerState": "pending",
              "amazonAddress": "2001:db8:1100:2f0:0:1:9cb4:4211/125",
              "asn": 64600
          }
      ],
      "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-fg1vuj3d\">\n  <vlan>125</
vlan>\n  <customer_address>169.254.255.2/30</customer_address>\n
 <amazon_address>169.254.255.1/30</amazon_address>\n  <bgp_asn>65000</
bgp_asn>\n  <bgp_auth_key>0xC_ukbCerl6EYA0uexample</bgp_auth_key>\n
 <ipv6_customer_address>2001:db8:1100:2f0:0:1:9cb4:4216/125</ipv6_customer_address>
\n  <ipv6_amazon_address>2001:db8:1100:2f0:0:1:9cb4:4211/125</ipv6_amazon_address>\n
  <ipv6_bgp_asn>64600</ipv6_bgp_asn>\n  <ipv6_bgp_auth_key>0xS27kAIU_VHPjjAexample</
ipv6_bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>\n
 <connection_type>private</connection_type>\n</logical_connection>\n",
      "amazonAddress": "169.254.255.1/30",
      "virtualInterfaceType": "private",
      "virtualInterfaceName": "Test"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateBgpPeer。

create-connection

以下程式碼範例顯示如何使用 create-connection。

AWS CLI

建立從網路到 AWS Direct Connect 位置的連線

下列create-connection命令會建立從網路到 AWS Direct Connect 位置的連線：

```
aws directconnect create-connection --location TIVIT --bandwidth 1Gbps --connection-
name "Connection to AWS"
```

輸出：

```
{
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-fg31dyv6",
    "connectionState": "requested",
    "bandwidth": "1Gbps",
    "location": "TIVIT",
    "connectionName": "Connection to AWS",
    "region": "sa-east-1"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateConnection。

create-direct-connect-gateway-association-proposal

以下程式碼範例顯示如何使用 create-direct-connect-gateway-association-proposal。

AWS CLI

建立提案，將指定的傳輸閘道與指定的 Direct Connect 閘道建立關聯

下列create-direct-connect-gateway-association-proposal範例會建立提案，將指定的傳輸閘道與指定的 Direct Connect 閘道建立關聯。

```
aws directconnect create-direct-connect-gateway-association-proposal \
    --direct-connect-gateway-id 11460968-4ac1-4fd3-bdb2-00599EXAMPLE \
    --direct-connect-gateway-owner-account 111122223333 \
```

```
    --gateway-id tgw-02f776b1a7EXAMPLE \
    --add-allowed-prefixes-to-direct-connect-gateway cidr=192.168.1.0/30
```

輸出：

```
{
    "directConnectGatewayAssociationProposal": {
        "proposalId": "cb7f41cb-8128-43a5-93b1-dcaedEXAMPLE",
        "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
        "directConnectGatewayOwnerAccount": "111122223333",
        "proposalState": "requested",
        "associatedGateway": {
            "id": "tgw-02f776b1a7EXAMPLE",
            "type": "transitGateway",
            "ownerAccount": "111122223333",
            "region": "us-east-1"
        },
        "requestedAllowedPrefixesToDirectConnectGateway": [
            {
                "cidr": "192.168.1.0/30"
            }
        ]
    }
}
```

如需詳細資訊，請參閱 AWS Direct Connect 使用者指南中的建立傳輸閘道關聯提案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDirectConnectGatewayAssociationProposal。

create-direct-connect-gateway-association

以下程式碼範例顯示如何使用 create-direct-connect-gateway-association。

AWS CLI

將虛擬私有閘道與 Direct Connect 閘道建立關聯

下列範例會將虛擬私有閘道vgw-6efe725e與 Direct Connect 閘道 建立關聯5f294f92-bafb-4011-916d-9b0bexample。您必須在虛擬私有閘道所在的區域中執行 命令。

命令：

```
aws directconnect create-direct-connect-gateway-association --direct-connect-
gateway-id 5f294f92-bafb-4011-916d-9b0bexample --virtual-gateway-id vgw-6efe725e
```

輸出：

```
{
  "directConnectGatewayAssociation": {
      "associationState": "associating",
      "virtualGatewayOwnerAccount": "123456789012",
      "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
      "virtualGatewayId": "vgw-6efe725e",
      "virtualGatewayRegion": "us-east-2"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDirectConnectGatewayAssociation。

create-direct-connect-gateway

以下程式碼範例顯示如何使用 create-direct-connect-gateway。

AWS CLI

建立 Direct Connect 閘道

下列範例會建立名稱為 的 Direct Connect 閘道DxGateway1。

命令：

```
aws directconnect create-direct-connect-gateway --direct-connect-gateway-
name "DxGateway1"
```

輸出：

```
{
  "directConnectGateway": {
      "amazonSideAsn": 64512,
      "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bdexample",
      "ownerAccount": "123456789012",
```

```
        "directConnectGatewayName": "DxGateway1",
        "directConnectGatewayState": "available"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDirectConnectGateway。

`create-interconnect`

以下程式碼範例顯示如何使用 `create-interconnect`。

AWS CLI

在合作夥伴的網路與 之間建立互連 AWS

下列`create-interconnect`命令會在 AWS Direct Connect 合作夥伴的網路與特定 AWS Direct Connect 位置之間建立互連：

```
aws directconnect create-interconnect --interconnect-name "1G Interconnect to AWS"
 --bandwidth 1Gbps --location TIVIT
```

輸出：

```
{
    "region": "sa-east-1",
    "bandwidth": "1Gbps",
    "location": "TIVIT",
    "interconnectName": "1G Interconnect to AWS",
    "interconnectId": "dxcon-fgktov66",
    "interconnectState": "requested"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateInterconnect。

`create-lag`

以下程式碼範例顯示如何使用 `create-lag`。

AWS CLI

使用新連線建立 LAG

下列範例會建立 LAG，並為頻寬為 1 Gbps 的 LAG 請求兩個新的 AWS Direct Connect 連線。

命令：

```
aws directconnect create-lag --location CSVA1 --number-of-connections 2 --
connections-bandwidth 1Gbps --lag-name 1GBLag
```

輸出：

```
{
  "awsDevice": "CSVA1-23u8tlpaz8iks",
  "numberOfConnections": 2,
  "lagState": "pending",
  "ownerAccount": "123456789012",
  "lagName": "1GBLag",
  "connections": [
      {
          "ownerAccount": "123456789012",
          "connectionId": "dxcon-ffqr6x5q",
          "lagId": "dxlag-ffjhj9lx",
          "connectionState": "requested",
          "bandwidth": "1Gbps",
          "location": "CSVA1",
          "connectionName": "Requested Connection 1 for Lag dxlag-ffjhj9lx",
          "region": "us-east-1"
      },
      {
          "ownerAccount": "123456789012",
          "connectionId": "dxcon-fflqyj95",
          "lagId": "dxlag-ffjhj9lx",
          "connectionState": "requested",
          "bandwidth": "1Gbps",
          "location": "CSVA1",
          "connectionName": "Requested Connection 2 for Lag dxlag-ffjhj9lx",
          "region": "us-east-1"
      }
  ],
  "lagId": "dxlag-ffjhj9lx",
  "minimumLinks": 0,
  "connectionsBandwidth": "1Gbps",
  "region": "us-east-1",
  "location": "CSVA1"
}
```

使用現有連線建立 LAG

下列範例會從您帳戶中的現有連線建立 LAG，並針對與現有連線具有相同頻寬和位置的 LAG 請求第二個新連線。

命令：

```
aws directconnect create-lag --location EqDC2 --number-of-connections 2 --
connections-bandwidth 1Gbps --lag-name 2ConnLAG --connection-id dxcon-fgk145dr
```

輸出：

```
{
  "awsDevice": "EqDC2-4h6ce2r1bes6",
  "numberOfConnections": 2,
  "lagState": "pending",
  "ownerAccount": "123456789012",
  "lagName": "2ConnLAG",
  "connections": [
      {
          "ownerAccount": "123456789012",
          "connectionId": "dxcon-fh6ljcvo",
          "lagId": "dxlag-fhccu14t",
          "connectionState": "requested",
          "bandwidth": "1Gbps",
          "location": "EqDC2",
          "connectionName": "Requested Connection 1 for Lag dxlag-fhccu14t",
          "region": "us-east-1"
      },
      {
          "ownerAccount": "123456789012",
          "connectionId": "dxcon-fgk145dr",
          "lagId": "dxlag-fhccu14t",
          "connectionState": "down",
          "bandwidth": "1Gbps",
          "location": "EqDC2",
          "connectionName": "VAConn1",
          "region": "us-east-1"
      }
  ],
  "lagId": "dxlag-fhccu14t",
  "minimumLinks": 0,
  "connectionsBandwidth": "1Gbps",
```

```
  "region": "us-east-1",
  "location": "EqDC2"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLag。

create-private-virtual-interface

以下程式碼範例顯示如何使用 create-private-virtual-interface。

AWS CLI

建立私有虛擬介面

下列create-private-virtual-interface命令會建立私有虛擬介面：

```
aws directconnect create-private-virtual-interface --connection-id dxcon-ffjrkx17 --
new-private-virtual-
interface virtualInterfaceName=PrivateVirtualInterface,vlan=101,asn=65000,authKey=asdf34exam
aba37db6
```

輸出：

```
{
    "virtualInterfaceState": "pending",
    "asn": 65000,
    "vlan": 101,
    "customerAddress": "192.168.1.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffjrkx17",
    "virtualGatewayId": "vgw-aba37db6",
    "virtualInterfaceId": "dxvif-ffhhk74f",
    "authKey": "asdf34example",
    "routeFilterPrefixes": [],
    "location": "TIVIT",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-ffhhk74f\">\n  <vlan>101</
vlan>\n  <customer_address>192.168.1.2/30</customer_address>\n
 <amazon_address>192.168.1.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
 <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\n  <connection_type>private</connection_type>\n</logical_connection>\n",
    "amazonAddress": "192.168.1.1/30",
```

```
    "virtualInterfaceType": "private",
    "virtualInterfaceName": "PrivateVirtualInterface"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePrivateVirtualInterface。

create-public-virtual-interface

以下程式碼範例顯示如何使用 create-public-virtual-interface。

AWS CLI

建立公有虛擬介面

下列create-public-virtual-interface命令會建立公有虛擬介面：

```
aws directconnect create-public-virtual-interface --connection-id dxcon-ffjrkx17 --
new-public-virtual-
interface virtualInterfaceName=PublicVirtualInterface,vlan=2000,asn=65000,authKey=asdf34exam
{cidr=203.0.113.4/30}]
```

輸出：

```
{
    "virtualInterfaceState": "verifying",
    "asn": 65000,
    "vlan": 2000,
    "customerAddress": "203.0.113.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffjrkx17",
    "virtualInterfaceId": "dxvif-fgh0hcrk",
    "authKey": "asdf34example",
    "routeFilterPrefixes": [
        {
            "cidr": "203.0.113.0/30"
        },
        {
            "cidr": "203.0.113.4/30"
        }
    ],
    "location": "TIVIT",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>\n<logical_connection id=\"dxvif-fgh0hcrk\">\n  <vlan>2000</
```

```
vlan>\n  <customer_address>203.0.113.2/30</customer_address>\n
 <amazon_address>203.0.113.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
 <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\n  <connection_type>public</connection_type>\n</logical_connection>\n",
    "amazonAddress": "203.0.113.1/30",
    "virtualInterfaceType": "public",
    "virtualInterfaceName": "PublicVirtualInterface"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePublicVirtualInterface。

create-transit-virtual-interface

以下程式碼範例顯示如何使用 create-transit-virtual-interface。

AWS CLI

建立傳輸虛擬界面

下列create-transit-virtual-interface範例會為指定的連線建立傳輸虛擬介面。

```
aws directconnect create-transit-virtual-interface \
    --connection-id dxlag-fEXAMPLE \
    --new-transit-virtual-interface "virtualInterfaceName=Example Transit Virtual
 Interface,vlan=126,asn=65110,mtu=1500,authKey=0xzxgA9YoW9h58u8SvEXAMPLE,amazonAddress=192.1
aada-5a1baEXAMPLE,tags=[{key=Tag,value=Example}]"
```

輸出：

```
{
    "virtualInterface": {
        "ownerAccount": "1111222233333",
        "virtualInterfaceId": "dxvif-fEXAMPLE",
        "location": "loc1",
        "connectionId": "dxlag-fEXAMPLE",
        "virtualInterfaceType": "transit",
        "virtualInterfaceName": "Example Transit Virtual Interface",
        "vlan": 126,
        "asn": 65110,
        "amazonSideAsn": 4200000000,
        "authKey": "0xzxgA9YoW9h58u8SEXAMPLE",
        "amazonAddress": "192.168.1.1/30",
```

```
        "customerAddress": "192.168.1.2/30",
        "addressFamily": "ipv4",
        "virtualInterfaceState": "pending",
        "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-fEXAMPLE\">\n  <vlan>126</
vlan>\n  <customer_address>192.168.1.2/30</customer_address>\n
 <amazon_address>192.168.1.1/30</amazon_address>\n  <bgp_asn>65110</
bgp_asn>\n  <bgp_auth_key>0xzxgA9YoW9h58u8SvOmXRTw</bgp_auth_key>\n
 <amazon_bgp_asn>4200000000</amazon_bgp_asn>\n  <connection_type>transit</
connection_type>\n</logical_connection>\n",
        "mtu": 1500,
        "jumboFrameCapable": true,
        "virtualGatewayId": "",
        "directConnectGatewayId": "8384da05-13ce-4a91-aada-5a1baEXAMPLE",
        "routeFilterPrefixes": [],
        "bgpPeers": [
            {
                "bgpPeerId": "dxpeer-EXAMPLE",
                "asn": 65110,
                "authKey": "0xzxgA9YoW9h58u8SEXAMPLE",
                "addressFamily": "ipv4",
                "amazonAddress": "192.168.1.1/30",
                "customerAddress": "192.168.1.2/30",
                "bgpPeerState": "pending",
                "bgpStatus": "down",
                "awsDeviceV2": "loc1-26wz6vEXAMPLE"
            }
        ],
        "region": "sa-east-1",
        "awsDeviceV2": "loc1-26wz6vEXAMPLE",
        "tags": [
            {
                "key": "Tag",
                "value": "Example"
            }
        ]
    }
}
```

如需詳細資訊，請參閱 Direct Connect 使用者指南中的建立 Direct Connect Gateway 的傳輸虛擬介面。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTransitVirtualInterface。

delete-bgp-peer

以下程式碼範例顯示如何使用 delete-bgp-peer。

AWS CLI

從虛擬介面刪除 BGP 對等

下列範例會從虛擬介面 刪除 IPv6 BGP 對等dxvif-fg1vuj3d。

命令：

```
aws directconnect delete-bgp-peer --virtual-interface-id dxvif-fg1vuj3d --asn 64600
 --customer-address 2001:db8:1100:2f0:0:1:9cb4:4216/125
```

輸出：

```
{
  "virtualInterface": {
      "virtualInterfaceState": "available",
      "asn": 65000,
      "vlan": 125,
      "customerAddress": "169.254.255.2/30",
      "ownerAccount": "123456789012",
      "connectionId": "dxcon-fguhmqlc",
      "addressFamily": "ipv4",
      "virtualGatewayId": "vgw-f9eb0c90",
      "virtualInterfaceId": "dxvif-fg1vuj3d",
      "authKey": "0xC_ukbCerl6EYA0example",
      "routeFilterPrefixes": [],
      "location": "EqDC2",
      "bgpPeers": [
          {
              "bgpStatus": "down",
              "customerAddress": "169.254.255.2/30",
              "addressFamily": "ipv4",
              "authKey": "0xC_ukbCerl6EYA0uexample",
              "bgpPeerState": "available",
              "amazonAddress": "169.254.255.1/30",
              "asn": 65000
          },
          {
              "bgpStatus": "down",
```

```
            "customerAddress": "2001:db8:1100:2f0:0:1:9cb4:4216/125",
            "addressFamily": "ipv6",
            "authKey": "0xS27kAIU_VHPjjAexample",
            "bgpPeerState": "deleting",
            "amazonAddress": "2001:db8:1100:2f0:0:1:9cb4:4211/125",
            "asn": 64600
        }
    ],
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-fg1vuj3d\">\n  <vlan>125</
vlan>\n  <customer_address>169.254.255.2/30</customer_address>\n
 <amazon_address>169.254.255.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
  <bgp_auth_key>0xC_ukbCerl6EYA0example</bgp_auth_key>\n  <amazon_bgp_asn>7224</
amazon_bgp_asn>\n  <connection_type>private</connection_type>\n</logical_connection>
\n",
    "amazonAddress": "169.254.255.1/30",
    "virtualInterfaceType": "private",
    "virtualInterfaceName": "Test"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBgpPeer。

delete-connection

以下程式碼範例顯示如何使用 delete-connection。

AWS CLI

刪除連線

下列delete-connection命令會刪除指定的連線：

```
aws directconnect delete-connection --connection-id dxcon-fg31dyv6
```

輸出：

```
{
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-fg31dyv6",
    "connectionState": "deleted",
    "bandwidth": "1Gbps",
```

```
    "location": "TIVIT",
    "connectionName": "Connection to AWS",
    "region": "sa-east-1"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteConnection。

delete-direct-connect-gateway-association

以下程式碼範例顯示如何使用 delete-direct-connect-gateway-association。

AWS CLI

刪除 Direct Connect 閘道關聯

下列delete-direct-connect-gateway-association範例會刪除 Direct Connect 閘道與具有指定關聯 ID 之傳輸閘道的關聯。

```
aws directconnect delete-direct-connect-gateway-association --association-id
  be85116d-46eb-4b43-a27a-da0c2ad648de
```

輸出：

```
{
    "directConnectGatewayAssociation": {
        "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPlE",
        "directConnectGatewayOwnerAccount": "123456789012",
        "associationState": "disassociating",
        "associatedGateway": {
            "id": "tgw-095b3b0b54EXAMPLE",
            "type": "transitGateway",
            "ownerAccount": "123456789012",
            "region": "us-east-1"
       },
        "associationId": " be85116d-46eb-4b43-a27a-da0c2ad648deEXAMPLE ",
        "allowedPrefixesToDirectConnectGateway": [
            {
                "cidr": "192.0.1.0/28"
            }
        ]
    }
}
```

如需詳細資訊，請參閱 AWS Direct Connect 使用者指南中的關聯和取消關聯傳輸閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDirectConnectGatewayAssociation。

delete-direct-connect-gateway

以下程式碼範例顯示如何使用 delete-direct-connect-gateway。

AWS CLI

刪除 Direct Connect 閘道

下列範例會刪除 Direct Connect 閘道 5f294f92-bafb-4011-916d-9b0bexample。

命令：

```
aws directconnect delete-direct-connect-gateway --direct-connect-gateway-
id 5f294f92-bafb-4011-916d-9b0bexample
```

輸出：

```
{
  "directConnectGateway": {
      "amazonSideAsn": 64512,
      "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
      "ownerAccount": "123456789012",
      "directConnectGatewayName": "DxGateway1",
      "directConnectGatewayState": "deleting"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDirectConnectGateway。

delete-interconnect

以下程式碼範例顯示如何使用 delete-interconnect。

AWS CLI

刪除互連

下列delete-interconnect命令會刪除指定的互連：

```
aws directconnect delete-interconnect --interconnect-id dxcon-fgktov66
```

輸出：

```
{
    "interconnectState": "deleted"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteInterconnect。

delete-lag

以下程式碼範例顯示如何使用 delete-lag。

AWS CLI

刪除 LAG

下列範例會刪除指定的 LAG。

命令：

```
aws directconnect delete-lag --lag-id dxlag-ffrhowd9
```

輸出：

```
{
  "awsDevice": "EqDC2-4h6ce2r1bes6",
  "numberOfConnections": 0,
  "lagState": "deleted",
  "ownerAccount": "123456789012",
  "lagName": "TestLAG",
  "connections": [],
  "lagId": "dxlag-ffrhowd9",
  "minimumLinks": 0,
  "connectionsBandwidth": "1Gbps",
  "region": "us-east-1",
  "location": "EqDC2"
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLag。

delete-virtual-interface

以下程式碼範例顯示如何使用 delete-virtual-interface。

AWS CLI

刪除虛擬界面

下列delete-virtual-interface命令會刪除指定的虛擬介面：

```
aws directconnect delete-virtual-interface --virtual-interface-id dxvif-ffhhk74f
```

輸出：

```
{
    "virtualInterfaceState": "deleting"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVirtualInterface。

describe-connection-loa

以下程式碼範例顯示如何使用 describe-connection-loa。

AWS CLI

使用 Linux 或 Mac OS X 描述連線的 LOA-CFA

下列範例說明連線 的 LOA-CFAdxcon-fh6ayh1d。LOA-CFA 的內容以 base64 編碼。此命令使用 --output和 --query 參數來控制輸出並擷取loaContent結構的內容。命令的最後一部分會使用 base64公用程式解碼內容，並將輸出傳送至 PDF 檔案。

```
aws directconnect describe-connection-loa --connection-id dxcon-fh6ayh1d --
output text --query loa.loaContent|base64 --decode > myLoaCfa.pdf
```

使用 Windows 描述連線的 LOA-CFA

先前的範例需要使用 base64公用程式來解碼輸出。在 Windows 電腦上，您可以certutil改為使用 。在下列範例中，第一個命令描述您的 LOA-CFA 以進行連線，dxcon-fh6ayh1d並使用 --output和 --query 參數來控制輸出，並將loaContent結構的內容擷取至名為 的檔案myLoaCfa.base64。第二個命令使用 certutil 公用程式將該檔案解碼並傳送輸出至 PDF 檔案。

```
aws directconnect describe-connection-loa --connection-id dxcon-fh6ayh1d --
output text --query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

如需控制 AWS CLI 輸出的詳細資訊，請參閱《 命令列界面使用者指南》中的從 AWS 命令列界面控制命令輸出。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeConnectionLoa。

describe-connections-on-interconnect

以下程式碼範例顯示如何使用 describe-connections-on-interconnect。

AWS CLI

列出互連上的連線

下列describe-connections-on-interconnect命令列出已在指定互連上佈建的連線：

```
aws directconnect describe-connections-on-interconnect --interconnect-id dxcon-
fgktov66
```

輸出：

```
{
    "connections": [
        {
            "partnerName": "TIVIT",
            "vlan": 101,
            "ownerAccount": "123456789012",
            "connectionId": "dxcon-ffzc51m1",
            "connectionState": "ordering",
            "bandwidth": "500Mbps",
```

```
            "location": "TIVIT",
            "connectionName": "mydcinterconnect",
            "region": "sa-east-1"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeConnectionsOnInterconnect。

describe-connections

以下程式碼範例顯示如何使用 describe-connections。

AWS CLI

列出目前區域中的所有連線

下列describe-connections命令會列出目前區域中的所有連線：

```
aws directconnect describe-connections
```

輸出：

```
{
  "connections": [
      {
          "awsDevice": "EqDC2-123h49s71dabc",
          "ownerAccount": "123456789012",
          "connectionId": "dxcon-fguhmqlc",
          "lagId": "dxlag-ffrz71kw",
          "connectionState": "down",
          "bandwidth": "1Gbps",
          "location": "EqDC2",
          "connectionName": "My_Connection",
          "loaIssueTime": 1491568964.0,
          "region": "us-east-1"
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeConnections。

describe-direct-connect-gateway-association-proposals

以下程式碼範例顯示如何使用 describe-direct-connect-gateway-association-proposals。

AWS CLI

描述您的 Direct Connect 閘道關聯提案

下列describe-direct-connect-gateway-association-proposals範例顯示 Direct Connect 閘道關聯提案的詳細資訊。

```
aws directconnect describe-direct-connect-gateway-association-proposals
```

輸出：

```
{
    "directConnectGatewayAssociationProposals": [
        {
            "proposalId": "c2ede9b4-bbc6-4d33-923c-bc4feEXAMPLE",
            "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
            "directConnectGatewayOwnerAccount": "111122223333",
            "proposalState": "requested",
            "associatedGateway": {
                "id": "tgw-02f776b1a7EXAMPLE",
                "type": "transitGateway",
                "ownerAccount": "111122223333",
                "region": "us-east-1"
            },
            "existingAllowedPrefixesToDirectConnectGateway": [
                {
                    "cidr": "192.168.2.0/30"
                },
                {
                    "cidr": "192.168.1.0/30"
                }
            ],
            "requestedAllowedPrefixesToDirectConnectGateway": [
                {
                    "cidr": "192.168.1.0/30"
                }
            ]
        },
```

```
        {
            "proposalId": "cb7f41cb-8128-43a5-93b1-dcaedEXAMPLE",
            "directConnectGatewayId": "11560968-4ac1-4fd3-bcb2-00599EXAMPLE",
            "directConnectGatewayOwnerAccount": "111122223333",
            "proposalState": "accepted",
            "associatedGateway": {
                "id": "tgw-045776b1a7EXAMPLE",
                "type": "transitGateway",
                "ownerAccount": "111122223333",
                "region": "us-east-1"
            },
            "existingAllowedPrefixesToDirectConnectGateway": [
                {
                    "cidr": "192.168.4.0/30"
                },
                {
                    "cidr": "192.168.5.0/30"
                }
            ],
            "requestedAllowedPrefixesToDirectConnectGateway": [
                {
                    "cidr": "192.168.5.0/30"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Direct Connect 使用者指南中的關聯和取消關聯傳輸閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDirectConnectGatewayAssociationProposals。

describe-direct-connect-gateway-associations

以下程式碼範例顯示如何使用 `describe-direct-connect-gateway-associations`。

AWS CLI

描述 Direct Connect 閘道關聯

下列範例說明與 Direct Connect 閘道 的所有關聯`5f294f92-bafb-4011-916d-9b0bexample`。

命令：

```
aws directconnect describe-direct-connect-gateway-associations --direct-connect-
gateway-id 5f294f92-bafb-4011-916d-9b0bexample
```

輸出：

```
{
  "nextToken":
 "eyJ2IjoxLCJzIjoxLCJpIjoiOU83OTFodzdycnZCbkN4MExHeHVwQT09IiwiYyI6InIxTEN0UEVHV0I1UFlkaWFnNl
  "directConnectGatewayAssociations": [
      {
          "associationState": "associating",
          "virtualGatewayOwnerAccount": "123456789012",
          "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
          "virtualGatewayId": "vgw-6efe725e",
          "virtualGatewayRegion": "us-east-2"
      },
      {
          "associationState": "disassociating",
          "virtualGatewayOwnerAccount": "123456789012",
          "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
          "virtualGatewayId": "vgw-ebaa27db",
          "virtualGatewayRegion": "us-east-2"
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDirectConnectGatewayAssociations。

describe-direct-connect-gateway-attachments

以下程式碼範例顯示如何使用 describe-direct-connect-gateway-attachments。

AWS CLI

描述 Direct Connect 閘道附件

下列範例說明連接至 Direct Connect 閘道 的虛擬介面5f294f92-bafb-4011-916d-9b0bexample。

命令：

```
aws directconnect describe-direct-connect-gateway-attachments --direct-connect-
gateway-id 5f294f92-bafb-4011-916d-9b0bexample
```

輸出：

```
{
  "directConnectGatewayAttachments": [
      {
          "virtualInterfaceOwnerAccount": "123456789012",
          "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
          "virtualInterfaceRegion": "us-east-2",
          "attachmentState": "attaching",
          "virtualInterfaceId": "dxvif-fg9zyabc"
      }
  ],
  "nextToken":
 "eyJ2IjoxLCJzIjoxLCJpIjoibEhXdlNpUXF5RzhoL1JyUW52SlV2QT09IiwiYyI6Im5wQjFHQ0RyQUdRS3puNnNXcU
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDirectConnectGatewayAttachments。

describe-direct-connect-gateways

以下程式碼範例顯示如何使用 `describe-direct-connect-gateways`。

AWS CLI

描述您的 Direct Connect 閘道

下列範例說明所有 Direct Connect 閘道。

命令：

```
aws directconnect describe-direct-connect-gateways
```

輸出：

```
{
  "directConnectGateways": [
      {
```

```
            "amazonSideAsn": 64512,
            "directConnectGatewayId": "cf68415c-f4ae-48f2-87a7-3b52cexample",
            "ownerAccount": "123456789012",
            "directConnectGatewayName": "DxGateway2",
            "directConnectGatewayState": "available"
        },
        {
            "amazonSideAsn": 64512,
            "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bdexample",
            "ownerAccount": "123456789012",
            "directConnectGatewayName": "DxGateway1",
            "directConnectGatewayState": "available"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDirectConnectGateways。

describe-hosted-connections

以下程式碼範例顯示如何使用 describe-hosted-connections。

AWS CLI

列出互連上的連線

下列範例列出已在指定互連上佈建的連線。

命令：

```
aws directconnect describe-hosted-connections --connection-id dxcon-fgktov66
```

輸出：

```
{
    "connections": [
        {
            "partnerName": "TIVIT",
            "vlan": 101,
            "ownerAccount": "123456789012",
            "connectionId": "dxcon-ffzc51m1",
            "connectionState": "ordering",
```

```
            "bandwidth": "500Mbps",
            "location": "TIVIT",
            "connectionName": "mydcinterconnect",
            "region": "sa-east-1"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeHostedConnections。

describe-interconnect-loa

以下程式碼範例顯示如何使用 describe-interconnect-loa。

AWS CLI

使用 Linux 或 Mac OS X 描述互連的 LOA-CFA

下列範例說明用於互連 的 LOA-CFAdxcon-fh6ayh1d。LOA-CFA 的內容以 base64 編碼。此命令使用 --output和 --query 參數來控制輸出並擷取loaContent結構的內容。命令的最後一部分會使用 base64公用程式解碼內容，並將輸出傳送至 PDF 檔案。

```
aws directconnect describe-interconnect-loa --interconnect-id dxcon-fh6ayh1d --
output text --query loa.loaContent|base64 --decode > myLoaCfa.pdf
```

使用 Windows 描述 互連的 LOA-CFA

先前的範例需要使用 base64公用程式來解碼輸出。在 Windows 電腦上，您可以certutil改為使用 。在下列範例中，第一個命令描述 LOA-CFA 以進行互連，dxcon-fh6ayh1d並使用 --output和 --query 參數來控制輸出，並將loaContent結構的內容擷取至名為 的檔案myLoaCfa.base64。第二個命令使用 certutil 公用程式將該檔案解碼並傳送輸出至 PDF 檔案。

```
aws directconnect describe-interconnect-loa --interconnect-id dxcon-fh6ayh1d --
output text --query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

如需控制 AWS CLI 輸出的詳細資訊，請參閱《 命令列界面使用者指南》中的從 AWS 命令列界面控制命令輸出。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInterconnectLoa。

describe-interconnects

以下程式碼範例顯示如何使用 describe-interconnects。

AWS CLI

列出互連

下列describe-interconnects命令會列出 AWS 您的帳戶擁有的互連：

```
aws directconnect describe-interconnects
```

輸出：

```
{
    "interconnects": [
        {
            "region": "sa-east-1",
            "bandwidth": "1Gbps",
            "location": "TIVIT",
            "interconnectName": "1G Interconnect to AWS",
            "interconnectId": "dxcon-fgktov66",
            "interconnectState": "down"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInterconnects。

describe-lags

以下程式碼範例顯示如何使用 describe-lags。

AWS CLI

描述您的 LAGs

下列命令說明目前區域的所有 LAGs。

命令：

```
aws directconnect describe-lags
```

輸出：

```
{
  "lags": [
      {
          "awsDevice": "EqDC2-19y7z3m17xpuz",
          "numberOfConnections": 2,
          "lagState": "down",
          "ownerAccount": "123456789012",
          "lagName": "DA-LAG",
          "connections": [
              {
                  "ownerAccount": "123456789012",
                  "connectionId": "dxcon-ffnikghc",
                  "lagId": "dxlag-fgsu9erb",
                  "connectionState": "requested",
                  "bandwidth": "10Gbps",
                  "location": "EqDC2",
                  "connectionName": "Requested Connection 1 for Lag dxlag-fgsu9erb",
                  "region": "us-east-1"
              },
              {
                  "ownerAccount": "123456789012",
                  "connectionId": "dxcon-fglgbdea",
                  "lagId": "dxlag-fgsu9erb",
                  "connectionState": "requested",
                  "bandwidth": "10Gbps",
                  "location": "EqDC2",
                  "connectionName": "Requested Connection 2 for Lag dxlag-fgsu9erb",
                  "region": "us-east-1"
              }
          ],
          "lagId": "dxlag-fgsu9erb",
          "minimumLinks": 0,
          "connectionsBandwidth": "10Gbps",
          "region": "us-east-1",
          "location": "EqDC2"
      }
  ]
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLags。

describe-loa

以下程式碼範例顯示如何使用 describe-loa。

AWS CLI

使用 Linux 或 Mac OS X 描述連線的 LOA-CFA

下列範例說明連線 的 LOA-CFAdxcon-fh6ayh1d。LOA-CFA 的內容以 base64 編碼。此命令使用 --output和 --query 參數來控制輸出並擷取loaContent結構的內容。命令的最後一部分會使用 base64公用程式解碼內容，並將輸出傳送至 PDF 檔案。

```
aws directconnect describe-loa --connection-id dxcon-fh6ayh1d --output text --
query loa.loaContent|base64 --decode > myLoaCfa.pdf
```

使用 Windows 描述連線的 LOA-CFA

先前的範例需要使用 base64公用程式來解碼輸出。在 Windows 電腦上，您可以certutil改為使用 。在下列範例中，第一個命令描述您的 LOA-CFA 以進行連線，dxcon-fh6ayh1d並使用 --output和 --query 參數來控制輸出，並將loaContent結構的內容擷取至名為 的檔案myLoaCfa.base64。第二個命令使用 certutil 公用程式將該檔案解碼並傳送輸出至 PDF 檔案。

```
aws directconnect describe-loa --connection-id dxcon-fh6ayh1d --output text --
query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

如需控制 AWS CLI 輸出的詳細資訊，請參閱《 命令列界面使用者指南》中的從 AWS 命令列界面控制命令輸出。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLoa。

describe-locations

以下程式碼範例顯示如何使用 describe-locations。

AWS CLI

列出 AWS Direct Connect 合作夥伴和位置

下列`describe-locations`命令列出目前區域中的 AWS Direct Connect 合作夥伴和位置：

```
aws directconnect describe-locations
```

輸出：

```
{
    "locations": [
        {
            "locationName": "NAP do Brasil, Barueri, Sao Paulo",
            "locationCode": "TNDB"
        },
        {
            "locationName": "Tivit - Site Transamerica (Sao Paulo)",
            "locationCode": "TIVIT"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLocations。

`describe-tags`

以下程式碼範例顯示如何使用 `describe-tags`。

AWS CLI

描述 AWS Direct Connect 資源的標籤

下列命令說明連線 的標籤`dxcon-abcabc12`。

命令：

```
aws directconnect describe-tags --resource-arns arn:aws:directconnect:us-
east-1:123456789012:dxcon/dxcon-abcabc12
```

輸出：

```
{
  "resourceTags": [
      {
          "resourceArn": "arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-
abcabc12",
          "tags": [
              {
                  "value": "VAConnection",
                  "key": "Name"
              }
          ]
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTags。

describe-virtual-gateways

以下程式碼範例顯示如何使用 describe-virtual-gateways。

AWS CLI

列出虛擬私有閘道

下列describe-virtual-gateways命令會列出您 AWS 帳戶擁有的虛擬私有閘道：

```
aws directconnect describe-virtual-gateways
```

輸出：

```
{
    "virtualGateways": [
        {
            "virtualGatewayId": "vgw-aba37db6",
            "virtualGatewayState": "available"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVirtualGateways。

describe-virtual-interfaces

以下程式碼範例顯示如何使用 describe-virtual-interfaces。

AWS CLI

列出所有虛擬介面

下列describe-virtual-interfaces命令會列出與 AWS 您的帳戶相關聯的所有虛擬介面的相關資訊：

```
aws directconnect describe-virtual-interfaces --connection-id dxcon-ffjrkx17
```

輸出：

```
{
    "virtualInterfaces": [
        {
            "virtualInterfaceState": "down",
            "asn": 65000,
            "vlan": 101,
            "customerAddress": "192.168.1.2/30",
            "ownerAccount": "123456789012",
            "connectionId": "dxcon-ffjrkx17",
            "virtualGatewayId": "vgw-aba37db6",
            "virtualInterfaceId": "dxvif-ffhhk74f",
            "authKey": "asdf34example",
            "routeFilterPrefixes": [],
            "location": "TIVIT",
            "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-ffhhk74f\">\n  <vlan>101</
vlan>\n  <customer_address>192.168.1.2/30</customer_address>\n
 <amazon_address>192.168.1.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
 <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\n  <connection_type>private</connection_type>\n</logical_connection>\n",
            "amazonAddress": "192.168.1.1/30",
            "virtualInterfaceType": "private",
            "virtualInterfaceName": "PrivateVirtualInterface"
        },
        {
            "virtualInterfaceState": "verifying",
            "asn": 65000,
            "vlan": 2000,
```

```
            "customerAddress": "203.0.113.2/30",
            "ownerAccount": "123456789012",
            "connectionId": "dxcon-ffjrkx17",
            "virtualGatewayId": "",
            "virtualInterfaceId": "dxvif-fgh0hcrk",
            "authKey": "asdf34example",
            "routeFilterPrefixes": [
                {
                    "cidr": "203.0.113.4/30"
                },
                {
                    "cidr": "203.0.113.0/30"
                }
            ],
            "location": "TIVIT",
            "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-fgh0hcrk\">\n  <vlan>2000</
vlan>\n  <customer_address>203.0.113.2/30</customer_address>\n
 <amazon_address>203.0.113.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
 <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\n  <connection_type>public</connection_type>\n</logical_connection>\n",
            "amazonAddress": "203.0.113.1/30",
            "virtualInterfaceType": "public",
            "virtualInterfaceName": "PublicVirtualInterface"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVirtualInterfaces。

disassociate-connection-from-lag

以下程式碼範例顯示如何使用 disassociate-connection-from-lag。

AWS CLI

取消連線與 LAG 的關聯

下列範例會將指定的連線與指定的 LAG 取消關聯。

命令：

```
aws directconnect disassociate-connection-from-lag --lag-id dxlag-fhccu14t --
connection-id  dxcon-fg9607vm
```

輸出：

```
{
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg9607vm",
  "connectionState": "requested",
  "bandwidth": "1Gbps",
  "location": "EqDC2",
  "connectionName": "Con2ForLag",
  "region": "us-east-1"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateConnectionFromLag。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤新增至 AWS Direct Connect 資源

下列命令會將索引鍵為 Name且值為 的標籤VAConnection新增至連線 dxcon-abcabc12。如果命令成功，則不會傳回任何輸出。

命令：

```
aws directconnect tag-resource --resource-arn arn:aws:directconnect:us-
east-1:123456789012:dxcon/dxcon-abcabc12 --tags "key=Name,value=VAConnection"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從 AWS Direct Connect 資源移除標籤

下列命令Name會從連線 移除具有 金鑰的標籤dxcon-abcabc12。如果命令成功，則不會傳回任何輸出。

命令：

```
aws directconnect untag-resource --resource-arn arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-abcabc12 --tag-keys Name
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-direct-connect-gateway-association

以下程式碼範例顯示如何使用 update-direct-connect-gateway-association。

AWS CLI

更新 Direct Connect 閘道關聯的指定屬性

下列update-direct-connect-gateway-association範例會將指定的 CIDR 區塊新增至 Direct Connect 閘道關聯。

```
aws directconnect update-direct-connect-gateway-association \
    --association-id 820a6e4f-5374-4004-8317-3f64bEXAMPLE \
    --add-allowed-prefixes-to-direct-connect-gateway cidr=192.168.2.0/30
```

輸出：

```
{
    "directConnectGatewayAssociation": {
        "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
        "directConnectGatewayOwnerAccount": "111122223333",
        "associationState": "updating",
        "associatedGateway": {
            "id": "tgw-02f776b1a7EXAMPLE",
            "type": "transitGateway",
            "ownerAccount": "111122223333",
            "region": "us-east-1"
```

```
        },
        "associationId": "820a6e4f-5374-4004-8317-3f64bEXAMPLE",
        "allowedPrefixesToDirectConnectGateway": [
            {
                "cidr": "192.168.2.0/30"
            },
            {
                "cidr": "192.168.1.0/30"
            }
        ]
    }
}
```

如需詳細資訊，請參閱 Direct Connect 使用者指南中的使用 Direct Connect 閘道。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDirectConnectGatewayAssociation。

update-lag

以下程式碼範例顯示如何使用 update-lag。

AWS CLI

更新 LAG

下列範例會變更指定 LAG 的名稱。

命令：

```
aws directconnect update-lag --lag-id dxlag-ffjhj9lx --lag-name 2ConnLag
```

輸出：

```
{
  "awsDevice": "CSVA1-23u8tlpaz8iks",
  "numberOfConnections": 2,
  "lagState": "down",
  "ownerAccount": "123456789012",
  "lagName": "2ConnLag",
  "connections": [
      {
```

```
            "ownerAccount": "123456789012",
            "connectionId": "dxcon-fflqyj95",
            "lagId": "dxlag-ffjhj9lx",
            "connectionState": "requested",
            "bandwidth": "1Gbps",
            "location": "CSVA1",
            "connectionName": "Requested Connection 2 for Lag dxlag-ffjhj9lx",
            "region": "us-east-1"
        },
        {
            "ownerAccount": "123456789012",
            "connectionId": "dxcon-ffqr6x5q",
            "lagId": "dxlag-ffjhj9lx",
            "connectionState": "requested",
            "bandwidth": "1Gbps",
            "location": "CSVA1",
            "connectionName": "Requested Connection 1 for Lag dxlag-ffjhj9lx",
            "region": "us-east-1"
        }
    ],
    "lagId": "dxlag-ffjhj9lx",
    "minimumLinks": 0,
    "connectionsBandwidth": "1Gbps",
    "region": "us-east-1",
    "location": "CSVA1"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateLag。

update-virtual-interface-attributes

以下程式碼範例顯示如何使用 update-virtual-interface-attributes。

AWS CLI

更新虛擬介面的 MTU

下列update-virtual-interface-attributes範例會更新指定虛擬介面的 MTU。

```
aws directconnect update-virtual-interface-attributes \
    --virtual-interface-id dxvif-fEXAMPLE \
    --mtu 1500
```

輸出：


```
{
    "ownerAccount": "1111222233333",
    "virtualInterfaceId": "dxvif-fEXAMPLE",
    "location": "loc1",
    "connectionId": "dxlag-fEXAMPLE",
    "virtualInterfaceType": "transit",
    "virtualInterfaceName": "example transit virtual interface",
    "vlan": 125,
    "asn": 650001,
    "amazonSideAsn": 64512,
    "authKey": "0xzxgA9YoW9h58u8SEXAMPLE",
    "amazonAddress": "169.254.248.1/30",
    "customerAddress": "169.254.248.2/30",
    "addressFamily": "ipv4",
    "virtualInterfaceState": "down",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>\n<logical_connection id=\"dxvif-fEXAMPLE\">\n  <vlan>125</vlan>
\n  <customer_address>169.254.248.2/30</customer_address>\n
 <amazon_address>169.254.248.1/30</amazon_address>\n  <bgp_asn>650001</bgp_asn>\n
  <bgp_auth_key>0xzxgA9YoW9h58u8SEXAMPLE</bgp_auth_key>\n  <amazon_bgp_asn>64512</
amazon_bgp_asn>\n  <connection_type>transit</connection_type>\n</logical_connection>
\n",
    "mtu": 1500,
    "jumboFrameCapable": true,
    "virtualGatewayId": "",
    "directConnectGatewayId": "879b76a1-403d-4700-8b53-4a56ed85436e",
    "routeFilterPrefixes": [],
    "bgpPeers": [
        {
            "bgpPeerId": "dxpeer-fEXAMPLE",
            "asn": 650001,
            "authKey": "0xzxgA9YoW9h58u8SEXAMPLE",
            "addressFamily": "ipv4",
            "amazonAddress": "169.254.248.1/30",
            "customerAddress": "169.254.248.2/30",
            "bgpPeerState": "available",
            "bgpStatus": "down",
            "awsDeviceV2": "loc1-26wz6vEXAMPLE"
        }
    ],
    "region": "sa-east-1",
    "awsDeviceV2": "loc1-26wz6vEXAMPLE",
```

```
    "tags": []
}
```

如需詳細資訊，請參閱 AWS Direct Connect 使用者指南中的設定私有虛擬介面的網路 MTU 或傳輸虛擬介面。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateVirtualInterfaceAttributes。

AWS Directory Service 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS Directory Service。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`describe-directories`

以下程式碼範例顯示如何使用 `describe-directories`。

AWS CLI

取得目錄的詳細資訊

下列`describe-directories`範例顯示指定目錄的詳細資訊。

```
aws ds describe-directories \
   --directory-id d-a1b2c3d4e5
```

輸出：

```
{
```

```
    "DirectoryDescriptions": [
        {
            "DirectoryId": "d-a1b2c3d4e5",
            "Name": "mydirectory.example.com",
            "ShortName": "mydirectory",
            "Size": "Small",
            "Edition": "Standard",
            "Alias": "d-a1b2c3d4e5",
            "AccessUrl": "d-a1b2c3d4e5.awsapps.com",
            "Stage": "Active",
            "ShareStatus": "Shared",
            "ShareMethod": "HANDSHAKE",
            "ShareNotes": "These are my share notes",
            "LaunchTime": "2019-07-08T15:33:46.327000-07:00",
            "StageLastUpdatedDateTime": "2019-07-08T15:59:12.307000-07:00",
            "Type": "SharedMicrosoftAD",
            "SsoEnabled": false,
            "DesiredNumberOfDomainControllers": 0,
            "OwnerDirectoryDescription": {
                "DirectoryId": "d-b2c3d4e5f6",
                "AccountId": "123456789111",
                "DnsIpAddrs": [
                    "203.113.0.248",
                    "203.113.0.253"
                ],
                "VpcSettings": {
                    "VpcId": "vpc-a1b2c3d4",
                    "SubnetIds": [
                        "subnet-a1b2c3d4",
                        "subnet-d4c3b2a1"
                    ],
                    "AvailabilityZones": [
                        "us-west-2a",
                        "us-west-2c"
                    ]
                }
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDirectories。

describe-trusts

以下程式碼範例顯示如何使用 describe-trusts。

AWS CLI

取得有關信任關係的詳細資訊

下列describe-trusts範例顯示指定目錄之信任關係的詳細資訊。

```
aws ds describe-trusts \
   --directory-id d-a1b2c3d4e5
```

輸出：

```
{
    "Trusts": [
        {
            "DirectoryId": "d-a1b2c3d4e5",
            "TrustId": "t-9a8b7c6d5e",
            "RemoteDomainName": "other.example.com",
            "TrustType": "Forest",
            "TrustDirection": "Two-Way",
            "TrustState": "Verified",
            "CreatedDateTime": "2017-06-20T18:08:45.614000-07:00",
            "LastUpdatedDateTime": "2019-06-04T10:52:12.410000-07:00",
            "StateLastUpdatedDateTime": "2019-06-04T10:52:12.410000-07:00",
            "SelectiveAuth": "Disabled"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTrusts。

AWS Directory Service 使用 的資料範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 AWS Directory Service 資料來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`add-group-member`

以下程式碼範例顯示如何使用 `add-group-member`。

AWS CLI

將群組成員新增至目錄

下列`add-group-member`範例會將指定的使用者新增至指定目錄中的指定群組。

```
aws ds-data add-group-member \
    --directory-id d-1234567890 \
    --group-name 'sales' \
    --member-name 'john.doe'
```

此命令不會產生輸出。

如需詳細資訊，請參閱AWS 《 Directory Service 管理指南》中的將 AWS Managed Microsoft AD 成員新增至群組或將群組新增至群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddGroupMember。

`create-group`

以下程式碼範例顯示如何使用 `create-group`。

AWS CLI

列出可用的小工具

下列`create-group`範例會在指定的目錄中建立群組。

```
aws ds-data create-group \
    --directory-id d-1234567890 \
```

```
    --sam-account-name "sales"
```

輸出：

```
{
    "DirectoryId": "d-1234567890",
    "SAMAccountName": "sales",
    "SID": "S-1-2-34-5567891234-5678912345-67891234567-8912"
}
```

如需詳細資訊，請參閱《 AWS Directory Service 管理指南》中的建立 AWS Managed Microsoft AD 群組。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateGroup。

create-user

以下程式碼範例顯示如何使用 create-user。

AWS CLI

建立使用者

下列create-user範例會在指定的目錄中建立使用者。

```
aws ds-data create-user \
    --directory-id d-1234567890 \
    --sam-account-name 'john.doe'
```

輸出：

```
{
    "DirectoryId": "d-1234567890",
    "SAMAccountName": "john.doe",
    "SID": "S-1-2-34-5567891234-5678912345-67891234567-8912"
}
```

如需詳細資訊，請參閱《 AWS Directory Service 管理指南》中的建立 AWS Managed Microsoft AD 使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateUser。

delete-group

以下程式碼範例顯示如何使用 delete-group。

AWS CLI

刪除群組

下列delete-group範例會從指定的目錄刪除指定的群組。

```
aws ds-data delete-group \
    --directory-id d-1234567890 \
    --sam-account-name 'sales'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 AWS Directory Service 管理指南》中的刪除 AWS Managed Microsoft AD 群組。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteGroup。

delete-user

以下程式碼範例顯示如何使用 delete-user。

AWS CLI

若要刪除使用者

下列delete-user範例會從指定的目錄刪除指定的使用者。

```
aws ds-data delete-user \
    --directory-id d-1234567890 \
    --sam-account-name 'john.doe'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 AWS Directory Service 管理指南》中的刪除 AWS Managed Microsoft AD 使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteUser。

describe-group

以下程式碼範例顯示如何使用 describe-group。

AWS CLI

列出群組的詳細資訊

下列describe-group範例會取得指定目錄中指定群組的資訊。

```
aws ds-data describe-group \
    --directory-id d-1234567890 \
    --sam-account-name 'sales'
```

輸出：

```
{
    "DirectoryId": "d-1234567890",
    "DistinguishedName": "CN=sales,OU=Users,OU=CORP,DC=corp,DC=example,DC=com",
    "GroupScope": "Global",
    "GroupType": "Security",
    "Realm": "corp.example.com",
    "SAMAccountName": "sales",
    "SID": "S-1-2-34-5567891234-5678912345-67891234567-8912"
}
```

如需詳細資訊，請參閱《 AWS Directory Service 管理指南》中的檢視和更新 AWS Managed Microsoft AD 群組的詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeGroup。

describe-user

以下程式碼範例顯示如何使用 describe-user。

AWS CLI

列出使用者的資訊

下列describe-user範例會取得指定目錄中指定使用者的資訊。

```
aws ds-data describe-user command-name \
    --directory-id d-1234567890 \
```

```
    --sam-account-name 'john.doe'
```

輸出：

```
{
    "DirectoryId": "d-1234567890",
    "DistinguishedName": "CN=john.doe,OU=Users,OU=CORP,DC=corp,DC=example,DC=com",
    "Enabled": false,
    "Realm": "corp.example.com",
    "SAMAccountName": "john.doe",
    "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567",
    "UserPrincipalName": "john.doe@CORP.EXAMPLE.COM"
}
```

如需詳細資訊，請參閱《 AWS Directory Service 管理指南》中的檢視和更新 AWS Managed Microsoft AD 使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeUser。

disable-directory-data-access

以下程式碼範例顯示如何使用 disable-directory-data-access。

AWS CLI

停用目錄的目錄服務資料 API

下列disable-directory-data-access範例會停用指定目錄的 Directory Service Data API。

```
aws ds disable-directory-data-access \
    --directory-id d-1234567890
```

此命令不會產生輸出。

如需詳細資訊，請參閱AWS 《 Directory Service 管理指南》中的啟用或停用使用者和群組管理或 Directory Service Data。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableDirectoryDataAccess。

disable-user

以下程式碼範例顯示如何使用 disable-user。

AWS CLI

停用使用者

下列disable-user範例會停用指定目錄中的指定使用者。

```
aws ds-data disable-user \
    --directory-id d-1234567890 \
    --sam-account-name 'john.doe'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 AWS Directory Service 管理指南》中的停用 AWS Managed Microsoft AD 使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableUser。

enable-directory-data-access

以下程式碼範例顯示如何使用 enable-directory-data-access。

AWS CLI

為目錄啟用目錄服務資料 API

下列enable-directory-data-access範例會啟用指定目錄的 Directory Service Data API。

```
aws ds enable-directory-data-access \
    --directory-id d-1234567890
```

此命令不會產生輸出。

如需詳細資訊，請參閱AWS 《目錄服務管理指南》中的啟用或停用使用者和群組管理或目錄服務資料。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableDirectoryDataAccess。

list-group-members

以下程式碼範例顯示如何使用 list-group-members。

AWS CLI

列出目錄的群組成員

下列list-group-members範例列出指定目錄中指定群組的群組成員。

```
aws ds-data list-group-members \
    --directory-id d-1234567890 \
    --sam-account-name 'sales'
```

輸出：

```
{
    "Members": [
        {
            "MemberType": "USER",
            "SAMAccountName": "Jane Doe",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4568"
        },
        {
            "MemberType": "USER",
            "SAMAccountName": "John Doe",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4569"
        }
    ],
    "DirectoryId": "d-1234567890",
    "MemberRealm": "corp.example.com",
    "Realm": "corp.example.com"
}
```

如需詳細資訊，請參閱《 AWS Directory Service 管理指南》中的檢視和更新 AWS Managed Microsoft AD 群組的詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListGroupMembers。

list-groups-for-member

以下程式碼範例顯示如何使用 list-groups-for-member。

AWS CLI

列出目錄的群組成員資格

下列list-groups-for-member範例列出指定目錄中指定使用者的群組成員資格。

```
aws ds-data list-groups-for-member \
    --directory-id d-1234567890 \
    --sam-account-name 'john.doe'
```

輸出：

```
{
    "Groups": [
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Domain Users",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
        }
    ],
    "DirectoryId": "d-1234567890",
    "MemberRealm": "corp.example.com",
    "Realm": "corp.example.com"
}
```

如需詳細資訊，請參閱《 AWS Directory Service 管理指南》中的檢視和更新 AWS Managed Microsoft AD 使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListGroupsForMember。

list-groups

以下程式碼範例顯示如何使用 list-groups。

AWS CLI

列出目錄的群組

下列list-groups範例列出指定目錄中的群組。

```
aws ds-data list-groups \
    --directory-id d-1234567890
```

輸出：

```
{
    "Groups": [
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Administrators",
            "SID": "S-1-2-33-441"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Users",
            "SID": "S-1-2-33-442"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Guests",
            "SID": "S-1-2-33-443"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Print Operators",
            "SID": "S-1-2-33-444"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Backup Operators",
            "SID": "S-1-2-33-445"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Replicator",
            "SID": "S-1-2-33-446"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Remote Desktop Users",
            "SID": "S-1-2-33-447"
```

```
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Network Configuration Operators",
            "SID": "S-1-2-33-448"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Performance Monitor Users",
            "SID": "S-1-2-33-449"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Performance Log Users",
            "SID": "S-1-2-33-450"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Distributed COM Users",
            "SID": "S-1-2-33-451"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "IIS_IUSRS",
            "SID": "S-1-2-33-452"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Cryptographic Operators",
            "SID": "S-1-2-33-453"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Event Log Readers",
            "SID": "S-1-2-33-454"
        },
        {
```

```
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Certificate Service DCOM Access",
            "SID": "S-1-2-33-456"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "RDS Remote Access Servers",
            "SID": "S-1-2-33-457"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "RDS Endpoint Servers",
            "SID": "S-1-2-33-458"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "RDS Management Servers",
            "SID": "S-1-2-33-459"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Hyper-V Administrators",
            "SID": "S-1-2-33-460"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Access Control Assistance Operators",
            "SID": "S-1-2-33-461"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Remote Management Users",
            "SID": "S-1-2-33-462"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
```

```
            "SAMAccountName": "Storage Replica Administrators",
            "SID": "S-1-2-33-463"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Domain Computers",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-789"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Domain Controllers",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-790"
        },
        {
            "GroupScope": "Universal",
            "GroupType": "Security",
            "SAMAccountName": "Schema Admins",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-791"
        },
        {
            "GroupScope": "Universal",
            "GroupType": "Security",
            "SAMAccountName": "Enterprise Admins",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-792"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "Cert Publishers",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-793"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Domain Admins",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-794"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Domain Users",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-795"
```

```
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Domain Guests",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-796"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Group Policy Creator Owners",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-797"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "RAS and IAS Servers",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-798"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Server Operators",
            "SID": "S-1-2-33-464"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Account Operators",
            "SID": "S-1-2-33-465"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Pre-Windows 2000 Compatible Access",
            "SID": "S-1-2-33-466"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Incoming Forest Trust Builders",
            "SID": "S-1-2-33-467"
        },
        {
```

```
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Windows Authorization Access Group",
            "SID": "S-1-2-33-468"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Terminal Server License Servers",
            "SID": "S-1-2-33-469"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "Allowed RODC Password Replication Group",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-798"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "Denied RODC Password Replication Group",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-799"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Read-only Domain Controllers",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-800"
        },
        {
            "GroupScope": "Universal",
            "GroupType": "Security",
            "SAMAccountName": "Enterprise Read-only Domain Controllers",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-801"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Cloneable Domain Controllers",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-802"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
```

```
            "SAMAccountName": "Protected Users",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-803"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Key Admins",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-804"
        },
        {
            "GroupScope": "Universal",
            "GroupType": "Security",
            "SAMAccountName": "Enterprise Key Admins",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-805"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "DnsAdmins",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "DnsUpdateProxy",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4568"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "Admins",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4569"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWSAdministrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4570"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Object Management Service Accounts",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4571"
```

```
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Private CA Connector for AD Delegated Group",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4572"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Application and Service Delegated Group",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4573"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4574"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated FSx Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4575"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Account Operators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4576"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Active Directory Based Activation
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4577"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Allowed to Authenticate Objects",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4578"
        },
```

```
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Allowed to Authenticate to Domain
Controllers",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4579"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Deleted Object Lifetime
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4580"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Distributed File System
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4581"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Dynamic Host Configuration Protocol
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4582"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Enterprise Certificate Authority
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4583"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Fine Grained Password Policy
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4584"
        },
        {
            "GroupScope": "DomainLocal",
```

```
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Group Policy Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4585"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Managed Service Account
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4586"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Read Foreign Security Principals",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4587"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Remote Access Service Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4588"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Replicate Directory Changes
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4588"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Sites and Services Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4589"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated System Management Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4590"
        },
        {
            "GroupScope": "DomainLocal",
```

```
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Terminal Server Licensing
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4591"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated User Principal Name Suffix
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4592"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Add Workstations To Domain Users",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4593"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Domain Name System Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4594"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Kerberos Delegation Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4595"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Server Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4596"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated MS-NPRC Non-Compliant Devices",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4597"
        },
        {
            "GroupScope": "Global",
```

```
            "GroupType": "Security",
            "SAMAccountName": "Remote Access",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4598"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Accounting",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4599"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Distribution",
            "SAMAccountName": "sales",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
        }
    ],
    "DirectoryId": "d-1234567890",
    "Realm": "corp.example.com"
}
```

如需詳細資訊，請參閱《 AWS Directory Service 管理指南》中的檢視和更新 AWS Managed Microsoft AD 群組的詳細資訊。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListGroups。

`list-users`

以下程式碼範例顯示如何使用 `list-users`。

AWS CLI

列出目錄的使用者

下列`list-users`範例列出指定目錄中的使用者。

```
aws ds-data list-users \
    --directory-id d-1234567890
```

輸出：

```
{
    "Users": [
```

```
        {
            "Enabled": true,
            "SAMAccountName": "Administrator",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-345"
        },
        {
            "Enabled": false,
            "SAMAccountName": "Guest",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-345"
        },
        {
            "Enabled": false,
            "SAMAccountName": "krbtgt",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-346"
        },
        {
            "Enabled": true,
            "SAMAccountName": "Admin",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-347"
        },
        {
            "Enabled": true,
            "SAMAccountName": "Richard Roe",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-348"
        },
        {
            "Enabled": true,
            "SAMAccountName": "Jane Doe",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-349"
        },
        {
            "Enabled": true,
            "SAMAccountName": "AWS_WGnzYlN6YyY",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
        },
        {
            "Enabled": true,
            "SAMAccountName": "john.doe",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4568"
        }
    ],
    "DirectoryId": "d-1234567890",
    "Realm": "corp.example.com"
```

```
}
```

如需詳細資訊，請參閱《 AWS Directory Service 管理指南》中的檢視和更新 AWS Managed Microsoft AD 使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListUsers。

remove-group-member

以下程式碼範例顯示如何使用 remove-group-member。

AWS CLI

從目錄移除群組成員

下列remove-group-member範例會從指定目錄中的指定群組移除指定的群組成員。

```
aws ds-data remove-group-member \
    --directory-id d-1234567890 \
    --group-name 'sales' \
    --member-name 'john.doe'
```

此命令不會產生輸出。

如需詳細資訊，請參閱AWS 《 Directory Service 管理指南》中的將 AWS Managed Microsoft AD 成員新增至群組，並將群組新增至群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveGroupMember。

reset-user-password

以下程式碼範例顯示如何使用 reset-user-password。

AWS CLI

在目錄中重設使用者密碼

下列reset-user-password範例會重設並啟用指定目錄中的指定使用者。

```
aws ds reset-user-password \
    --directory-id d-1234567890 \
    --user-name 'john.doe' \
```

```
    --new-password 'password'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 AWS Directory Service 管理指南》中的重設和啟用 AWS Managed Microsoft AD 使用者的密碼。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetUserPassword。

search-groups

以下程式碼範例顯示如何使用 search-groups。

AWS CLI

在目錄中搜尋群組

下列search-groups範例會搜尋指定目錄中的指定群組。

```
aws ds-data search-groups \
    --directory-id d-1234567890 \
    --search-attributes 'SamAccountName' \
    --search-string 'sales'
```

輸出：

```
{
    "Groups": [
        {
            "GroupScope": "Global",
            "GroupType": "Distribution",
            "SAMAccountName": "sales",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
        }
    ],
    "DirectoryId": "d-1234567890",
    "Realm": "corp.example.com"
}
```

如需詳細資訊，請參閱《 AWS Directory Service 管理指南》中的檢視和更新 AWS Managed Microsoft AD 群組的詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchGroups。

search-users

以下程式碼範例顯示如何使用 search-users。

AWS CLI

在目錄中搜尋使用者

下列search-users範例會在指定的目錄中搜尋指定的使用者。

```
aws ds-data search-users \
    --directory-id d-1234567890 \
    --search-attributes 'SamAccountName' \
    --Search-string 'john.doe'
```

輸出：

```
{
    "Users": [
        {
            "Enabled": true,
            "SAMAccountName": "john.doe",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
        }
    ],
    "DirectoryId": "d-1234567890",
    "Realm": "corp.example.com"
}
```

如需詳細資訊，請參閱《 AWS Directory Service 管理指南》中的檢視和更新 AWS Managed Microsoft AD 使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchUsers。

update-group

以下程式碼範例顯示如何使用 update-group。

AWS CLI

在目錄中更新群組的屬性

下列update-group範例會更新指定目錄中指定群組的指定屬性。

```
aws ds-data update-group \
    --directory-id d-1234567890 \
    --sam-account-name 'sales' \
    --update-type 'REPLACE' \
    --group-type 'Distribution'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 AWS Directory Service 管理指南》中的檢視和更新 AWS Managed Microsoft AD 群組的詳細資訊。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UpdateGroup。

update-user

以下程式碼範例顯示如何使用 update-user。

AWS CLI

在目錄中更新使用者的屬性

下列update-user範例會更新指定目錄中指定使用者的指定屬性。

```
aws ds-data update-user \
    --directory-id d-1234567890 \
    --sam-account-name 'john.doe' \
    --update-type 'ADD' \
    --email-address 'example.corp.com'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 AWS Directory Service 管理指南》中的檢視和更新 AWS Managed Microsoft AD 使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateUser。

AWS DMS 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS DMS。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

add-tags-to-resource

以下程式碼範例顯示如何使用 add-tags-to-resource。

AWS CLI

將標籤新增至資源

下列add-tags-to-resource範例會將標籤新增至複寫執行個體。

```
aws dms add-tags-to-resource \
    --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
 \
    --tags Key=Environment,Value=PROD Key=Project,Value=dbMigration
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 AWS Database Migration Service 使用者指南》中的標記資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AddTagsToResource。

create-endpoint

以下程式碼範例顯示如何使用 create-endpoint。

AWS CLI

建立端點

下列create-endpoint範例會建立 Amazon S3 來源的端點。

```
aws dms create-endpoint \
    --endpoint-type source \
```

```
    --engine-name s3 \
    --endpoint-identifier src-endpoint \
    --s3-settings file://s3-settings.json
```

s3-settings.json 的內容：

```
{
    "BucketName":"my-corp-data",
    "BucketFolder":"sourcedata",
    "ServiceAccessRoleArn":"arn:aws:iam::123456789012:role/my-s3-access-role"
}
```

輸出：

```
{
    "Endpoint": {
        "EndpointIdentifier": "src-endpoint",
        "EndpointType": "SOURCE",
        "EngineName": "s3",
        "EngineDisplayName": "Amazon S3",
        "ExtraConnectionAttributes": "bucketFolder=sourcedata;bucketName=my-corp-
data;compressionType=NONE;csvDelimiter=,;csvRowDelimiter=\\n;",
        "Status": "active",
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:GUVAFG34EECUOJ6QVZ56DAHT3U",
        "SslMode": "none",
        "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-role",
        "S3Settings": {
            "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-
role",
            "CsvRowDelimiter": "\\n",
            "CsvDelimiter": ",",
            "BucketFolder": "sourcedata",
            "BucketName": "my-corp-data",
            "CompressionType": "NONE",
            "EnableStatistics": true
        }
    }
}
```

如需詳細資訊，請參閱《 Database Migration Service AWS 使用者指南》中的使用 DMS 端點。
AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateEndpoint。

create-event-subscription

以下程式碼範例顯示如何使用 create-event-subscription。

AWS CLI

列出事件訂閱

下列create-event-subscription範例會建立 Amazon SNS 主題 (my-sns-topic) 的事件訂閱。

```
aws dms create-event-subscription \
    --subscription-name my-dms-events \
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:my-sns-topic
```

輸出：

```
{
    "EventSubscription": {
        "CustomerAwsId": "123456789012",
        "CustSubscriptionId": "my-dms-events",
        "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:my-sns-topic",
        "Status": "creating",
        "SubscriptionCreationTime": "2020-05-21 21:58:38.598",
        "Enabled": true
    }
}
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的使用事件和通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateEventSubscription。

create-replication-instance

以下程式碼範例顯示如何使用 create-replication-instance。

AWS CLI

建立複寫執行個體

下列create-replication-instance範例會建立複寫執行個體。

```
aws dms create-replication-instance \
    --replication-instance-identifier my-repl-instance \
    --replication-instance-class dms.t2.micro \
    --allocated-storage 5
```

輸出：

```
{
    "ReplicationInstance": {
        "ReplicationInstanceIdentifier": "my-repl-instance",
        "ReplicationInstanceClass": "dms.t2.micro",
        "ReplicationInstanceStatus": "creating",
        "AllocatedStorage": 5,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-f839b688",
                "Status": "active"
            }
        ],
        "ReplicationSubnetGroup": {
            "ReplicationSubnetGroupIdentifier": "default",
            "ReplicationSubnetGroupDescription": "default",
            "VpcId": "vpc-136a4c6a",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-da327bf6",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-42599426",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1d"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-bac383e0",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1c"
```

```
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-6746046b",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1f"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-d7c825e8",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1e"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-cbfff283",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "sat:12:35-sat:13:05",
        "PendingModifiedValues": {},
        "MultiAZ": false,
        "EngineVersion": "3.3.2",
        "AutoMinorVersionUpgrade": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/f7bc0f8e-1a3a-4ace-9faa-
e8494fa3921a",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:ZK2VQBUWFDBAWHIXHAYG5G2PKY",
        "PubliclyAccessible": true
    }
}
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的使用 AWS DMS 複寫執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateReplicationInstance。

create-replication-subnet-group

以下程式碼範例顯示如何使用 create-replication-subnet-group。

AWS CLI

建立子網路群組

下列create-replication-subnet-group範例會建立由 3 個子網路組成的群組。

```
aws dms create-replication-subnet-group \
    --replication-subnet-group-identifier my-subnet-group \
    --replication-subnet-group-description "my subnet group" \
    --subnet-ids subnet-da327bf6 subnet-bac383e0 subnet-d7c825e8
```

輸出：

```
{
    "ReplicationSubnetGroup": {
        "ReplicationSubnetGroupIdentifier": "my-subnet-group",
        "ReplicationSubnetGroupDescription": "my subnet group",
        "VpcId": "vpc-136a4c6a",
        "SubnetGroupStatus": "Complete",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-da327bf6",
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1a"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-bac383e0",
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1c"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-d7c825e8",
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1e"
                },
                "SubnetStatus": "Active"
```

```
            }
        ]
    }
}
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的設定複寫執行個體的網路。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateReplicationSubnetGroup。

create-replication-task

以下程式碼範例顯示如何使用 create-replication-task。

AWS CLI

建立複寫任務

下列create-replication-task範例會建立複寫任務。

```
aws dms create-replication-task \
    --replication-task-identifier movedata \
    --source-endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA \
    --target-endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U \
    --replication-instance-arn $RI_ARN \
    --migration-type full-load \
    --table-mappings file://table-mappings.json
```

table-mappings.json 的內容：

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "prodrep",
                "table-name": "%"
            },
            "rule-action": "include",
```

```
            "filters": []
        }
    ]
}
```

輸出：

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
        "TableMappings": ...output omitted... ,
        "ReplicationTaskSettings": ...output omitted... ,
        "Status": "creating",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
}
```

如需詳細資訊，請參閱 Database Migration Service AWS 使用者指南中的使用 DMS 任務。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateReplicationTask。

delete-connection

以下程式碼範例顯示如何使用 delete-connection。

AWS CLI

刪除連線

下列delete-connection範例會取消端點與複寫執行個體的關聯。

```
aws dms delete-connection \
    --endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA \
```

```
    --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
```

輸出：

```
{
    "Connection": {
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "Status": "deleting",
        "EndpointIdentifier": "src-database-1",
        "ReplicationInstanceIdentifier": "my-repl-instance"
    }
}
```

如需詳細資訊，請參閱《 AWS Database Migration Service 使用者指南》中的 https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Endpoints.Creating.html：//。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteConnection。

delete-endpoint

以下程式碼範例顯示如何使用 delete-endpoint。

AWS CLI

刪除端點

下列delete-endpoint範例會刪除端點。

```
aws dms delete-endpoint \
    --endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:OUJJVXO4XZ4CYTSEG5XGMN2R3Y
```

輸出：

```
{
    "Endpoint": {
        "EndpointIdentifier": "src-endpoint",
        "EndpointType": "SOURCE",
```

```
        "EngineName": "s3",
        "EngineDisplayName": "Amazon S3",
        "ExtraConnectionAttributes": "bucketFolder=sourcedata;bucketName=my-corp-
data;compressionType=NONE;csvDelimiter=,;csvRowDelimiter=\\n;",
        "Status": "deleting",
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:OUJJVXO4XZ4CYTSEG5XGMN2R3Y",
        "SslMode": "none",
        "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-role",
        "S3Settings": {
            "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-
role",
            "CsvRowDelimiter": "\\n",
            "CsvDelimiter": ",",
            "BucketFolder": "sourcedata",
            "BucketName": "my-corp-data",
            "CompressionType": "NONE",
            "EnableStatistics": true
        }
    }
}
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的使用 AWS DMS 端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteEndpoint。

delete-event-subscription

以下程式碼範例顯示如何使用 delete-event-subscription。

AWS CLI

刪除事件訂閱

下列delete-event-subscription範例會刪除 Amazon SNS 主題的訂閱。

```
aws dms delete-event-subscription \
    --subscription-name "my-dms-events"
```

輸出：

```
{
    "EventSubscription": {
```

```
        "CustomerAwsId": "123456789012",
        "CustSubscriptionId": "my-dms-events",
        "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:my-sns-topic",
        "Status": "deleting",
        "SubscriptionCreationTime": "2020-05-21 21:58:38.598",
        "Enabled": true
    }
}
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的使用事件和通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteEventSubscription。

delete-replication-instance

以下程式碼範例顯示如何使用 delete-replication-instance。

AWS CLI

刪除複寫執行個體

以下 delete-replication-instance 範例會刪除複寫執行個體。

```
aws dms delete-replication-instance \
    --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
```

輸出：

```
{
    "ReplicationInstance": {
        "ReplicationInstanceIdentifier": "my-repl-instance",
        "ReplicationInstanceClass": "dms.t2.micro",
        "ReplicationInstanceStatus": "deleting",
        "AllocatedStorage": 5,
        "InstanceCreateTime": 1590011235.952,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-f839b688",
                "Status": "active"
            }
        ],
        "AvailabilityZone": "us-east-1e",
```

```
        "ReplicationSubnetGroup": {
            "ReplicationSubnetGroupIdentifier": "default",
            "ReplicationSubnetGroupDescription": "default",
            "VpcId": "vpc-136a4c6a",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-da327bf6",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-42599426",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1d"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-bac383e0",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1c"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-6746046b",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1f"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-d7c825e8",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1e"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-cbfff283",
                    "SubnetAvailabilityZone": {
```

```
                        "Name": "us-east-1b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "wed:11:42-wed:12:12",
        "PendingModifiedValues": {},
        "MultiAZ": true,
        "EngineVersion": "3.3.2",
        "AutoMinorVersionUpgrade": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/f7bc0f8e-1a3a-4ace-9faa-
e8494fa3921a",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "ReplicationInstancePublicIpAddress": "54.225.120.92",
        "ReplicationInstancePrivateIpAddress": "172.31.30.121",
        "ReplicationInstancePublicIpAddresses": [
            "54.225.120.92",
            "3.230.18.248"
        ],
        "ReplicationInstancePrivateIpAddresses": [
            "172.31.30.121",
            "172.31.75.90"
        ],
        "PubliclyAccessible": true,
        "SecondaryAvailabilityZone": "us-east-1b"
    }
}
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的使用 AWS DMS 複寫執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteReplicationInstance。

delete-replication-subnet-group

以下程式碼範例顯示如何使用 delete-replication-subnet-group。

AWS CLI

刪除子網路群組

下列delete-replication-subnet-group範例會刪除子網路群組。

```
aws dms delete-replication-subnet-group \
--replication-subnet-group-identifier my-subnet-group
```

輸出：

```
(none)
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的設定複寫執行個體的網路。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteReplicationSubnetGroup。

delete-replication-task

以下程式碼範例顯示如何使用 delete-replication-task。

AWS CLI

刪除複寫任務

下列delete-replication-task範例會刪除複寫任務。

```
aws dms delete-replication-task \
    --replication-task-arn arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

輸出：

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
        "TableMappings": ...output omitted...,
        "ReplicationTaskSettings": ...output omitted...,
```

```
        "Status": "deleting",
        "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskStartDate": 1590789988.677,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
}
```

如需詳細資訊，請參閱《 Database Migration Service AWS 使用者指南》中的使用 DMS 任務。AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteReplicationTask。

describe-account-attributes

以下程式碼範例顯示如何使用 describe-account-attributes。

AWS CLI

描述帳戶屬性

下列describe-account-attributes範例列出您 AWS 帳戶的屬性。

```
aws dms describe-account-attributes
```

輸出：

```
{
    "AccountQuotas": [
        {
            "AccountQuotaName": "ReplicationInstances",
            "Used": 1,
            "Max": 20
        },
        {
            "AccountQuotaName": "AllocatedStorage",
            "Used": 5,
            "Max": 10000
        },

        ...remaining output omitted...
```

```
    ],
    "UniqueAccountIdentifier": "cqahfbfy5xee"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAccountAttributes。

describe-certificates

以下程式碼範例顯示如何使用 describe-certificates。

AWS CLI

列出可用的憑證

下列describe-certificates範例列出您 AWS 帳戶中可用的憑證。

```
aws dms describe-certificates
```

輸出：

```
{
    "Certificates": [
        {
            "CertificateIdentifier": "my-cert",
            "CertificateCreationDate": 1543259542.506,
            "CertificatePem": "-----BEGIN CERTIFICATE-----
\nMIID9DCCAtygAwIBAgIBQjANBgkqhkiG9w0BAQ ...U"

            ... remaining output omittted ...

        }
    ]
}
```

如需詳細資訊，請參閱《 AWS Database Migration Service 使用者指南》中的使用 SSL。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCertificates。

describe-connections

以下程式碼範例顯示如何使用 describe-connections。

AWS CLI

描述連線

下列describe-connections範例列出您已在複寫執行個體和端點之間測試的連線。

```
aws dms describe-connections
```

輸出：

```
{
    "Connections": [
        {
            "Status": "successful",
            "ReplicationInstanceIdentifier": "test",
            "EndpointArn": "arn:aws:dms:us-east-arn:aws:dms:us-
east-1:123456789012:endpoint:ZW5UAN6P4E77EC7YWHK4RZZ3BE",
            "EndpointIdentifier": "testsrc1",
            "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:6UTDJGBOUS3VI3SUWA66XFJCJQ"
        }
    ]
}
```

如需詳細資訊，請參閱《 AWS Database Migration Service 使用者指南》中的建立來源和目標端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeConnections。

describe-endpoint-types

以下程式碼範例顯示如何使用 describe-endpoint-types。

AWS CLI

列出可用的端點類型

下列describe-endpoint-types範例列出可用的 MySQL 端點類型。

```
aws dms describe-endpoint-types \
    --filters "Name=engine-name,Values=mysql"
```

輸出：

```
{
    "SupportedEndpointTypes": [
        {
            "EngineName": "mysql",
            "SupportsCDC": true,
            "EndpointType": "source",
            "EngineDisplayName": "MySQL"
        },
        {
            "EngineName": "mysql",
            "SupportsCDC": true,
            "EndpointType": "target",
            "EngineDisplayName": "MySQL"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的使用 AWS DMS 端點 <https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Endpoints.html>`__。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEndpointTypes。

describe-endpoints

以下程式碼範例顯示如何使用 describe-endpoints。

AWS CLI

描述端點

下列describe-endpoints範例列出您 AWS 帳戶中的端點。

```
aws dms describe-endpoints
```

輸出：

```
{
    "Endpoints": [
        {
            "Username": "dms",
```

```
            "Status": "active",
            "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:SF2WOFLWYWKVEOHID2EKLP3SJI",
            "ServerName": "ec2-52-32-48-61.us-west-2.compute.amazonaws.com",
            "EndpointType": "SOURCE",
            "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/94d5c4e7-4e4c-44be-
b58a-c8da7adf57cd",
            "DatabaseName": "test",
            "EngineName": "mysql",
            "EndpointIdentifier": "pri100",
            "Port": 8193
        },
        {
            "Username": "admin",
            "Status": "active",
            "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:TJJZCIH3CJ24TJRU4VC32WEWFR",
            "ServerName": "test.example.com",
            "EndpointType": "SOURCE",
            "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/2431021b-1cf2-
a2d4-77b2-59a9e4bce323",
            "DatabaseName": "EMPL",
            "EngineName": "oracle",
            "EndpointIdentifier": "test",
            "Port": 1521
        }
    ]
}
```

如需詳細資訊，請參閱《 Database Migration Service AWS 使用者指南》中的使用 DMS 端點。AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEndpoints。

describe-event-categories

以下程式碼範例顯示如何使用 describe-event-categories。

AWS CLI

描述事件類別

下列describe-event-categories範例列出可用的事件類別。

```
aws dms describe-event-categories
```

輸出：

```
{
    "EventCategoryGroupList": [
        {
            "SourceType": "replication-instance",
            "EventCategories": [
                "low storage",
                "configuration change",
                "maintenance",
                "deletion",
                "creation",
                "failover",
                "failure"
            ]
        },
        {
            "SourceType": "replication-task",
            "EventCategories": [
                "configuration change",
                "state change",
                "deletion",
                "creation",
                "failure"
            ]
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的使用事件和通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEventCategories。

describe-event-subscriptions

以下程式碼範例顯示如何使用 describe-event-subscriptions。

AWS CLI

描述事件訂閱

下列describe-event-subscriptions範例列出 Amazon SNS 主題的事件訂閱。

```
aws dms describe-event-subscriptions
```

輸出：

```
{
    "EventSubscriptionsList": [
        {
            "CustomerAwsId": "123456789012",
            "CustSubscriptionId": "my-dms-events",
            "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:my-sns-topic",
            "Status": "deleting",
            "SubscriptionCreationTime": "2020-05-21 22:28:51.924",
            "Enabled": true
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的使用事件和通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEventSubscriptions。

describe-events

以下程式碼範例顯示如何使用 describe-events。

AWS CLI

列出 DMS 事件

下列describe-events範例列出源自複寫執行個體的事件。

```
aws dms describe-events \
    --source-type "replication-instance"
```

輸出：

```
{
    "Events": [
        {
            "SourceIdentifier": "my-repl-instance",
```

```
            "SourceType": "replication-instance",
            "Message": "Replication application shutdown",
            "EventCategories": [],
            "Date": 1590771645.776
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的使用事件和通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEvents。

describe-orderable-replication-instances

以下程式碼範例顯示如何使用 describe-orderable-replication-instances。

AWS CLI

描述可排序的複寫執行個體

下列describe-orderable-replication-instances範例列出您可以訂購的複寫執行個體類型。

```
aws dms describe-orderable-replication-instances
```

輸出：

```
{
    "OrderableReplicationInstances": [
        {
            "EngineVersion": "3.3.2",
            "ReplicationInstanceClass": "dms.c4.2xlarge",
            "StorageType": "gp2",
            "MinAllocatedStorage": 5,
            "MaxAllocatedStorage": 6144,
            "DefaultAllocatedStorage": 100,
            "IncludedAllocatedStorage": 100,
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1c",
                "us-east-1d",
```

```
                "us-east-1e",
                "us-east-1f"
            ]
        },
        {
            "EngineVersion": "3.3.2",
            "ReplicationInstanceClass": "dms.c4.4xlarge",
            "StorageType": "gp2",
            "MinAllocatedStorage": 5,
            "MaxAllocatedStorage": 6144,
            "DefaultAllocatedStorage": 100,
            "IncludedAllocatedStorage": 100,
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1c",
                "us-east-1d",
                "us-east-1e",
                "us-east-1f"
            ]
        },

        ...remaining output omitted...

    }
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的使用 AWS DMS 複寫執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeOrderableReplicationInstances。

describe-refresh-schemas-status

以下程式碼範例顯示如何使用 describe-refresh-schemas-status。

AWS CLI

列出端點的重新整理狀態

下列describe-refresh-schemas-status範例會傳回先前重新整理請求的狀態。

```
aws dms describe-refresh-schemas-status \
```

```
    --endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA
```

輸出：

```
{
    "RefreshSchemasStatus": {
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "Status": "successful",
        "LastRefreshDate": 1590786544.605
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRefreshSchemasStatus。

describe-replication-instances

以下程式碼範例顯示如何使用 describe-replication-instances。

AWS CLI

描述複寫執行個體

下列describe-replication-instances範例列出您 AWS 帳戶中的複寫執行個體。

```
aws dms describe-replication-instances
```

輸出：

```
{
    "ReplicationInstances": [
        {
            "ReplicationInstanceIdentifier": "my-repl-instance",
            "ReplicationInstanceClass": "dms.t2.micro",
            "ReplicationInstanceStatus": "available",
            "AllocatedStorage": 5,
            "InstanceCreateTime": 1590011235.952,
            "VpcSecurityGroups": [
                {
```

```
                "VpcSecurityGroupId": "sg-f839b688",
                "Status": "active"
            }
        ],
        "AvailabilityZone": "us-east-1e",
        "ReplicationSubnetGroup": {
            "ReplicationSubnetGroupIdentifier": "default",
            "ReplicationSubnetGroupDescription": "default",
            "VpcId": "vpc-136a4c6a",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-da327bf6",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-42599426",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1d"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-bac383e0",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1c"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-6746046b",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1f"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-d7c825e8",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1e"
                    },
```

```
                        "SubnetStatus": "Active"
                    },
                    {
                        "SubnetIdentifier": "subnet-cbfff283",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1b"
                        },
                        "SubnetStatus": "Active"
                    }
                ]
            },
            "PreferredMaintenanceWindow": "wed:11:42-wed:12:12",
            "PendingModifiedValues": {
                "MultiAZ": true
            },
            "MultiAZ": false,
            "EngineVersion": "3.3.2",
            "AutoMinorVersionUpgrade": true,
            "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
f7bc0f8e-1a3a-4ace-9faa-e8494fa3921a",
            "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
            "ReplicationInstancePublicIpAddress": "3.230.18.248",
            "ReplicationInstancePrivateIpAddress": "172.31.75.90",
            "ReplicationInstancePublicIpAddresses": [
                "3.230.18.248"
            ],
            "ReplicationInstancePrivateIpAddresses": [
                "172.31.75.90"
            ],
            "PubliclyAccessible": true,
            "FreeUntil": 1590194829.267
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的使用 AWS DMS 複寫執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReplicationInstances。

describe-replication-subnet-groups

以下程式碼範例顯示如何使用 describe-replication-subnet-groups。

AWS CLI

顯示可用的子網路群組

下列describe-replication-subnet-groups範例列出可用的子網路群組。

```
aws dms describe-replication-subnet-groups \
    --filter "Name=replication-subnet-group-id,Values=my-subnet-group"
```

輸出：

```
{
    "ReplicationSubnetGroups": [
        {
            "ReplicationSubnetGroupIdentifier": "my-subnet-group",
            "ReplicationSubnetGroupDescription": "my subnet group",
            "VpcId": "vpc-136a4c6a",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-da327bf6",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-bac383e0",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1c"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-d7c825e8",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1e"
                    },
                    "SubnetStatus": "Active"
```

```
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的設定複寫執行個體的網路。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReplicationSubnetGroups。

describe-replication-task-assessment-results

以下程式碼範例顯示如何使用 describe-replication-task-assessment-results。

AWS CLI

列出複寫任務評估的結果

下列describe-replication-task-assessment-results範例列出先前任務評估的結果。

```
aws dms describe-replication-task-assessment-results
```

輸出：

```
{
    "ReplicationTaskAssessmentResults": [
        {
            "ReplicationTaskIdentifier": "moveit2",
            "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII",
            "ReplicationTaskLastAssessmentDate": 1590790230.0,
            "AssessmentStatus": "No issues found",
            "AssessmentResultsFile": "moveit2/2020-05-29-22-10"
        }
    ]
}
```

如需詳細資訊，請參閱《 AWS Database Migration Service 使用者指南》中的建立任務評估報告。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReplicationTaskAssessmentResults。

describe-replication-tasks

以下程式碼範例顯示如何使用 describe-replication-tasks。

AWS CLI

描述複寫任務

下列describe-replication-tasks範例說明目前的複寫任務。

```
aws dms describe-replication-tasks
```

輸出：

```
{
    "ReplicationTasks": [
        {
            "ReplicationTaskIdentifier": "moveit2",
            "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
            "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
            "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
            "MigrationType": "full-load",
            "TableMappings": ...output omitted... ,
            "ReplicationTaskSettings": ...output omitted... ,
            "Status": "stopped",
            "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
            "ReplicationTaskCreationDate": 1590524772.505,
            "ReplicationTaskStartDate": 1590619805.212,
            "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII",
            "ReplicationTaskStats": {
                "FullLoadProgressPercent": 100,
                "ElapsedTimeMillis": 0,
                "TablesLoaded": 0,
                "TablesLoading": 0,
                "TablesQueued": 0,
                "TablesErrored": 0,
                "FreshStartDate": 1590619811.528,
                "StartDate": 1590619811.528,
                "StopDate": 1590619842.068
            }
```

```
        }
    ]
}
```

如需詳細資訊，請參閱 Database Migration Service AWS 使用者指南中的使用 DMS 任務。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReplicationTasks。

describe-schemas

以下程式碼範例顯示如何使用 describe-schemas。

AWS CLI

描述資料庫結構描述

下列describe-schemas範例列出端點上可用的資料表。

```
aws dms describe-schemas \
    --endpoint-arn "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA"
```

輸出：

```
{
    "Schemas": [
        "prodrep"
    ]
}
```

如需詳細資訊，請參閱《 AWS Database Migration Service 使用者指南》中的這是主題標題。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSchemas。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出資源的標籤

下列list-tags-for-resource範例列出複寫執行個體的標籤。

```
aws dms list-tags-for-resource \
    --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
```

輸出：

```
{
    "TagList": [
        {
            "Key": "Project",
            "Value": "dbMigration"
        },
        {
            "Key": "Environment",
            "Value": "PROD"
        }
    ]
}
```

如需詳細資訊，請參閱《 AWS Database Migration Service 使用者指南》中的標記資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

modify-endpoint

以下程式碼範例顯示如何使用 modify-endpoint。

AWS CLI

修改端點

下列modify-endpoint範例會將額外的連線屬性新增至 端點。

```
aws dms modify-endpoint \
    --endpoint-arn "arn:aws:dms:us-
east-1:123456789012:endpoint:GUVAFG34EECUOJ6QVZ56DAHT3U" \
    --extra-connection-attributes "compressionType=GZIP"
```

輸出：

```
{
    "Endpoint": {
        "EndpointIdentifier": "src-endpoint",
```

```
        "EndpointType": "SOURCE",
        "EngineName": "s3",
        "EngineDisplayName": "Amazon S3",
        "ExtraConnectionAttributes":
 "compressionType=GZIP;csvDelimiter=,;csvRowDelimiter=\\n;",
        "Status": "active",
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:GUVAFG34EECUOJ6QVZ56DAHT3U",
        "SslMode": "none",
        "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-role",
        "S3Settings": {
            "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-
role",
            "CsvRowDelimiter": "\\n",
            "CsvDelimiter": ",",
            "BucketFolder": "",
            "BucketName": "",
            "CompressionType": "GZIP",
            "EnableStatistics": true
        }
    }
}
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的使用 AWS DMS 端點 <https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Endpoints.html>`__。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyEndpoint。

`modify-event-subscription`

以下程式碼範例顯示如何使用 `modify-event-subscription`。

AWS CLI

修改事件訂閱

下列`modify-event-subscription`範例會變更事件訂閱的來源類型。

```
aws dms modify-event-subscription \
    --subscription-name "my-dms-events" \
    --source-type replication-task
```

輸出：

```
{
    "EventSubscription": {
        "CustomerAwsId": "123456789012",
        "CustSubscriptionId": "my-dms-events",
        "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:my-sns-topic",
        "Status": "modifying",
        "SubscriptionCreationTime": "2020-05-29 17:04:40.262",
        "SourceType": "replication-task",
        "Enabled": true
    }
}
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的使用事件和通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyEventSubscription。

`modify-replication-instance`

以下程式碼範例顯示如何使用 `modify-replication-instance`。

AWS CLI

修改複寫執行個體

下列`modify-replication-instance`範例會修改複寫執行個體，使其使用異地同步備份部署。

```
aws dms modify-replication-instance \
     --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE \
     --multi-az
```

輸出：

```
{
    "ReplicationInstance": {
        "ReplicationInstanceIdentifier": "my-repl-instance",
        "ReplicationInstanceClass": "dms.t2.micro",
        "ReplicationInstanceStatus": "available",
        "AllocatedStorage": 5,
        "InstanceCreateTime": 1590011235.952,

        ...output omitted...
```

```
        "PendingModifiedValues": {
            "MultiAZ": true
        },
        "MultiAZ": false,
        "EngineVersion": "3.3.2",
        "AutoMinorVersionUpgrade": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/f7bc0f8e-1a3a-4ace-9faa-
e8494fa3921a",

        ...output omitted...

    }
}
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的使用 AWS DMS 複寫執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyReplicationInstance。

modify-replication-subnet-group

以下程式碼範例顯示如何使用 modify-replication-subnet-group。

AWS CLI

修改子網路群組

下列modify-replication-subnet-group範例會變更與子網路群組相關聯的子網路清單。

```
aws dms modify-replication-subnet-group \
    --replication-subnet-group-identifier my-subnet-group \
    --subnet-id subnet-da327bf6 subnet-bac383e0
```

輸出：

```
{
    "ReplicationSubnetGroup": {
        "ReplicationSubnetGroupIdentifier": "my-subnet-group",
        "ReplicationSubnetGroupDescription": "my subnet group",
        "VpcId": "vpc-136a4c6a",
        "SubnetGroupStatus": "Complete",
        "Subnets": [
```

```
            {
                "SubnetIdentifier": "subnet-da327bf6",
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1a"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-bac383e0",
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1c"
                },
                "SubnetStatus": "Active"
            }
        ]
    }
}
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的設定複寫執行個體的網路。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyReplicationSubnetGroup。

modify-replication-task

以下程式碼範例顯示如何使用 modify-replication-task。

AWS CLI

修改複寫任務

下列modify-replication-task範例會變更任務的資料表映射。

```
aws dms modify-replication-task \
    --replication-task-arn "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII" \
    --table-mappings file://table-mappings.json
```

table-mappings.json 的內容：

```
{
    "rules": [
        {
```

```
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "prodrep",
                "table-name": "ACCT_%"
            },
            "rule-action": "include",
            "filters": []
        }
    ]
}
```

輸出：

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
        "TableMappings": ...output omitted...,
        "ReplicationTaskSettings": ...output omitted...,
        "Status": "modifying",
        "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskStartDate": 1590789424.653,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
}
```

如需詳細資訊，請參閱 Database Migration Service AWS 使用者指南中的使用 DMS 任務。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyReplicationTask。

reboot-replication-instance

以下程式碼範例顯示如何使用 reboot-replication-instance。

AWS CLI

重新啟動複寫執行個體

以下 reboot-replication-instance 範例會重新開機複寫執行個體。

```
aws dms reboot-replication-instance \
    --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
```

輸出：

```
{
    "ReplicationInstance": {
        "ReplicationInstanceIdentifier": "my-repl-instance",
        "ReplicationInstanceClass": "dms.t2.micro",
        "ReplicationInstanceStatus": "rebooting",
        "AllocatedStorage": 5,
        "InstanceCreateTime": 1590011235.952,
    ... output omitted ...
    }
}
```

如需詳細資訊，請參閱 AWS Database Migration Service 使用者指南中的使用 AWS DMS 複寫執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RebootReplicationInstance。

refresh-schemas

以下程式碼範例顯示如何使用 refresh-schemas。

AWS CLI

重新整理資料庫結構描述

下列refresh-schemas範例請求 AWS DMS 重新整理端點的結構描述清單。

```
aws dms refresh-schemas \
    --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE \
```

```
    --endpoint-arn "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA"
```

輸出：

```
{
    "RefreshSchemasStatus": {
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "Status": "refreshing",
        "LastRefreshDate": 1590019949.103
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RefreshSchemas。

reload-tables

以下程式碼範例顯示如何使用 reload-tables。

AWS CLI

重新整理端點上可用的資料表清單

下列reload-tables範例會重新載入端點上可用資料表的清單。

```
aws dms reload-tables \
    --replication-task-arn "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII" \
    --tables-to-reload "SchemaName=prodrep,TableName=ACCT_BAL"
```

輸出：

```
{
    "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReloadTables。

remove-tags-from-resource

以下程式碼範例顯示如何使用 remove-tags-from-resource。

AWS CLI

從複寫執行個體移除標籤

下列remove-tags-from-resource範例會從複寫執行個體移除標籤。

```
aws dms remove-tags-from-resource \
    --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
 \
    --tag-keys Environment Project
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 AWS Database Migration Service 使用者指南》中的標記資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 RemoveTagsFromResource。

start-replication-task-assessment

以下程式碼範例顯示如何使用 start-replication-task-assessment。

AWS CLI

啟動任務評估

下列start-replication-task-assessment範例會啟動複寫任務評估。

```
aws dms start-replication-task-assessment \
    --replication-task-arn arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

輸出：

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
```

```
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
        "TableMappings": ...output omitted...,
        "ReplicationTaskSettings": ...output omitted...,
        "Status": "testing",
        "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskStartDate": 1590789988.677,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
}
```

如需詳細資訊，請參閱《 AWS Database Migration Service 使用者指南》中的建立任務評估報告。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartReplicationTaskAssessment。

start-replication-task

以下程式碼範例顯示如何使用 start-replication-task。

AWS CLI

啟動複寫任務

下列command-name範例列出您 AWS 帳戶中可用的小工具。

```
aws dms start-replication-task \
    --replication-task-arn arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII \
    --start-replication-task-type reload-target
```

輸出：

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
```

```
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
        "TableMappings": ...output omitted... ,
        "ReplicationTaskSettings": ...output omitted... ,
        "Status": "starting",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskStartDate": 1590619805.212,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
}
```

如需詳細資訊，請參閱《 Database Migration Service AWS 使用者指南》中的使用 DMS 任務。AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartReplicationTask。

stop-replication-task

以下程式碼範例顯示如何使用 stop-replication-task。

AWS CLI

停止任務

下列stop-replication-task範例會停止任務。

```
aws dms stop-replication-task \
    --replication-task-arn arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

輸出：

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
```

```
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
        "TableMappings": ...output omitted...,
        "ReplicationTaskSettings": ...output omitted...,
        "Status": "stopping",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskStartDate": 1590789424.653,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
}
```

如需詳細資訊，請參閱 Database Migration Service AWS 使用者指南中的使用 DMS 任務。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopReplicationTask。

test-connection

以下程式碼範例顯示如何使用 test-connection。

AWS CLI

測試與端點的連線

下列test-connection範例會測試是否可以從複寫執行個體存取端點。

```
aws dms test-connection \
    --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE \
    --endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA
```

輸出：

```
{
    "Connection": {
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "Status": "testing",
        "EndpointIdentifier": "src-database-1",
```

```
        "ReplicationInstanceIdentifier": "my-repl-instance"
    }
}
```

如需詳細資訊，請參閱《 AWS Database Migration Service 使用者指南》中的建立來源和目標端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TestConnection。

使用 的 Amazon DocumentDB 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon DocumentDB 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`add-tags-to-resource`

以下程式碼範例顯示如何使用 `add-tags-to-resource`。

AWS CLI

將一或多個標籤新增至指定的資源

下列`add-tags-to-resource`範例會將三個標籤新增至 `sample-cluster`。一個標籤 (CropB) 具有金鑰名稱，但沒有值。

```
aws docdb add-tags-to-resource \
    --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster \
    --tags Key="CropA",Value="Apple" Key="CropB" Key="CropC",Value="Corn"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的標記 Amazon DocumentDB 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AddTagsToResource。

apply-pending-maintenance-action

以下程式碼範例顯示如何使用 apply-pending-maintenance-action。

AWS CLI

讓待定的維護動作在下一個維護時段進行

下列apply-pending-maintenance-action範例會在下一個排定的維護時段執行所有系統更新動作。

```
aws docdb apply-pending-maintenance-action \
--resource-identifier arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster \
--apply-action system-update \
--opt-in-type next-maintenance
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的套用 Amazon DocumentDB 更新。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ApplyPendingMaintenanceAction。

copy-db-cluster-parameter-group

以下程式碼範例顯示如何使用 copy-db-cluster-parameter-group。

AWS CLI

複製現有的資料庫叢集參數群組

下列copy-db-cluster-parameter-group範例會複製custom-docdb3-6名為 的參數群組custom-docdb3-6-copy。複製時，它會將標籤新增至新的參數群組。

```
aws docdb copy-db-cluster-parameter-group \
    --source-db-cluster-parameter-group-identifier custom-docdb3-6 \
    --target-db-cluster-parameter-group-identifier custom-docdb3-6-copy \
    --target-db-cluster-parameter-group-description "Copy of custom-docdb3-6" \
```

```
    --tags Key="CopyNumber",Value="1" Key="Modifiable",Value="Yes"
```

輸出：

```
{
    "DBClusterParameterGroup": {
        "DBParameterGroupFamily": "docdb3.6",
        "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:12345678901:cluster-
pg:custom-docdb3-6-copy",
        "DBClusterParameterGroupName": "custom-docdb3-6-copy",
        "Description": "Copy of custom-docdb3-6"
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的複製 Amazon DocumentDB 叢集參數群組。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CopyDbClusterParameterGroup。

copy-db-cluster-snapshot

以下程式碼範例顯示如何使用 copy-db-cluster-snapshot。

AWS CLI

建立快照的副本

以下 copy-db-cluster-snapshot 範例會複製 sample-cluster-snapshot，並將複本命名為 sample-cluster-snapshot-copy。複本具有原始 的所有標籤，加上金鑰名稱為 的新標籤CopyNumber。

```
aws docdb copy-db-cluster-snapshot \
    --source-db-cluster-snapshot-identifier sample-cluster-snapshot \
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy \
    --copy-tags \
    --tags Key="CopyNumber",Value="1"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的複製叢集快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CopyDbClusterSnapshot。

create-db-cluster-parameter-group

以下程式碼範例顯示如何使用 create-db-cluster-parameter-group。

AWS CLI

建立 Amazon DocumentDB 叢集參數群組

下列create-db-cluster-parameter-group範例sample-parameter-group會使用 docdb3.6 系列建立資料庫叢集參數群組。

```
aws docdb create-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group \
    --db-parameter-group-family docdb3.6 \
    --description "Sample parameter group based on docdb3.6"
```

輸出：

```
{
    "DBClusterParameterGroup": {
        "Description": "Sample parameter group based on docdb3.6",
        "DBParameterGroupFamily": "docdb3.6",
        "DBClusterParameterGroupArn": "arn:aws:rds:us-west-2:123456789012:cluster-
pg:sample-parameter-group",
        "DBClusterParameterGroupName": "sample-parameter-group"
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的建立 Amazon DocumentDB 叢集參數群組。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDbClusterParameterGroup。

create-db-cluster-snapshot

以下程式碼範例顯示如何使用 create-db-cluster-snapshot。

AWS CLI

建立手動 Amazon DocumentDB 叢集快照

下列create-db-cluster-snapshot範例會建立名為 sample-cluster-snapshot 的 Amazon 資料庫叢集快照。

```
aws docdb create-db-cluster-snapshot \
   --db-cluster-identifier sample-cluster \
   --db-cluster-snapshot-identifier sample-cluster-snapshot
```

輸出：

```
{
    "DBClusterSnapshot": {
        "MasterUsername": "master-user",
        "SnapshotCreateTime": "2019-03-18T18:27:14.794Z",
        "AvailabilityZones": [
            "us-west-2a",
            "us-west-2b",
            "us-west-2c",
            "us-west-2d",
            "us-west-2e",
            "us-west-2f"
        ],
        "SnapshotType": "manual",
        "DBClusterSnapshotArn": "arn:aws:rds:us-west-2:123456789012:cluster-
snapshot:sample-cluster-snapshot",
        "EngineVersion": "3.6.0",
        "PercentProgress": 0,
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
        "Engine": "docdb",
        "DBClusterIdentifier": "sample-cluster",
        "Status": "creating",
        "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
        "Port": 0,
        "StorageEncrypted": false,
        "VpcId": "vpc-91280df6"
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的建立手動叢集快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDbClusterSnapshot。

create-db-cluster

以下程式碼範例顯示如何使用 create-db-cluster。

AWS CLI

建立 Amazon DocumentDB 叢集

下列create-db-cluster範例會在週日 20：30 到 11：00 之間建立名為 sample-cluster且具有偏好維護時段的 Amazon DocumentDB 叢集。

```
aws docdb create-db-cluster \
    --db-cluster-identifier sample-cluster \
    --engine docdb \
    --master-username master-user \
    --master-user-password password \
    --preferred-maintenance-window Sun:20:30-Sun:21:00
```

輸出：

```
{
    "DBCluster": {
        "DBClusterParameterGroup": "default.docdb3.6",
        "AssociatedRoles": [],
        "DBSubnetGroup": "default",
        "ClusterCreateTime": "2019-03-18T18:06:34.616Z",
        "Status": "creating",
        "Port": 27017,
        "PreferredMaintenanceWindow": "sun:20:30-sun:21:00",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "DBClusterMembers": [],
        "Engine": "docdb",
        "DBClusterIdentifier": "sample-cluster",
        "PreferredBackupWindow": "10:12-10:42",
        "AvailabilityZones": [
            "us-west-2d",
            "us-west-2f",
            "us-west-2e"
        ],
        "MasterUsername": "master-user",
        "BackupRetentionPeriod": 1,
        "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
```

```
            }
        ],
        "StorageEncrypted": false,
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
        "DbClusterResourceId": "cluster-L3R4YRSBUYDP4GLMTJ2WF5GH5Q",
        "MultiAZ": false,
        "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "EngineVersion": "3.6.0"
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的建立 Amazon DocumentDB 叢集。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDbCluster。

`create-db-instance`

以下程式碼範例顯示如何使用 `create-db-instance`。

AWS CLI

建立 Amazon DocumentDB 叢集執行個體

下列`create-db-instance`範例程式碼會在 Amazon DocumentDB 叢集 `sample-cluster-instance-2`中建立執行個體`sample-cluster`。

```
aws docdb create-db-instance \
    --db-cluster-identifier sample-cluster \
    --db-instance-class db.r4.xlarge \
    --db-instance-identifier sample-cluster-instance-2 \
    --engine docdb
```

輸出：

```
{
    "DBInstance": {
        "DBInstanceStatus": "creating",
        "PendingModifiedValues": {
            "PendingCloudwatchLogsExports": {
                "LogTypesToEnable": [
                    "audit"
```

```
                ]
            }
        },
        "PubliclyAccessible": false,
        "PreferredBackupWindow": "00:00-00:30",
        "PromotionTier": 1,
        "EngineVersion": "3.6.0",
        "BackupRetentionPeriod": 3,
        "DBInstanceIdentifier": "sample-cluster-instance-2",
        "PreferredMaintenanceWindow": "tue:10:28-tue:10:58",
        "StorageEncrypted": false,
        "Engine": "docdb",
        "DBClusterIdentifier": "sample-cluster",
        "DBSubnetGroup": {
            "Subnets": [
                {
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active",
                    "SubnetIdentifier": "subnet-4e26d263"
                },
                {
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    },
                    "SubnetStatus": "Active",
                    "SubnetIdentifier": "subnet-afc329f4"
                },
                {
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    },
                    "SubnetStatus": "Active",
                    "SubnetIdentifier": "subnet-53ab3636"
                },
                {
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetStatus": "Active",
                    "SubnetIdentifier": "subnet-991cb8d0"
                }
            ],
```

```
            "DBSubnetGroupDescription": "default",
            "SubnetGroupStatus": "Complete",
            "VpcId": "vpc-91280df6",
            "DBSubnetGroupName": "default"
        },
        "DBInstanceClass": "db.r4.xlarge",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster-
instance-2",
        "DbiResourceId": "db-XEKJLEMGRV5ZKCARUVA4HO3ITE"
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的將 Amazon DocumentDB 執行個體新增至叢集。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDbInstance。

create-db-subnet-group

以下程式碼範例顯示如何使用 create-db-subnet-group。

AWS CLI

建立 Amazon DocumentDB 子網路群組

下列create-db-subnet-group範例會建立名為 的 Amazon DocumentDB 子網路群組sample-subnet-group。

```
aws docdb create-db-subnet-group \
    --db-subnet-group-description "a sample subnet group" \
    --db-subnet-group-name sample-subnet-group \
    --subnet-ids "subnet-29ab1025" "subnet-991cb8d0" "subnet-53ab3636"
```

輸出：

```
{
```

```
    "DBSubnetGroup": {
        "SubnetGroupStatus": "Complete",
        "DBSubnetGroupName": "sample-subnet-group",
        "DBSubnetGroupDescription": "a sample subnet group",
        "VpcId": "vpc-91280df6",
        "DBSubnetGroupArn": "arn:aws:rds:us-west-2:123456789012:subgrp:sample-
subnet-group",
        "Subnets": [
            {
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-53ab3636",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2d"
                }
            },
            {
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-991cb8d0",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                }
            },
            {
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-29ab1025",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2c"
                }
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的建立 Amazon DocumentDB 子網路群組。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDbSubnetGroup。

delete-db-cluster-parameter-group

以下程式碼範例顯示如何使用 delete-db-cluster-parameter-group。

AWS CLI

刪除 Amazon DocumentDB 叢集參數群組

下列delete-db-cluster-parameter-group範例會刪除 Amazon DocumentDB 參數群組 sample-parameter-group。

```
aws docdb delete-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的刪除 Amazon DocumentDB 叢集參數群組。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDbClusterParameterGroup。

delete-db-cluster-snapshot

以下程式碼範例顯示如何使用 delete-db-cluster-snapshot。

AWS CLI

刪除 Amazon DocumentDB 叢集快照

下列delete-db-cluster-snapshot範例會刪除 Amazon DocumentDB 叢集快照 sample-cluster-snapshot。

```
aws docdb delete-db-cluster-snapshot \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

輸出：

```
{
    "DBClusterSnapshot": {
        "DBClusterIdentifier": "sample-cluster",
        "AvailabilityZones": [
            "us-west-2a",
            "us-west-2b",
            "us-west-2c",
            "us-west-2d"
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
```

```
        "VpcId": "vpc-91280df6",
        "DBClusterSnapshotArn": "arn:aws:rds:us-west-2:123456789012:cluster-
snapshot:sample-cluster-snapshot",
        "EngineVersion": "3.6.0",
        "Engine": "docdb",
        "SnapshotCreateTime": "2019-03-18T18:27:14.794Z",
        "Status": "available",
        "MasterUsername": "master-user",
        "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
        "PercentProgress": 100,
        "StorageEncrypted": false,
        "SnapshotType": "manual",
        "Port": 0
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的刪除叢集快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDbClusterSnapshot。

delete-db-cluster

以下程式碼範例顯示如何使用 delete-db-cluster。

AWS CLI

刪除 Amazon DocumentDB 叢集

下列delete-db-cluster範例會刪除 Amazon DocumentDB 叢集 sample-cluster。在刪除叢集之前，不會進行任何備份。注意：您必須先刪除與叢集相關聯的所有執行個體，才能將其刪除。

```
aws docdb delete-db-cluster \
    --db-cluster-identifier sample-cluster \
    --skip-final-snapshot
```

輸出：

```
{
    "DBCluster": {
        "DBClusterIdentifier": "sample-cluster",
        "DBSubnetGroup": "default",
        "EngineVersion": "3.6.0",
        "Engine": "docdb",
```

```
        "LatestRestorableTime": "2019-03-18T18:07:24.610Z",
        "PreferredMaintenanceWindow": "sun:20:30-sun:21:00",
        "StorageEncrypted": false,
        "EarliestRestorableTime": "2019-03-18T18:07:24.610Z",
        "Port": 27017,
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "MultiAZ": false,
        "MasterUsername": "master-user",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
        "Status": "available",
        "PreferredBackupWindow": "10:12-10:42",
        "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "AvailabilityZones": [
            "us-west-2c",
            "us-west-2b",
            "us-west-2a"
        ],
        "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "DbClusterResourceId": "cluster-L3R4YRSBUYDP4GLMTJ2WF5GH5Q",
        "ClusterCreateTime": "2019-03-18T18:06:34.616Z",
        "AssociatedRoles": [],
        "DBClusterParameterGroup": "default.docdb3.6",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "BackupRetentionPeriod": 1,
        "DBClusterMembers": []
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的刪除 Amazon DocumentDB 叢集。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDbCluster。

delete-db-instance

以下程式碼範例顯示如何使用 delete-db-instance。

AWS CLI

刪除 Amazon DocumentDB 執行個體

下列delete-db-instance範例會刪除 Amazon DocumentDB 執行個體 sample-cluster-instance-2。

```
aws docdb delete-db-instance \
    --db-instance-identifier sample-cluster-instance-2
```

輸出：

```
{
    "DBInstance": {
        "DBSubnetGroup": {
            "Subnets": [
                {
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active",
                    "SubnetIdentifier": "subnet-4e26d263"
                },
                {
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    },
                    "SubnetStatus": "Active",
                    "SubnetIdentifier": "subnet-afc329f4"
                },
                {
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    },
                    "SubnetStatus": "Active",
                    "SubnetIdentifier": "subnet-53ab3636"
                },
                {
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetStatus": "Active",
                    "SubnetIdentifier": "subnet-991cb8d0"
```

```
                }
            ],
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-91280df6",
            "SubnetGroupStatus": "Complete"
        },
        "PreferredBackupWindow": "00:00-00:30",
        "InstanceCreateTime": "2019-03-18T18:37:33.709Z",
        "DBInstanceClass": "db.r4.xlarge",
        "DbiResourceId": "db-XEKJLEMGRV5ZKCARUVA4HO3ITE",
        "BackupRetentionPeriod": 3,
        "Engine": "docdb",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "AutoMinorVersionUpgrade": true,
        "PromotionTier": 1,
        "EngineVersion": "3.6.0",
        "Endpoint": {
            "Address": "sample-cluster-instance-2.corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
            "HostedZoneId": "ZNKXH85TT8WVW",
            "Port": 27017
        },
        "DBInstanceIdentifier": "sample-cluster-instance-2",
        "PreferredMaintenanceWindow": "tue:10:28-tue:10:58",
        "EnabledCloudwatchLogsExports": [
            "audit"
        ],
        "PendingModifiedValues": {},
        "DBInstanceStatus": "deleting",
        "PubliclyAccessible": false,
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster-
instance-2",
        "DBClusterIdentifier": "sample-cluster",
        "AvailabilityZone": "us-west-2c",
        "StorageEncrypted": false
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的刪除 Amazon DocumentDB 執行個體。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDbInstance。

delete-db-subnet-group

以下程式碼範例顯示如何使用 delete-db-subnet-group。

AWS CLI

刪除 Amazon DocumentDB 子網路群組

下列delete-db-subnet-group範例會刪除 Amazon DocumentDB 子網路群組 sample-subnet-group。

```
aws docdb delete-db-subnet-group \
    --db-subnet-group-name sample-subnet-group
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的刪除 Amazon DocumentDB 子網路群組。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDbSubnetGroup。

describe-db-cluster-parameter-groups

以下程式碼範例顯示如何使用 describe-db-cluster-parameter-groups。

AWS CLI

查看一或多個 Amazon DocumentDB 叢集參數群組的詳細資訊

下列describe-db-cluster-parameter-groups範例顯示 Amazon DocumentDB 叢集參數群組 的詳細資訊custom3-6-param-grp。

```
aws docdb describe-db-cluster-parameter-groups \
    --db-cluster-parameter-group-name custom3-6-param-grp
```

輸出：

```
{
    "DBClusterParameterGroups": [
        {
            "DBParameterGroupFamily": "docdb3.6",
            "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:custom3-6-param-grp",
            "Description": "Custom docdb3.6 parameter group",
            "DBClusterParameterGroupName": "custom3-6-param-grp"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的檢視 Amazon DocumentDB 叢集參數群組。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbClusterParameterGroups。

`describe-db-cluster-parameters`

以下程式碼範例顯示如何使用 `describe-db-cluster-parameters`。

AWS CLI

檢視 Amazon DocumentDB 叢集參數群組的詳細參數清單。

下列`describe-db-cluster-parameters`範例列出 Amazon DocumentDB 參數群組 custom3-6-param-grp 的參數。

```
aws docdb describe-db-cluster-parameters \
     --db-cluster-parameter-group-name custom3-6-param-grp
```

輸出：

```
{
    "Parameters": [
        {
            "DataType": "string",
            "ParameterName": "audit_logs",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot",
            "Source": "system",
```

```
            "ApplyType": "dynamic",
            "AllowedValues": "enabled,disabled",
            "Description": "Enables auditing on cluster.",
            "ParameterValue": "disabled"
        },
        {
            "DataType": "string",
            "ParameterName": "tls",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot",
            "Source": "system",
            "ApplyType": "static",
            "AllowedValues": "disabled,enabled",
            "Description": "Config to enable/disable TLS",
            "ParameterValue": "enabled"
        },
        {
            "DataType": "string",
            "ParameterName": "ttl_monitor",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot",
            "Source": "user",
            "ApplyType": "dynamic",
            "AllowedValues": "disabled,enabled",
            "Description": "Enables TTL Monitoring",
            "ParameterValue": "enabled"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的檢視 Amazon DocumentDB 叢集參數。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbClusterParameters。

`describe-db-cluster-snapshot-attributes`

以下程式碼範例顯示如何使用 `describe-db-cluster-snapshot-attributes`。

AWS CLI

列出 Amazon DocumentDB 快照屬性名稱和值

下列describe-db-cluster-snapshot-attributes範例列出 Amazon DocumentDB 快照 的屬性名稱和值sample-cluster-snapshot。

```
aws docdb describe-db-cluster-snapshot-attributes \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

輸出：

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": []
            }
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的DescribeDBClusterSnapshotAttributes。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的DescribeDbClusterSnapshotAttributes。

describe-db-cluster-snapshots

以下程式碼範例顯示如何使用 describe-db-cluster-snapshots。

AWS CLI

描述 Amazon DocumentDB 快照

下列describe-db-cluster-snapshots範例顯示 Amazon DocumentDB 快照 的詳細資訊sample-cluster-snapshot。

```
aws docdb describe-db-cluster-snapshots \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

輸出：

```
{
    "DBClusterSnapshots": [
        {
            "AvailabilityZones": [
                "us-west-2a",
                "us-west-2b",
                "us-west-2c",
                "us-west-2d"
            ],
            "Status": "available",
            "DBClusterSnapshotArn": "arn:aws:rds:us-west-2:123456789012:cluster-
snapshot:sample-cluster-snapshot",
            "SnapshotCreateTime": "2019-03-15T20:41:26.515Z",
            "SnapshotType": "manual",
            "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
            "DBClusterIdentifier": "sample-cluster",
            "MasterUsername": "master-user",
            "StorageEncrypted": false,
            "VpcId": "vpc-91280df6",
            "EngineVersion": "3.6.0",
            "PercentProgress": 100,
            "Port": 0,
            "Engine": "docdb",
            "ClusterCreateTime": "2019-03-15T20:29:58.836Z"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的 DescribeDBClusterSnapshots。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbClusterSnapshots。

describe-db-clusters

以下程式碼範例顯示如何使用 `describe-db-clusters`。

AWS CLI

取得一或多個 Amazon DocumentDB 叢集的詳細資訊。

下列`describe-db-clusters`範例顯示 Amazon DocumentDB 叢集 的詳細資訊`sample-cluster`。透過省略 `--db-cluster-identifier` 參數，您最多可以取得 100 個叢集的資訊。

```
aws docdb describe-db-clusters
    --db-cluster-identifier sample-cluster
```

輸出：

```
{
    "DBClusters": [
        {
            "DBClusterParameterGroup": "default.docdb3.6",
            "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
            "PreferredBackupWindow": "00:00-00:30",
            "DBClusterIdentifier": "sample-cluster",
            "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
            "LatestRestorableTime": "2019-03-18T20:28:03.239Z",
            "MasterUsername": "master-user",
            "DBClusterMembers": [
                {
                    "PromotionTier": 1,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "IsClusterWriter": false,
                    "DBInstanceIdentifier": "sample-cluster"
                },
                {
                    "PromotionTier": 1,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "IsClusterWriter": true,
                    "DBInstanceIdentifier": "sample-cluster2"
                }
            ],
            "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
            "VpcSecurityGroups": [
                {
                    "VpcSecurityGroupId": "sg-77186e0d",
                    "Status": "active"
                }
            ],
            "Engine": "docdb",
            "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
            "DBSubnetGroup": "default",
            "MultiAZ": true,
            "AvailabilityZones": [
```

```
                "us-west-2a",
                "us-west-2c",
                "us-west-2b"
            ],
            "EarliestRestorableTime": "2019-03-15T20:30:47.020Z",
            "DbClusterResourceId": "cluster-UP4EF2PVDDFVHHDJQTYDAIGHLE",
            "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-
cluster",
            "BackupRetentionPeriod": 3,
            "HostedZoneId": "ZNKXH85TT8WVW",
            "StorageEncrypted": false,
            "EnabledCloudwatchLogsExports": [
                "audit"
            ],
            "AssociatedRoles": [],
            "EngineVersion": "3.6.0",
            "Port": 27017,
            "Status": "available"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的描述 Amazon DocumentDB 叢集。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbClusters。

describe-db-engine-versions

以下程式碼範例顯示如何使用 describe-db-engine-versions。

AWS CLI

列出可用的 Amazon DocumentDB 引擎版本

下列describe-db-engine-versions範例列出所有可用的 Amazon DocumentDB 引擎版本。

```
aws docdb describe-db-engine-versions \
    --engine docdb
```

輸出：

```
{
```

```
    "DBEngineVersions": [
        {
            "DBEngineVersionDescription": "DocDB version 1.0.200837",
            "DBParameterGroupFamily": "docdb3.6",
            "EngineVersion": "3.6.0",
            "ValidUpgradeTarget": [],
            "DBEngineDescription": "Amazon DocumentDB (with MongoDB compatibility)",
            "SupportsLogExportsToCloudwatchLogs": true,
            "Engine": "docdb",
            "ExportableLogTypes": [
                "audit"
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的 DescribeDBEngineVersions。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbEngineVersions。

describe-db-instances

以下程式碼範例顯示如何使用 describe-db-instances。

AWS CLI

尋找佈建 Amazon DocumentDB 執行個體的相關資訊

下列describe-db-instances範例顯示 Amazon DocumentDB 執行個體 的詳細資訊sample-cluster-instance。省略 --db-instance-identifier 參數，即可取得最多 100 個執行個體的資訊。

```
aws docdb describe-db-instances \
    --db-instance-identifier sample-cluster-instance
```

輸出：

```
{
    "DBInstances": [
        {
            "Endpoint": {
                "HostedZoneId": "ZNKXH85TT8WVW",
```

```
                "Address": "sample-cluster-instance.corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
                "Port": 27017
            },
            "PreferredBackupWindow": "00:00-00:30",
            "DBInstanceStatus": "available",
            "DBInstanceClass": "db.r4.large",
            "EnabledCloudwatchLogsExports": [
                "audit"
            ],
            "DBInstanceIdentifier": "sample-cluster-instance",
            "DBSubnetGroup": {
                "Subnets": [
                    {
                        "SubnetStatus": "Active",
                        "SubnetIdentifier": "subnet-4e26d263",
                        "SubnetAvailabilityZone": {
                            "Name": "us-west-2a"
                        }
                    },
                    {
                        "SubnetStatus": "Active",
                        "SubnetIdentifier": "subnet-afc329f4",
                        "SubnetAvailabilityZone": {
                            "Name": "us-west-2c"
                        }
                    },
                    {
                        "SubnetStatus": "Active",
                        "SubnetIdentifier": "subnet-53ab3636",
                        "SubnetAvailabilityZone": {
                            "Name": "us-west-2d"
                        }
                    },
                    {
                        "SubnetStatus": "Active",
                        "SubnetIdentifier": "subnet-991cb8d0",
                        "SubnetAvailabilityZone": {
                            "Name": "us-west-2b"
                        }
                    }
                ],
                "DBSubnetGroupName": "default",
                "SubnetGroupStatus": "Complete",
```

```
                "DBSubnetGroupDescription": "default",
                "VpcId": "vpc-91280df6"
            },
            "InstanceCreateTime": "2019-03-15T20:36:06.338Z",
            "Engine": "docdb",
            "StorageEncrypted": false,
            "AutoMinorVersionUpgrade": true,
            "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster-
instance",
            "PreferredMaintenanceWindow": "tue:08:39-tue:09:09",
            "VpcSecurityGroups": [
                {
                    "Status": "active",
                    "VpcSecurityGroupId": "sg-77186e0d"
                }
            ],
            "DBClusterIdentifier": "sample-cluster",
            "PendingModifiedValues": {},
            "BackupRetentionPeriod": 3,
            "PubliclyAccessible": false,
            "EngineVersion": "3.6.0",
            "PromotionTier": 1,
            "AvailabilityZone": "us-west-2c",
            "DbiResourceId": "db-A2GIKUV6KPOHITGGKI2NHVISZA"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的描述 Amazon DocumentDB 執行個體。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbInstances。

`describe-db-subnet-groups`

以下程式碼範例顯示如何使用 `describe-db-subnet-groups`。

AWS CLI

擷取 Amazon DocumentDB 子網路描述的清單

下列`describe-db-subnet-groups`範例說明名為 之 Amazon DocumentDB 子網路的詳細資訊`default`。

```
aws docdb describe-db-subnet-groups \
    --db-subnet-group-name default
```

輸出：

```
{
    "DBSubnetGroups": [
        {
            "VpcId": "vpc-91280df6",
            "DBSubnetGroupArn": "arn:aws:rds:us-west-2:123456789012:subgrp:default",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-4e26d263",
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    }
                },
                {
                    "SubnetIdentifier": "subnet-afc329f4",
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    }
                },
                {
                    "SubnetIdentifier": "subnet-53ab3636",
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    }
                },
                {
                    "SubnetIdentifier": "subnet-991cb8d0",
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    }
                }
            ],
            "DBSubnetGroupName": "default",
            "SubnetGroupStatus": "Complete",
            "DBSubnetGroupDescription": "default"
```

```
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的描述子網路群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbSubnetGroups。

`describe-engine-default-cluster-parameters`

以下程式碼範例顯示如何使用 `describe-engine-default-cluster-parameters`。

AWS CLI

描述 Amazon DocumentDB 的預設引擎和系統參數資訊

下列`describe-engine-default-cluster-parameters`範例顯示 Amazon DocumentDB 參數群組 的預設引擎和系統參數資訊的詳細資訊`docdb3.6`。

```
aws docdb describe-engine-default-cluster-parameters \
    --db-parameter-group-family docdb3.6
```

輸出：

```
{
    "EngineDefaults": {
        "DBParameterGroupFamily": "docdb3.6",
        "Parameters": [
            {
                "ApplyType": "dynamic",
                "ParameterValue": "disabled",
                "Description": "Enables auditing on cluster.",
                "Source": "system",
                "DataType": "string",
                "MinimumEngineVersion": "3.6.0",
                "AllowedValues": "enabled,disabled",
                "ParameterName": "audit_logs",
                "IsModifiable": true
            },
            {
                "ApplyType": "static",
                "ParameterValue": "enabled",
                "Description": "Config to enable/disable TLS",
```

```
                "Source": "system",
                "DataType": "string",
                "MinimumEngineVersion": "3.6.0",
                "AllowedValues": "disabled,enabled",
                "ParameterName": "tls",
                "IsModifiable": true
            },
            {
                "ApplyType": "dynamic",
                "ParameterValue": "enabled",
                "Description": "Enables TTL Monitoring",
                "Source": "system",
                "DataType": "string",
                "MinimumEngineVersion": "3.6.0",
                "AllowedValues": "disabled,enabled",
                "ParameterName": "ttl_monitor",
                "IsModifiable": true
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的 DescribeEngineDefaultClusterParameters。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEngineDefaultClusterParameters。

describe-event-categories

以下程式碼範例顯示如何使用 `describe-event-categories`。

AWS CLI

描述所有 Amazon DocumentDB 事件類別

下列`describe-event-categories`範例列出 Amazon DocumentDB 事件來源類型 的所有類別`db-instance`。

```
aws docdb describe-event-categories \
    --source-type db-cluster
```

輸出：

```
{
    "EventCategoriesMapList": [
        {
            "SourceType": "db-cluster",
            "EventCategories": [
                "failover",
                "maintenance",
                "notification",
                "failure"
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的檢視事件類別。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEventCategories。

`describe-events`

以下程式碼範例顯示如何使用 `describe-events`。

AWS CLI

列出 Amazon DocumentDB 事件

下列`describe-events`範例列出過去 24 小時 (1440 分鐘） 的所有 Amazon DocumentDB 事件。

```
aws docdb describe-events \
    --duration 1440
```

此命令不會產生輸出。輸出：

```
{
    "Events": [
        {
            "EventCategories": [
                "failover"
            ],
            "Message": "Started cross AZ failover to DB instance: sample-cluster",
            "Date": "2019-03-18T21:36:29.807Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-
cluster",
```

```
            "SourceIdentifier": "sample-cluster",
            "SourceType": "db-cluster"
        },
        {
            "EventCategories": [
                "availability"
            ],
            "Message": "DB instance restarted",
            "Date": "2019-03-18T21:36:40.793Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster",
            "SourceIdentifier": "sample-cluster",
            "SourceType": "db-instance"
        },
        {
            "EventCategories": [],
            "Message": "A new writer was promoted. Restarting database as a
 reader.",
            "Date": "2019-03-18T21:36:43.873Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
            "SourceIdentifier": "sample-cluster2",
            "SourceType": "db-instance"
        },
        {
            "EventCategories": [
                "availability"
            ],
            "Message": "DB instance restarted",
            "Date": "2019-03-18T21:36:51.257Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
            "SourceIdentifier": "sample-cluster2",
            "SourceType": "db-instance"
        },
        {
            "EventCategories": [
                "failover"
            ],
            "Message": "Completed failover to DB instance: sample-cluster",
            "Date": "2019-03-18T21:36:53.462Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-
cluster",
            "SourceIdentifier": "sample-cluster",
            "SourceType": "db-cluster"
        },
        {
```

```
            "Date": "2019-03-19T16:51:48.847Z",
            "EventCategories": [
                "configuration change"
            ],
            "Message": "Updated parameter audit_logs to enabled with apply method
pending-reboot",
            "SourceIdentifier": "custom3-6-param-grp",
            "SourceType": "db-parameter-group"
        },
        {
            "EventCategories": [
                "configuration change"
            ],
            "Message": "Applying modification to database instance class",
            "Date": "2019-03-19T17:55:20.095Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
            "SourceIdentifier": "sample-cluster2",
            "SourceType": "db-instance"
        },
        {
            "EventCategories": [
                "availability"
            ],
            "Message": "DB instance shutdown",
            "Date": "2019-03-19T17:56:31.127Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
            "SourceIdentifier": "sample-cluster2",
            "SourceType": "db-instance"
        },
        {
            "EventCategories": [
                "configuration change"
            ],
            "Message": "Finished applying modification to DB instance class",
            "Date": "2019-03-19T18:00:45.822Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
            "SourceIdentifier": "sample-cluster2",
            "SourceType": "db-instance"
        },
        {
            "EventCategories": [
                "availability"
            ],
            "Message": "DB instance restarted",
```

```
            "Date": "2019-03-19T18:00:53.397Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
            "SourceIdentifier": "sample-cluster2",
            "SourceType": "db-instance"
        },
        {
            "EventCategories": [
                "availability"
            ],
            "Message": "DB instance shutdown",
            "Date": "2019-03-19T18:23:36.045Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
            "SourceIdentifier": "sample-cluster2",
            "SourceType": "db-instance"
        },
        {
            "EventCategories": [
                "availability"
            ],
            "Message": "DB instance restarted",
            "Date": "2019-03-19T18:23:46.209Z",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
            "SourceIdentifier": "sample-cluster2",
            "SourceType": "db-instance"
        },
        {
            "Date": "2019-03-19T18:39:05.822Z",
            "EventCategories": [
                "configuration change"
            ],
            "Message": "Updated parameter ttl_monitor to enabled with apply method
immediate",
            "SourceIdentifier": "custom3-6-param-grp",
            "SourceType": "db-parameter-group"
        },
        {
            "Date": "2019-03-19T18:39:48.067Z",
            "EventCategories": [
                "configuration change"
            ],
            "Message": "Updated parameter audit_logs to disabled with apply method
immediate",
            "SourceIdentifier": "custom3-6-param-grp",
            "SourceType": "db-parameter-group"
```

```
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的檢視 Amazon DocumentDB 事件。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEvents。

`describe-orderable-db-instance-options`

以下程式碼範例顯示如何使用 `describe-orderable-db-instance-options`。

AWS CLI

若要尋找您可以訂購的 Amazon DocumentDB 執行個體選項

下列`describe-orderable-db-instance-options`範例列出區域 Amazon DocumentDB 的所有執行個體選項。

```
aws docdb describe-orderable-db-instance-options \
    --engine docdb \
    --region us-east-1
```

輸出：

```
{
    "OrderableDBInstanceOptions": [
        {
            "Vpc": true,
            "AvailabilityZones": [
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
```

```
            ],
            "EngineVersion": "3.6.0",
            "DBInstanceClass": "db.r4.16xlarge",
            "LicenseModel": "na",
            "Engine": "docdb"
        },
        {
            "Vpc": true,
            "AvailabilityZones": [
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
                }
            ],
            "EngineVersion": "3.6.0",
            "DBInstanceClass": "db.r4.2xlarge",
            "LicenseModel": "na",
            "Engine": "docdb"
        },
        {
            "Vpc": true,
            "AvailabilityZones": [
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
            ],
```

```
            "EngineVersion": "3.6.0",
            "DBInstanceClass": "db.r4.4xlarge",
            "LicenseModel": "na",
            "Engine": "docdb"
        },
        {
            "Vpc": true,
            "AvailabilityZones": [
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
            ],
            "EngineVersion": "3.6.0",
            "DBInstanceClass": "db.r4.8xlarge",
            "LicenseModel": "na",
            "Engine": "docdb"
        },
        {
            "Vpc": true,
            "AvailabilityZones": [
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
            ],
            "EngineVersion": "3.6.0",
            "DBInstanceClass": "db.r4.large",
```

```
            "LicenseModel": "na",
            "Engine": "docdb"
        },
        {
            "Vpc": true,
            "AvailabilityZones": [
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
            ],
            "EngineVersion": "3.6.0",
            "DBInstanceClass": "db.r4.xlarge",
            "LicenseModel": "na",
            "Engine": "docdb"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的將 Amazon DocumentDB 執行個體新增至叢集。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeOrderableDbInstanceOptions。

`describe-pending-maintenance-actions`

以下程式碼範例顯示如何使用 `describe-pending-maintenance-actions`。

AWS CLI

列出待定的 Amazon DocumentDB 維護動作

下列`describe-pending-maintenance-actions`範例列出所有待定的 Amazon DocumentDB 維護動作。

```
aws docdb describe-pending-maintenance-actions
```

輸出：

```
{
    "PendingMaintenanceActions": []
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的維護 Amazon DocumentDB。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePendingMaintenanceActions。

failover-db-cluster

以下程式碼範例顯示如何使用 failover-db-cluster。

AWS CLI

強制 Amazon DocumentDB 叢集容錯移轉至複本

下列failover-db-cluster範例會導致 Amazon DocumentDB 叢集 sample-cluster 中的主要執行個體容錯移轉至複本。

```
aws docdb failover-db-cluster \
    --db-cluster-identifier sample-cluster
```

輸出：

```
{
    "DBCluster": {
        "AssociatedRoles": [],
        "DBClusterIdentifier": "sample-cluster",
        "EngineVersion": "3.6.0",
        "DBSubnetGroup": "default",
        "MasterUsername": "master-user",
        "EarliestRestorableTime": "2019-03-15T20:30:47.020Z",
        "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "AvailabilityZones": [
            "us-west-2a",
            "us-west-2c",
```

```
            "us-west-2b"
        ],
        "LatestRestorableTime": "2019-03-18T21:35:23.548Z",
        "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
        "PreferredBackupWindow": "00:00-00:30",
        "Port": 27017,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "StorageEncrypted": false,
        "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
        "MultiAZ": true,
        "Status": "available",
        "DBClusterMembers": [
            {
                "DBClusterParameterGroupStatus": "in-sync",
                "IsClusterWriter": false,
                "DBInstanceIdentifier": "sample-cluster",
                "PromotionTier": 1
            },
            {
                "DBClusterParameterGroupStatus": "in-sync",
                "IsClusterWriter": true,
                "DBInstanceIdentifier": "sample-cluster2",
                "PromotionTier": 2
            }
        ],
        "EnabledCloudwatchLogsExports": [
            "audit"
        ],
        "DBClusterParameterGroup": "default.docdb3.6",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
        "BackupRetentionPeriod": 3,
        "DbClusterResourceId": "cluster-UP4EF2PVDDFVHHDJQTYDAIGHLE",
        "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "Engine": "docdb"
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的 Amazon DocumentDB 容錯移轉。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 FailoverDbCluster。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出 Amazon DocumentDB 資源上的所有標籤

下列`list-tags-for-resource`範例列出 Amazon DocumentDB 叢集 上的所有標籤`sample-cluster`。

```
aws docdb list-tags-for-resource \
    --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster
```

輸出：

```
{
    "TagList": [
        {
            "Key": "A",
            "Value": "ALPHA"
        },
        {
            "Key": "B",
            "Value": ""
        },
        {
            "Key": "C",
            "Value": "CHARLIE"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的在 Amazon DocumentDB 資源上列出標籤。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

modify-db-cluster-parameter-group

以下程式碼範例顯示如何使用 modify-db-cluster-parameter-group。

AWS CLI

修改 Amazon DocumentDB 資料庫叢集參數群組

下列modify-db-cluster-parameter-group範例會將兩個參數 audit_logs和 設定為ttl_monitor啟用custom3-6-param-grp，以修改 Amazon DocumentDB 叢集參數群組。變更會在下次重新開機時套用。

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name custom3-6-param-grp \
    --
parameters ParameterName=audit_logs,ParameterValue=enabled,ApplyMethod=pending-
reboot \

  ParameterName=ttl_monitor,ParameterValue=enabled,ApplyMethod=pending-reboot
```

輸出：

```
{
    "DBClusterParameterGroupName": "custom3-6-param-grp"
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的修改 Amazon DocumentDB 叢集參數群組。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDbClusterParameterGroup。

modify-db-cluster-snapshot-attribute

以下程式碼範例顯示如何使用 modify-db-cluster-snapshot-attribute。

AWS CLI

範例 1：將屬性新增至 Amazon DocumentDB 快照

下列modify-db-cluster-snapshot-attribute範例會將四個屬性值新增至 Amazon DocumentDB 叢集快照。

```
aws docdb modify-db-cluster-snapshot-attribute \
    --db-cluster-snapshot-identifier sample-cluster-snapshot \
    --attribute-name restore \
    --values-to-add 123456789011 123456789012 123456789013
```

輸出：

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123456789011",
                    "123456789012",
                    "123456789013"
                ]
            }
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"
    }
}
```

範例 2：從 Amazon DocumentDB 快照移除屬性

下列`modify-db-cluster-snapshot-attribute`範例會從 Amazon DocumentDB 叢集快照中移除兩個屬性值。

```
aws docdb modify-db-cluster-snapshot-attribute \
    --db-cluster-snapshot-identifier sample-cluster-snapshot \
    --attribute-name restore \
    --values-to-remove 123456789012
```

輸出：

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123456789011",
```

```
                    "123456789013"
                ]
            }
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的 ModifyDBClusterSnapshotAttribute。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDbClusterSnapshotAttribute。

`modify-db-cluster`

以下程式碼範例顯示如何使用 `modify-db-cluster`。

AWS CLI

修改 Amazon DocumentDB 叢集

下列`modify-db-cluster`範例會將自動備份的保留期設為 7 天，並變更備份和維護的偏好時段，`sample-cluster`藉此修改 Amazon DocumentDB 叢集。所有變更都會在下一個維護時段套用。

```
aws docdb modify-db-cluster \
    --db-cluster-identifier sample-cluster \
    --no-apply-immediately \
    --backup-retention-period 7 \
    --preferred-backup-window 18:00-18:30 \
    --preferred-maintenance-window sun:20:00-sun:20:30
```

輸出：

```
{
    "DBCluster": {
        "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "DBClusterMembers": [
            {
                "DBClusterParameterGroupStatus": "in-sync",
                "DBInstanceIdentifier": "sample-cluster",
                "IsClusterWriter": true,
```

```
                "PromotionTier": 1
            },
            {
                "DBClusterParameterGroupStatus": "in-sync",
                "DBInstanceIdentifier": "sample-cluster2",
                "IsClusterWriter": false,
                "PromotionTier": 2
            }
        ],
        "HostedZoneId": "ZNKXH85TT8WVW",
        "StorageEncrypted": false,
        "PreferredBackupWindow": "18:00-18:30",
        "MultiAZ": true,
        "EngineVersion": "3.6.0",
        "MasterUsername": "master-user",
        "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "DBSubnetGroup": "default",
        "LatestRestorableTime": "2019-03-18T22:08:13.408Z",
        "EarliestRestorableTime": "2019-03-15T20:30:47.020Z",
        "PreferredMaintenanceWindow": "sun:20:00-sun:20:30",
        "AssociatedRoles": [],
        "EnabledCloudwatchLogsExports": [
            "audit"
        ],
        "Engine": "docdb",
        "DBClusterParameterGroup": "default.docdb3.6",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
        "BackupRetentionPeriod": 7,
        "DBClusterIdentifier": "sample-cluster",
        "AvailabilityZones": [
            "us-west-2a",
            "us-west-2c",
            "us-west-2b"
        ],
        "Status": "available",
        "DbClusterResourceId": "cluster-UP4EF2PVDDFVHHDJQTYDAIGHLE",
        "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
```

```
        "Port": 27017
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的修改 Amazon DocumentDB 叢集。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDbCluster。

`modify-db-instance`

以下程式碼範例顯示如何使用 `modify-db-instance`。

AWS CLI

修改 Amazon DocumentDB 執行個體

下列`modify-db-instance`範例會將 Amazon DocumentDB 執行個體的執行個體類別變更為 `db.r4.4xlarge`，並將其提升層變更為 `sample-cluster2`，藉此修改 Amazon DocumentDB 執行個體5。變更會立即套用，但只有在執行個體狀態可用後才能看到。

```
aws docdb modify-db-instance \
    --db-instance-identifier sample-cluster2 \
    --apply-immediately \
    --db-instance-class db.r4.4xlarge \
    --promotion-tier 5
```

輸出：

```
{
    "DBInstance": {
        "EngineVersion": "3.6.0",
        "StorageEncrypted": false,
        "DBInstanceClass": "db.r4.large",
        "PreferredMaintenanceWindow": "mon:08:39-mon:09:09",
        "AutoMinorVersionUpgrade": true,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "PreferredBackupWindow": "18:00-18:30",
```

```
        "EnabledCloudwatchLogsExports": [
            "audit"
        ],
        "AvailabilityZone": "us-west-2f",
        "DBInstanceIdentifier": "sample-cluster2",
        "InstanceCreateTime": "2019-03-15T20:36:06.338Z",
        "Engine": "docdb",
        "BackupRetentionPeriod": 7,
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-4e26d263",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-afc329f4",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-53ab3636",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-991cb8d0",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetStatus": "Active"
                }
            ],
            "VpcId": "vpc-91280df6"
        },
```

```
        "PromotionTier": 2,
        "Endpoint": {
            "Address": "sample-cluster2.corcjozrlsfc.us-west-2.docdb.amazonaws.com",
            "HostedZoneId": "ZNKXH85TT8WVW",
            "Port": 27017
        },
        "DbiResourceId": "db-A2GIKUV6KPOHITGGKI2NHVISZA",
        "DBClusterIdentifier": "sample-cluster",
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
        "PendingModifiedValues": {
            "DBInstanceClass": "db.r4.4xlarge"
        },
        "PubliclyAccessible": false,
        "DBInstanceStatus": "available"
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的修改 Amazon DocumentDB 執行個體。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDbInstance。

modify-db-subnet-group

以下程式碼範例顯示如何使用 modify-db-subnet-group。

AWS CLI

修改 Amazon DocumentDB 子網路群組

下列modify-db-subnet-group範例sample-subnet-group透過新增指定的子網路和新描述來修改子網路群組。

```
aws docdb modify-db-subnet-group \
    --db-subnet-group-name sample-subnet-group \
    --subnet-ids subnet-b3806e8f subnet-53ab3636 subnet-991cb8d0 \
    --db-subnet-group-description "New subnet description"
```

輸出：

```
{
    "DBSubnetGroup": {
        "DBSubnetGroupName": "sample-subnet-group",
```

```
        "SubnetGroupStatus": "Complete",
        "DBSubnetGroupArn": "arn:aws:rds:us-west-2:123456789012:subgrp:sample-
subnet-group",
        "VpcId": "vpc-91280df6",
        "DBSubnetGroupDescription": "New subnet description",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-b3806e8f",
                "SubnetStatus": "Active",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                }
            },
            {
                "SubnetIdentifier": "subnet-53ab3636",
                "SubnetStatus": "Active",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2c"
                }
            },
            {
                "SubnetIdentifier": "subnet-991cb8d0",
                "SubnetStatus": "Active",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                }
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的修改 Amazon DocumentDB 子網路群組。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDbSubnetGroup。

`reboot-db-instance`

以下程式碼範例顯示如何使用 `reboot-db-instance`。

AWS CLI

重新啟動 Amazon DocumentDB 執行個體

下列reboot-db-instance範例會重新啟動 Amazon DocumentDB 執行個體 sample-cluster2。

```
aws docdb reboot-db-instance \
    --db-instance-identifier sample-cluster2
```

此命令不會產生輸出。輸出：

```
{
    "DBInstance": {
        "PreferredBackupWindow": "18:00-18:30",
        "DBInstanceIdentifier": "sample-cluster2",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "DBSubnetGroup": {
            "VpcId": "vpc-91280df6",
            "Subnets": [
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetIdentifier": "subnet-4e26d263"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    },
                    "SubnetIdentifier": "subnet-afc329f4"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    },
                    "SubnetIdentifier": "subnet-53ab3636"
                },
                {
```

```
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetIdentifier": "subnet-991cb8d0"
                }
            ],
            "SubnetGroupStatus": "Complete",
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default"
        },
        "PendingModifiedValues": {},
        "Endpoint": {
            "Address": "sample-cluster2.corcjozrlsfc.us-west-2.docdb.amazonaws.com",
            "HostedZoneId": "ZNKXH85TT8WVW",
            "Port": 27017
        },
        "EnabledCloudwatchLogsExports": [
            "audit"
        ],
        "StorageEncrypted": false,
        "DbiResourceId": "db-A2GIKUV6KPOHITGGKI2NHVISZA",
        "AutoMinorVersionUpgrade": true,
        "Engine": "docdb",
        "InstanceCreateTime": "2019-03-15T20:36:06.338Z",
        "EngineVersion": "3.6.0",
        "PromotionTier": 5,
        "BackupRetentionPeriod": 7,
        "DBClusterIdentifier": "sample-cluster",
        "PreferredMaintenanceWindow": "mon:08:39-mon:09:09",
        "PubliclyAccessible": false,
        "DBInstanceClass": "db.r4.4xlarge",
        "AvailabilityZone": "us-west-2d",
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
        "DBInstanceStatus": "rebooting"
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的重新啟動 Amazon DocumentDB ILnstance。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RebootDbInstance。

remove-tags-from-resource

以下程式碼範例顯示如何使用 remove-tags-from-resource。

AWS CLI

從 Amazon DocumentDB 資源移除標籤

下列remove-tags-from-resource範例B會從 Amazon DocumentDB 叢集 中移除名為 的金鑰標籤sample-cluster。

```
aws docdb remove-tags-from-resource \
    --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster \
    --tag-keys B
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的從 Amazon DocumentDBResource 移除標籤。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 RemoveTagsFromResource。

reset-db-cluster-parameter-group

以下程式碼範例顯示如何使用 reset-db-cluster-parameter-group。

AWS CLI

在 Amazon DocumentDB 參數群組中將指定的參數值重設為其預設值

下列reset-db-cluster-parameter-group範例會將 Amazon DocumentDB 參數群組ttl_monitor中的 參數重設custom3-6-param-grp為其預設值。

```
aws docdb reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name custom3-6-param-grp \
    --parameters ParameterName=ttl_monitor,ApplyMethod=immediate
```

輸出：

```
{
```

```
    "DBClusterParameterGroupName": "custom3-6-param-grp"
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的標題。

將指定或所有參數值重設為 Amazon DocumentDB 參數群組中的預設值

下列`reset-db-cluster-parameter-group`範例會將 Amazon DocumentDB 參數群組中的所有參數重設`custom3-6-param-grp`為其預設值。

```
aws docdb reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name custom3-6-param-grp \
    --reset-all-parameters
```

輸出：

```
{
    "DBClusterParameterGroupName": "custom3-6-param-grp"
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的重設 Amazon DocumentDB 叢集參數群組。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetDbClusterParameterGroup。

`restore-db-cluster-from-snapshot`

以下程式碼範例顯示如何使用 `restore-db-cluster-from-snapshot`。

AWS CLI

從自動或手動快照還原 Amazon DocumentDB 叢集

下列`restore-db-cluster-from-snapshot`範例`sample-cluster-2019-03-16-00-01-restored`會從快照 建立名為 的新 Amazon DocumentDB 叢集`rds:sample-cluster-2019-03-16-00-01`。

```
aws docdb restore-db-cluster-from-snapshot \
    --db-cluster-identifier sample-cluster-2019-03-16-00-01-restored \
    --engine docdb \
```

```
    --snapshot-identifier rds:sample-cluster-2019-03-16-00-01
```

輸出：

```
{
    "DBCluster": {
        "ClusterCreateTime": "2019-03-19T18:45:01.857Z",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "Engine": "docdb",
        "DBClusterMembers": [],
        "MultiAZ": false,
        "AvailabilityZones": [
            "us-west-2a",
            "us-west-2c",
            "us-west-2b"
        ],
        "StorageEncrypted": false,
        "ReaderEndpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-ro-
corcjozrlsfc.us-west-2.docdb.amazonaws.com",
        "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-
corcjozrlsfc.us-west-2.docdb.amazonaws.com",
        "Port": 27017,
        "PreferredBackupWindow": "00:00-00:30",
        "DBSubnetGroup": "default",
        "DBClusterIdentifier": "sample-cluster-2019-03-16-00-01-restored",
        "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-
cluster-2019-03-16-00-01-restored",
        "DBClusterParameterGroup": "default.docdb3.6",
        "DbClusterResourceId": "cluster-XOO46Q3RH4LWSYNH3NMZKXPISU",
        "MasterUsername": "master-user",
        "EngineVersion": "3.6.0",
        "BackupRetentionPeriod": 3,
        "AssociatedRoles": [],
        "Status": "creating",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的從叢集快照還原。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreDbClusterFromSnapshot。

`restore-db-cluster-to-point-in-time`

以下程式碼範例顯示如何使用 `restore-db-cluster-to-point-in-time`。

AWS CLI

從手動快照將 Amazon DocumentDB 叢集還原至point-in-time

下列`restore-db-cluster-to-point-in-time`範例使用 `sample-cluster-snapshot`，使用最新的可還原時間建立新的 Amazon DocumentDB 叢集 `sample-cluster-pit`。

```
aws docdb restore-db-cluster-to-point-in-time \
    --db-cluster-identifier sample-cluster-pit \
    --source-db-cluster-identifier arn:aws:rds:us-
west-2:123456789012:cluster:sample-cluster \
    --use-latest-restorable-time
```

輸出：

```
{
    "DBCluster": {
        "StorageEncrypted": false,
        "BackupRetentionPeriod": 3,
        "MasterUsername": "master-user",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "PreferredBackupWindow": "00:00-00:30",
        "MultiAZ": false,
        "DBClusterIdentifier": "sample-cluster-pit",
        "DBSubnetGroup": "default",
        "ClusterCreateTime": "2019-04-03T15:55:21.320Z",
        "AssociatedRoles": [],
        "DBClusterParameterGroup": "default.docdb3.6",
        "DBClusterMembers": [],
        "Status": "creating",
        "AvailabilityZones": [
            "us-west-2a",
            "us-west-2d",
            "us-west-2b"
```

```
        ],
        "ReaderEndpoint": "sample-cluster-pit.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "Port": 27017,
        "Engine": "docdb",
        "EngineVersion": "3.6.0",
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
        "Endpoint": "sample-cluster-pit.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "DbClusterResourceId": "cluster-NLCABBXOSE2QPQ4GOLZIFWEPLM",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster-
pit"
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的將快照還原至時間點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreDbClusterToPointInTime。

`start-db-cluster`

以下程式碼範例顯示如何使用 `start-db-cluster`。

AWS CLI

啟動已停止的 Amazon DocumentDB 叢集

下列`start-db-cluster`範例會啟動指定的 Amazon DocumentDB 叢集。

```
aws docdb start-db-cluster \
    --db-cluster-identifier sample-cluster
```

輸出：

```
{
    "DBCluster": {
```

```
        "ClusterCreateTime": "2019-03-19T18:45:01.857Z",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "Engine": "docdb",
        "DBClusterMembers": [],
        "MultiAZ": false,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1f"
        ],
        "StorageEncrypted": false,
        "ReaderEndpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-ro-
corcjozrlsfc.us-east-1.docdb.amazonaws.com",
        "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-
corcjozrlsfc.us-east-1.docdb.amazonaws.com",
        "Port": 27017,
        "PreferredBackupWindow": "00:00-00:30",
        "DBSubnetGroup": "default",
        "DBClusterIdentifier": "sample-cluster-2019-03-16-00-01-restored",
        "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-
cluster-2019-03-16-00-01-restored",
        "DBClusterParameterGroup": "default.docdb3.6",
        "DbClusterResourceId": "cluster-XOO46Q3RH4LWSYNH3NMZKXPISU",
        "MasterUsername": "master-user",
        "EngineVersion": "3.6.0",
        "BackupRetentionPeriod": 3,
        "AssociatedRoles": [],
        "Status": "creating",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的停止和啟動 Amazon DocumentDB 叢集。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartDbCluster。

stop-db-cluster

以下程式碼範例顯示如何使用 stop-db-cluster。

AWS CLI

停止執行中的 Amazon DocumentDB 叢集

下列stop-db-cluster範例會停止指定的 Amazon DocumentDB 叢集。

```
aws docdb stop-db-cluster \
    --db-cluster-identifier sample-cluster
```

輸出：

```
{
    "DBCluster": {
        "ClusterCreateTime": "2019-03-19T18:45:01.857Z",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "Engine": "docdb",
        "DBClusterMembers": [],
        "MultiAZ": false,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1f"
        ],
        "StorageEncrypted": false,
        "ReaderEndpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-ro-
corcjozrlsfc.us-east-1.docdb.amazonaws.com",
        "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-
corcjozrlsfc.us-east-1.docdb.amazonaws.com",
        "Port": 27017,
        "PreferredBackupWindow": "00:00-00:30",
        "DBSubnetGroup": "default",
        "DBClusterIdentifier": "sample-cluster-2019-03-16-00-01-restored",
        "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-
cluster-2019-03-16-00-01-restored",
        "DBClusterParameterGroup": "default.docdb3.6",
        "DbClusterResourceId": "cluster-XOO46Q3RH4LWSYNH3NMZKXPISU",
        "MasterUsername": "master-user",
        "EngineVersion": "3.6.0",
```

```
        "BackupRetentionPeriod": 3,
        "AssociatedRoles": [],
        "Status": "creating",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon DocumentDB 開發人員指南》中的停止和啟動 Amazon DocumentDB 叢集。 Amazon DocumentDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopDbCluster。

使用 的 DynamoDB 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 DynamoDB 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`batch-get-item`

以下程式碼範例顯示如何使用 `batch-get-item`。

AWS CLI

從資料表擷取多個項目

下列`batch-get-items`範例會使用批次三個`GetItem`請求從`MusicCollection`資料表讀取多個項目，並請求 操作使用的讀取容量單位數量。命令只會傳回 `AlbumTitle` 屬性。

```
aws dynamodb batch-get-item \
    --request-items file://request-items.json \
    --return-consumed-capacity TOTAL
```

request-items.json 的內容：

```
{
    "MusicCollection": {
        "Keys": [
            {
                "Artist": {"S": "No One You Know"},
                "SongTitle": {"S": "Call Me Today"}
            },
            {
                "Artist": {"S": "Acme Band"},
                "SongTitle": {"S": "Happy Day"}
            },
            {
                "Artist": {"S": "No One You Know"},
                "SongTitle": {"S": "Scared of My Shadow"}
            }
        ],
        "ProjectionExpression":"AlbumTitle"
    }
}
```

輸出：

```
{
    "Responses": {
        "MusicCollection": [
            {
                "AlbumTitle": {
                    "S": "Somewhat Famous"
                }
            },
            {
                "AlbumTitle": {
                    "S": "Blue Sky Blues"
                }
            },
            {
```

```
                "AlbumTitle": {
                    "S": "Louder Than Ever"
                }
            }
        ]
    },
    "UnprocessedKeys": {},
    "ConsumedCapacity": [
        {
            "TableName": "MusicCollection",
            "CapacityUnits": 1.5
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的批次操作。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 BatchGetItem。

batch-write-item

以下程式碼範例顯示如何使用 batch-write-item。

AWS CLI

將多個項目新增至資料表

下列batch-write-item範例使用批次三個PutItem請求，將三個新項目新增至MusicCollection資料表。它還請求有關操作消耗的寫入容量單位數量以及操作修改的任何項目集合的資訊。

```
aws dynamodb batch-write-item \
    --request-items file://request-items.json \
    --return-consumed-capacity INDEXES \
    --return-item-collection-metrics SIZE
```

request-items.json 的內容：

```
{
    "MusicCollection": [
        {
            "PutRequest": {
```

```
                "Item": {
                    "Artist": {"S": "No One You Know"},
                    "SongTitle": {"S": "Call Me Today"},
                    "AlbumTitle": {"S": "Somewhat Famous"}
                }
            }
        },
        {
            "PutRequest": {
                "Item": {
                    "Artist": {"S": "Acme Band"},
                    "SongTitle": {"S": "Happy Day"},
                    "AlbumTitle": {"S": "Songs About Life"}
                }
            }
        },
        {
            "PutRequest": {
                "Item": {
                    "Artist": {"S": "No One You Know"},
                    "SongTitle": {"S": "Scared of My Shadow"},
                    "AlbumTitle": {"S": "Blue Sky Blues"}
                }
            }
        }
    ]
}
```

輸出：

```
{
    "UnprocessedItems": {},
    "ItemCollectionMetrics": {
        "MusicCollection": [
            {
                "ItemCollectionKey": {
                    "Artist": {
                        "S": "No One You Know"
                    }
                },
                "SizeEstimateRangeGB": [
                    0.0,
                    1.0
```

```
                ]
            },
            {
                "ItemCollectionKey": {
                    "Artist": {
                        "S": "Acme Band"
                    }
                },
                "SizeEstimateRangeGB": [
                    0.0,
                    1.0
                ]
            }
        ]
    },
    "ConsumedCapacity": [
        {
            "TableName": "MusicCollection",
            "CapacityUnits": 6.0,
            "Table": {
                "CapacityUnits": 3.0
            },
            "LocalSecondaryIndexes": {
                "AlbumTitleIndex": {
                    "CapacityUnits": 3.0
                }
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的批次操作。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 BatchWriteItem。

`create-backup`

以下程式碼範例顯示如何使用 `create-backup`。

AWS CLI

為現有的 DynamoDB 資料表建立備份

下列`create-backup`範例會建立 `MusicCollection` 資料表的備份。

```
aws dynamodb create-backup \
    --table-name MusicCollection \
    --backup-name MusicCollectionBackup
```

輸出：

```
{
    "BackupDetails": {
        "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/
backup/01576616366715-b4e58d3a",
        "BackupName": "MusicCollectionBackup",
        "BackupSizeBytes": 0,
        "BackupStatus": "CREATING",
        "BackupType": "USER",
        "BackupCreationDateTime": 1576616366.715
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 的隨需備份與還原。 DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateBackup。

create-global-table

以下程式碼範例顯示如何使用 create-global-table。

AWS CLI

建立全域資料表

下列create-global-table範例會從指定、個別 AWS 區域中的兩個相同資料表建立全域資料表。

```
aws dynamodb create-global-table \
    --global-table-name MusicCollection \
    --replication-group RegionName=us-east-2 RegionName=us-east-1 \
    --region us-east-2
```

輸出：

```
{
    "GlobalTableDescription": {
        "ReplicationGroup": [
            {
                "RegionName": "us-east-2"
            },
            {
                "RegionName": "us-east-1"
            }
        ],
        "GlobalTableArn": "arn:aws:dynamodb::123456789012:global-table/
MusicCollection",
        "CreationDateTime": 1576625818.532,
        "GlobalTableStatus": "CREATING",
        "GlobalTableName": "MusicCollection"
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 全域資料表。DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateGlobalTable。

create-table

以下程式碼範例顯示如何使用 create-table。

AWS CLI

範例 1：建立具有標籤的資料表

下列create-table範例使用指定的屬性和金鑰結構描述來建立名為 的資料表MusicCollection。此資料表使用佈建的輸送量，並使用預設 AWS 擁有的 CMK 進行靜態加密。命令也會將標籤套用至資料表，索引鍵為 Owner且值為 blueTeam。

```
aws dynamodb create-table \
    --table-name MusicCollection \
    --attribute-
definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S
 \
    --key-
schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE \
```

```
    --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5 \
    --tags Key=Owner,Value=blueTeam
```

輸出：

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "WriteCapacityUnits": 5,
            "ReadCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "TableName": "MusicCollection",
        "TableStatus": "CREATING",
        "KeySchema": [
            {
                "KeyType": "HASH",
                "AttributeName": "Artist"
            },
            {
                "KeyType": "RANGE",
                "AttributeName": "SongTitle"
            }
        ],
        "ItemCount": 0,
        "CreationDateTime": "2020-05-26T16:04:41.627000-07:00",
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的資料表的基本操作。

範例 2：在隨需模式下建立資料表

下列範例MusicCollection會使用隨需模式建立名為 的資料表，而非佈建的輸送量模式。這適用於工作負載無法預測的資料表。

```
aws dynamodb create-table \
    --table-name MusicCollection \
    --attribute-
definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S
 \
    --key-
schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE \
    --billing-mode PAY_PER_REQUEST
```

輸出：

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "TableName": "MusicCollection",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2020-05-27T11:44:10.807000-07:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
```

```
            "ReadCapacityUnits": 0,
            "WriteCapacityUnits": 0
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "BillingModeSummary": {
            "BillingMode": "PAY_PER_REQUEST"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的資料表的基本操作。

範例 3：建立資料表並使用客戶受管 CMK 加密

下列範例會建立名為 的資料表，MusicCollection並使用客戶受管 CMK 將其加密。

```
aws dynamodb create-table \
    --table-name MusicCollection \
    --attribute-
definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S
 \
    --key-
schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5 \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=abcd1234-abcd-1234-
a123-ab1234a1b234
```

輸出：

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
```

```
            }
        ],
        "TableName": "MusicCollection",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2020-05-27T11:12:16.431000-07:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "SSEDescription": {
            "Status": "ENABLED",
            "SSEType": "KMS",
            "KMSMasterKeyArn": "arn:aws:kms:us-west-2:123456789012:key/abcd1234-
abcd-1234-a123-ab1234a1b234"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的資料表的基本操作。

範例 4：使用本機次要索引建立資料表

下列範例使用指定的屬性和金鑰結構描述來建立名為 的資料表MusicCollection，其中包含名為 的 Local Secondary IndexAlbumTitleIndex。

```
aws dynamodb create-table \
    --table-name MusicCollection \
```

```
    --attribute-
definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S Att
 \
    --key-
schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
    --local-secondary-indexes \
        "[
            {
                \"IndexName\": \"AlbumTitleIndex\",
                \"KeySchema\": [
                    {\"AttributeName\": \"Artist\",\"KeyType\":\"HASH\"},
                    {\"AttributeName\": \"AlbumTitle\",\"KeyType\":\"RANGE\"}
                ],
                \"Projection\": {
                    \"ProjectionType\": \"INCLUDE\",
                    \"NonKeyAttributes\": [\"Genre\", \"Year\"]
                }
            }
        ]"
```

輸出：

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "AlbumTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "TableName": "MusicCollection",
        "KeySchema": [
            {
                "AttributeName": "Artist",
```

```
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "LocalSecondaryIndexes": [
            {
                "IndexName": "AlbumTitleIndex",
                "KeySchema": [
                    {
                        "AttributeName": "Artist",
                        "KeyType": "HASH"
                    },
                    {
                        "AttributeName": "AlbumTitle",
                        "KeyType": "RANGE"
                    }
                ],
                "Projection": {
                    "ProjectionType": "INCLUDE",
                    "NonKeyAttributes": [
                        "Genre",
                        "Year"
                    ]
                },
                "IndexSizeBytes": 0,
                "ItemCount": 0,
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitleIndex"
            }
        ]
```

```
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的資料表的基本操作。

範例 5：使用全域次要索引建立資料表

下列範例會使用名為 GameScores的全域次要索引建立名為 的資料表GameTitleIndex。基礎資料表具有分割區索引鍵 UserId 和排序索引鍵 GameTitle，可讓您有效率地找到個別使用者在特定遊戲中的最佳分數，而 GSI 具有分割區索引鍵 GameTitle 和排序索引鍵 TopScore，可讓您快速找到特定遊戲的整體最高分數。

```
aws dynamodb create-table \
    --table-name GameScores \
    --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S Att
 \
    --key-schema AttributeName=UserId,KeyType=HASH \
                 AttributeName=GameTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
    --global-secondary-indexes \
        "[
            {
                \"IndexName\": \"GameTitleIndex\",
                \"KeySchema\": [
                    {\"AttributeName\":\"GameTitle\",\"KeyType\":\"HASH\"},
                    {\"AttributeName\":\"TopScore\",\"KeyType\":\"RANGE\"}
                ],
                \"Projection\": {
                    \"ProjectionType\":\"INCLUDE\",
                    \"NonKeyAttributes\":[\"UserId\"]
                },
                \"ProvisionedThroughput\": {
                    \"ReadCapacityUnits\": 10,
                    \"WriteCapacityUnits\": 5
                }
            }
        ]"
```

輸出：

```
{
    "TableDescription": {
```

```
        "AttributeDefinitions": [
            {
                "AttributeName": "GameTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "TopScore",
                "AttributeType": "N"
            },
            {
                "AttributeName": "UserId",
                "AttributeType": "S"
            }
        ],
        "TableName": "GameScores",
        "KeySchema": [
            {
                "AttributeName": "UserId",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "GameTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2020-05-26T17:28:15.602000-07:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "GlobalSecondaryIndexes": [
            {
                "IndexName": "GameTitleIndex",
                "KeySchema": [
                    {
                        "AttributeName": "GameTitle",
                        "KeyType": "HASH"
                    },
```

```
                    {
                        "AttributeName": "TopScore",
                        "KeyType": "RANGE"
                    }
                ],
                "Projection": {
                    "ProjectionType": "INCLUDE",
                    "NonKeyAttributes": [
                        "UserId"
                    ]
                },
                "IndexStatus": "CREATING",
                "ProvisionedThroughput": {
                    "NumberOfDecreasesToday": 0,
                    "ReadCapacityUnits": 10,
                    "WriteCapacityUnits": 5
                },
                "IndexSizeBytes": 0,
                "ItemCount": 0,
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/index/GameTitleIndex"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的資料表的基本操作。

範例 6：建立具有多個全域次要索引的資料表

下列範例會建立名為 的資料表，GameScores其中包含兩個全域次要索引。GSI 結構描述是透過檔案傳遞，而不是在命令列上傳遞。

```
aws dynamodb create-table \
    --table-name GameScores \
    --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S Att
 \
    --key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
    --global-secondary-indexes file://gsi.json
```

gsi.json 的內容：

```
[
    {
        "IndexName": "GameTitleIndex",
        "KeySchema": [
            {
                "AttributeName": "GameTitle",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "TopScore",
                "KeyType": "RANGE"
            }
        ],
        "Projection": {
            "ProjectionType": "ALL"
        },
        "ProvisionedThroughput": {
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        }
    },
    {
        "IndexName": "GameDateIndex",
        "KeySchema": [
            {
                "AttributeName": "GameTitle",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "Date",
                "KeyType": "RANGE"
            }
        ],
        "Projection": {
            "ProjectionType": "ALL"
        },
        "ProvisionedThroughput": {
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 5
        }
    }
```

```
]
```

輸出：

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "Date",
                "AttributeType": "S"
            },
            {
                "AttributeName": "GameTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "TopScore",
                "AttributeType": "N"
            },
            {
                "AttributeName": "UserId",
                "AttributeType": "S"
            }
        ],
        "TableName": "GameScores",
        "KeySchema": [
            {
                "AttributeName": "UserId",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "GameTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2020-08-04T16:40:55.524000-07:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
```

```
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "GlobalSecondaryIndexes": [
            {
                "IndexName": "GameTitleIndex",
                "KeySchema": [
                    {
                        "AttributeName": "GameTitle",
                        "KeyType": "HASH"
                    },
                    {
                        "AttributeName": "TopScore",
                        "KeyType": "RANGE"
                    }
                ],
                "Projection": {
                    "ProjectionType": "ALL"
                },
                "IndexStatus": "CREATING",
                "ProvisionedThroughput": {
                    "NumberOfDecreasesToday": 0,
                    "ReadCapacityUnits": 10,
                    "WriteCapacityUnits": 5
                },
                "IndexSizeBytes": 0,
                "ItemCount": 0,
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/index/GameTitleIndex"
            },
            {
                "IndexName": "GameDateIndex",
                "KeySchema": [
                    {
                        "AttributeName": "GameTitle",
                        "KeyType": "HASH"
                    },
                    {
                        "AttributeName": "Date",
                        "KeyType": "RANGE"
                    }
                ],
                "Projection": {
                    "ProjectionType": "ALL"
```

```
                },
                "IndexStatus": "CREATING",
                "ProvisionedThroughput": {
                    "NumberOfDecreasesToday": 0,
                    "ReadCapacityUnits": 5,
                    "WriteCapacityUnits": 5
                },
                "IndexSizeBytes": 0,
                "ItemCount": 0,
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/index/GameDateIndex"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的資料表的基本操作。

範例 7：建立已啟用串流的資料表

下列範例會建立名為 GameScores且已啟用 DynamoDB Streams 的資料表。每個項目的新舊映像都會寫入串流。

```
aws dynamodb create-table \
    --table-name GameScores \
    --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S
 \
    --key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
    --stream-specification StreamEnabled=TRUE,StreamViewType=NEW_AND_OLD_IMAGES
```

輸出：

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "GameTitle",
                "AttributeType": "S"
            },
```

```
            {
                "AttributeName": "UserId",
                "AttributeType": "S"
            }
        ],
        "TableName": "GameScores",
        "KeySchema": [
            {
                "AttributeName": "UserId",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "GameTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2020-05-27T10:49:34.056000-07:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "StreamSpecification": {
            "StreamEnabled": true,
            "StreamViewType": "NEW_AND_OLD_IMAGES"
        },
        "LatestStreamLabel": "2020-05-27T17:49:34.056",
        "LatestStreamArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/stream/2020-05-27T17:49:34.056"
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的資料表的基本操作。

範例 8：建立已啟用僅限金鑰串流的資料表

下列範例會建立名為 GameScores且已啟用 DynamoDB Streams 的資料表。只有修改項目的關鍵屬性才會寫入串流。

```
aws dynamodb create-table \
    --table-name GameScores \
    --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S
 \
    --key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
    --stream-specification StreamEnabled=TRUE,StreamViewType=KEYS_ONLY
```

輸出：

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "GameTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "UserId",
                "AttributeType": "S"
            }
        ],
        "TableName": "GameScores",
        "KeySchema": [
            {
                "AttributeName": "UserId",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "GameTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2023-05-25T18:45:34.140000+00:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
```

```
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "StreamSpecification": {
            "StreamEnabled": true,
            "StreamViewType": "KEYS_ONLY"
        },
        "LatestStreamLabel": "2023-05-25T18:45:34.140",
        "LatestStreamArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/stream/2023-05-25T18:45:34.140",
        "DeletionProtectionEnabled": false
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的變更 DynamoDB Streams 的資料擷取。 DynamoDB

範例 9：使用標準不常存取類別建立資料表

下列範例會建立名為 的資料表，GameScores並指派 Standard-Infrequent Access (DynamoDB Standard-IA) 資料表類別。此資料表類別已針對主要成本的儲存進行最佳化。

```
aws dynamodb create-table \
    --table-name GameScores \
    --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S
 \
    --key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
    --table-class STANDARD_INFREQUENT_ACCESS
```

輸出：

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "GameTitle",
                "AttributeType": "S"
            },
            {
```

```
                "AttributeName": "UserId",
                "AttributeType": "S"
            }
        ],
        "TableName": "GameScores",
        "KeySchema": [
            {
                "AttributeName": "UserId",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "GameTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2023-05-25T18:33:07.581000+00:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "TableClassSummary": {
            "TableClass": "STANDARD_INFREQUENT_ACCESS"
        },
        "DeletionProtectionEnabled": false
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的資料表類別。

範例 10：建立已啟用刪除保護的資料表

下列範例會建立名為 的資料表，GameScores並啟用刪除保護。

```
aws dynamodb create-table \
    --table-name GameScores \
    --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S
 \
```

```
    --key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
    --deletion-protection-enabled
```

輸出：

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "GameTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "UserId",
                "AttributeType": "S"
            }
        ],
        "TableName": "GameScores",
        "KeySchema": [
            {
                "AttributeName": "UserId",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "GameTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2023-05-25T23:02:17.093000+00:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "DeletionProtectionEnabled": true
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的使用刪除保護。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateTable。

delete-backup

以下程式碼範例顯示如何使用 delete-backup。

AWS CLI

刪除現有的 DynamoDB 備份

下列delete-backup範例會刪除指定的現有備份。

```
aws dynamodb delete-backup \
    --backup-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/
backup/01576616366715-b4e58d3a
```

輸出：

```
{
    "BackupDescription": {
        "BackupDetails": {
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01576616366715-b4e58d3a",
            "BackupName": "MusicCollectionBackup",
            "BackupSizeBytes": 0,
            "BackupStatus": "DELETED",
            "BackupType": "USER",
            "BackupCreationDateTime": 1576616366.715
        },
        "SourceTableDetails": {
            "TableName": "MusicCollection",
            "TableId": "b0c04bcc-309b-4352-b2ae-9088af169fe2",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "TableSizeBytes": 0,
            "KeySchema": [
                {
                    "AttributeName": "Artist",
                    "KeyType": "HASH"
```

```
                },
                {
                    "AttributeName": "SongTitle",
                    "KeyType": "RANGE"
                }
            ],
            "TableCreationDateTime": 1576615228.571,
            "ProvisionedThroughput": {
                "ReadCapacityUnits": 5,
                "WriteCapacityUnits": 5
            },
            "ItemCount": 0,
            "BillingMode": "PROVISIONED"
        },
        "SourceTableFeatureDetails": {}
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 的隨需備份與還原。 DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBackup。

delete-item

以下程式碼範例顯示如何使用 delete-item。

AWS CLI

範例 1：刪除項目

下列delete-item範例會從MusicCollection資料表中刪除項目，並請求有關已刪除項目的詳細資訊，以及請求使用的容量。

```
aws dynamodb delete-item \
    --table-name MusicCollection \
    --key file://key.json \
    --return-values ALL_OLD \
    --return-consumed-capacity TOTAL \
    --return-item-collection-metrics SIZE
```

key.json 的內容：

```
{
    "Artist": {"S": "No One You Know"},
    "SongTitle": {"S": "Scared of My Shadow"}
}
```

輸出：

```
{
    "Attributes": {
        "AlbumTitle": {
            "S": "Blue Sky Blues"
        },
        "Artist": {
            "S": "No One You Know"
        },
        "SongTitle": {
            "S": "Scared of My Shadow"
        }
    },
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 2.0
    },
    "ItemCollectionMetrics": {
        "ItemCollectionKey": {
            "Artist": {
                "S": "No One You Know"
            }
        },
        "SizeEstimateRangeGB": [
            0.0,
            1.0
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的撰寫項目。

範例 2：有條件地刪除項目

下列範例只有在項目ProductCategory為 Sporting Goods或 Gardening Supplies且其價格介於 500 到 600 之間時，才會從ProductCatalog資料表中刪除項目。它會傳回已刪除項目的詳細資訊。

```
aws dynamodb delete-item \
    --table-name ProductCatalog \
    --key '{"Id":{"N":"456"}}' \
    --condition-expression "(ProductCategory IN (:cat1, :cat2)) and (#P between :lo
 and :hi)" \
    --expression-attribute-names file://names.json \
    --expression-attribute-values file://values.json \
    --return-values ALL_OLD
```

names.json 的內容：

```
{
    "#P": "Price"
}
```

values.json 的內容：

```
{
    ":cat1": {"S": "Sporting Goods"},
    ":cat2": {"S": "Gardening Supplies"},
    ":lo": {"N": "500"},
    ":hi": {"N": "600"}
}
```

輸出：

```
{
    "Attributes": {
        "Id": {
            "N": "456"
        },
        "Price": {
            "N": "550"
        },
        "ProductCategory": {
            "S": "Sporting Goods"
        }
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的撰寫項目。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteItem。

delete-table

以下程式碼範例顯示如何使用 delete-table。

AWS CLI

刪除資料表

下列delete-table範例會刪除MusicCollection資料表。

```
aws dynamodb delete-table \
    --table-name MusicCollection
```

輸出：

```
{
    "TableDescription": {
        "TableStatus": "DELETING",
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableName": "MusicCollection",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "WriteCapacityUnits": 5,
            "ReadCapacityUnits": 5
        }
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的刪除資料表。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteTable。

describe-backup

以下程式碼範例顯示如何使用 describe-backup。

AWS CLI

取得資料表現有備份的相關資訊

下列describe-backup範例顯示指定現有備份的相關資訊。

```
aws dynamodb describe-backup \
    --backup-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/
backup/01576616366715-b4e58d3a
```

輸出：

```
{
    "BackupDescription": {
        "BackupDetails": {
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01576616366715-b4e58d3a",
            "BackupName": "MusicCollectionBackup",
            "BackupSizeBytes": 0,
            "BackupStatus": "AVAILABLE",
            "BackupType": "USER",
            "BackupCreationDateTime": 1576616366.715
        },
        "SourceTableDetails": {
            "TableName": "MusicCollection",
            "TableId": "b0c04bcc-309b-4352-b2ae-9088af169fe2",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "TableSizeBytes": 0,
            "KeySchema": [
                {
                    "AttributeName": "Artist",
                    "KeyType": "HASH"
                },
                {
                    "AttributeName": "SongTitle",
                    "KeyType": "RANGE"
                }
            ],
            "TableCreationDateTime": 1576615228.571,
            "ProvisionedThroughput": {
                "ReadCapacityUnits": 5,
                "WriteCapacityUnits": 5
```

```
            },
            "ItemCount": 0,
            "BillingMode": "PROVISIONED"
        },
        "SourceTableFeatureDetails": {}
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 的隨需備份與還原。 DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeBackup。

`describe-continuous-backups`

以下程式碼範例顯示如何使用 `describe-continuous-backups`。

AWS CLI

取得 DynamoDB 資料表持續備份的相關資訊

下列`describe-continuous-backups`範例顯示`MusicCollection`資料表連續備份設定的詳細資訊。

```
aws dynamodb describe-continuous-backups \
    --table-name MusicCollection
```

輸出：

```
{
    "ContinuousBackupsDescription": {
        "ContinuousBackupsStatus": "ENABLED",
        "PointInTimeRecoveryDescription": {
            "PointInTimeRecoveryStatus": "DISABLED"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB Point-in-Time復原。 DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeContinuousBackups。

describe-contributor-insights

以下程式碼範例顯示如何使用 describe-contributor-insights。

AWS CLI

檢視 DynamoDB 資料表的 Contributor Insights 設定

下列describe-contributor-insights範例顯示MusicCollection資料表的 Contributor Insights 設定和AlbumTitle-index全域次要索引。

```
aws dynamodb describe-contributor-insights \
    --table-name MusicCollection \
    --index-name AlbumTitle-index
```

輸出：

```
{
    "TableName": "MusicCollection",
    "IndexName": "AlbumTitle-index",
    "ContributorInsightsRuleList": [
        "DynamoDBContributorInsights-PKC-MusicCollection-1576629651520",
        "DynamoDBContributorInsights-SKC-MusicCollection-1576629651520",
        "DynamoDBContributorInsights-PKT-MusicCollection-1576629651520",
        "DynamoDBContributorInsights-SKT-MusicCollection-1576629651520"
    ],
    "ContributorInsightsStatus": "ENABLED",
    "LastUpdateDateTime": 1576629654.78
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的使用適用於 DynamoDB 的 CloudWatch Contributor Insights 分析資料存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeContributorInsights。

describe-endpoints

以下程式碼範例顯示如何使用 describe-endpoints。

AWS CLI

檢視區域端點資訊

下列describe-endpoints範例顯示目前 AWS 區域端點的詳細資訊。

```
aws dynamodb describe-endpoints
```

輸出：

```
{
    "Endpoints": [
        {
            "Address": "dynamodb.us-west-2.amazonaws.com",
            "CachePeriodInMinutes": 1440
        }
    ]
}
```

如需詳細資訊，請參閱《 AWS 一般參考》中的 Amazon DynamoDB 端點和配額。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEndpoints。

describe-global-table-settings

以下程式碼範例顯示如何使用 describe-global-table-settings。

AWS CLI

取得 DynamoDB 全域資料表設定的相關資訊

下列describe-global-table-settings範例顯示MusicCollection全域資料表的設定。

```
aws dynamodb describe-global-table-settings \
    --global-table-name MusicCollection
```

輸出：

```
{
    "GlobalTableName": "MusicCollection",
    "ReplicaSettings": [
        {
            "RegionName": "us-east-1",
            "ReplicaStatus": "ACTIVE",
            "ReplicaProvisionedReadCapacityUnits": 10,
            "ReplicaProvisionedReadCapacityAutoScalingSettings": {
```

```
                "AutoScalingDisabled": true
            },
            "ReplicaProvisionedWriteCapacityUnits": 5,
            "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            }
        },
        {
            "RegionName": "us-east-2",
            "ReplicaStatus": "ACTIVE",
            "ReplicaProvisionedReadCapacityUnits": 10,
            "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            },
            "ReplicaProvisionedWriteCapacityUnits": 5,
            "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 全域資料表。DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeGlobalTableSettings。

describe-global-table

以下程式碼範例顯示如何使用 describe-global-table。

AWS CLI

顯示 DynamoDB 全域資料表的相關資訊

下列describe-global-table範例顯示MusicCollection全域資料表的詳細資訊。

```
aws dynamodb describe-global-table \
    --global-table-name MusicCollection
```

輸出：

```
{
```

```
    "GlobalTableDescription": {
        "ReplicationGroup": [
            {
                "RegionName": "us-east-2"
            },
            {
                "RegionName": "us-east-1"
            }
        ],
        "GlobalTableArn": "arn:aws:dynamodb::123456789012:global-table/
MusicCollection",
        "CreationDateTime": 1576625818.532,
        "GlobalTableStatus": "ACTIVE",
        "GlobalTableName": "MusicCollection"
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 全域資料表。DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeGlobalTable。

describe-limits

以下程式碼範例顯示如何使用 `describe-limits`。

AWS CLI

檢視佈建容量限制

下列`describe-limits`範例顯示目前 AWS 區域中您帳戶的佈建容量限制。

```
aws dynamodb describe-limits
```

輸出：

```
{
    "AccountMaxReadCapacityUnits": 80000,
    "AccountMaxWriteCapacityUnits": 80000,
    "TableMaxReadCapacityUnits": 40000,
    "TableMaxWriteCapacityUnits": 40000
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 中的限制。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLimits。

describe-table-replica-auto-scaling

以下程式碼範例顯示如何使用 describe-table-replica-auto-scaling。

AWS CLI

檢視全域資料表複本之間的自動擴展設定

下列describe-table-replica-auto-scaling範例顯示MusicCollection全域資料表複本之間的自動擴展設定。

```
aws dynamodb describe-table-replica-auto-scaling \
    --table-name MusicCollection
```

輸出：

```
{
    "TableAutoScalingDescription": {
        "TableName": "MusicCollection",
        "TableStatus": "ACTIVE",
        "Replicas": [
            {
                "RegionName": "us-east-1",
                "GlobalSecondaryIndexes": [],
                "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                    "MinimumUnits": 5,
                    "MaximumUnits": 40000,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 70.0
                            }
                        }
```

```
                    ]
                },
                "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                    "MinimumUnits": 5,
                    "MaximumUnits": 40000,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 70.0
                            }
                        }
                    ]
                },
                "ReplicaStatus": "ACTIVE"
            },
            {
                "RegionName": "us-east-2",
                "GlobalSecondaryIndexes": [],
                "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                    "MinimumUnits": 5,
                    "MaximumUnits": 40000,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 70.0
                            }
                        }
                    ]
                },
                "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                    "MinimumUnits": 5,
                    "MaximumUnits": 40000,
```

```
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 70.0
                            }
                        }
                    ]
                },
                "ReplicaStatus": "ACTIVE"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 全域資料表。DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTableReplicaAutoScaling。

describe-table

以下程式碼範例顯示如何使用 describe-table。

AWS CLI

描述資料表

下列describe-table範例說明 MusicCollection資料表。

```
aws dynamodb describe-table \
    --table-name MusicCollection
```

輸出：

```
{
    "Table": {
        "AttributeDefinitions": [
            {
```

```
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "WriteCapacityUnits": 5,
            "ReadCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "TableName": "MusicCollection",
        "TableStatus": "ACTIVE",
        "KeySchema": [
            {
                "KeyType": "HASH",
                "AttributeName": "Artist"
            },
            {
                "KeyType": "RANGE",
                "AttributeName": "SongTitle"
            }
        ],
        "ItemCount": 0,
        "CreationDateTime": 1421866952.062
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的描述資料表。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeTable。

describe-time-to-live

以下程式碼範例顯示如何使用 describe-time-to-live。

AWS CLI

檢視資料表的存留時間設定

下列describe-time-to-live範例顯示MusicCollection資料表的存留時間設定。

```
aws dynamodb describe-time-to-live \
    --table-name MusicCollection
```

輸出：

```
{
    "TimeToLiveDescription": {
        "TimeToLiveStatus": "ENABLED",
        "AttributeName": "ttl"
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的存留時間。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTimeToLive。

get-item

以下程式碼範例顯示如何使用 get-item。

AWS CLI

範例 1：讀取資料表中的項目

下列get-item範例會從 MusicCollection 資料表擷取項目。資料表具有hash-and-range主索引鍵 (Artist 和 SongTitle)，因此您必須指定這兩個屬性。命令也會請求操作所耗用讀取容量的相關資訊。

```
aws dynamodb get-item \
    --table-name MusicCollection \
    --key file://key.json \
    --return-consumed-capacity TOTAL
```

key.json 的內容：

```
{
    "Artist": {"S": "Acme Band"},
    "SongTitle": {"S": "Happy Day"}
}
```

輸出：

```
{
    "Item": {
        "AlbumTitle": {
            "S": "Songs About Life"
        },
        "SongTitle": {
            "S": "Happy Day"
        },
        "Artist": {
            "S": "Acme Band"
        }
    },
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 0.5
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的讀取項目。

範例 2：使用一致讀取讀取項目

下列範例使用強式一致讀取從MusicCollection資料表擷取項目。

```
aws dynamodb get-item \
    --table-name MusicCollection \
    --key file://key.json \
    --consistent-read \
    --return-consumed-capacity TOTAL
```

key.json 的內容：

```
{
    "Artist": {"S": "Acme Band"},
    "SongTitle": {"S": "Happy Day"}
}
```

輸出：

```
{
    "Item": {
        "AlbumTitle": {
```

```
            "S": "Songs About Life"
        },
        "SongTitle": {
            "S": "Happy Day"
        },
        "Artist": {
            "S": "Acme Band"
        }
    },
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 1.0
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的讀取項目。

範例 3：擷取項目的特定屬性

下列範例使用投影表達式，僅擷取所需項目的三個屬性。

```
aws dynamodb get-item \
    --table-name ProductCatalog \
    --key '{"Id": {"N": "102"}}' \
    --projection-expression "#T, #C, #P" \
    --expression-attribute-names file://names.json
```

names.json 的內容：

```
{
    "#T": "Title",
    "#C": "ProductCategory",
    "#P": "Price"
}
```

輸出：

```
{
    "Item": {
        "Price": {
            "N": "20"
        },
        "Title": {
```

```
            "S": "Book 102 Title"
        },
        "ProductCategory": {
            "S": "Book"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的讀取項目。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetItem。

list-backups

以下程式碼範例顯示如何使用 `list-backups`。

AWS CLI

範例 1：列出所有現有的 DynamoDB 備份

下列`list-backups`範例列出所有現有的備份。

```
aws dynamodb list-backups
```

輸出：

```
{
    "BackupSummaries": [
        {
            "TableName": "MusicCollection",
            "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-a1bcd234",
            "BackupName": "MusicCollectionBackup1",
            "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
            "BackupStatus": "AVAILABLE",
            "BackupType": "USER",
            "BackupSizeBytes": 170
        },
        {
            "TableName": "MusicCollection",
```

```
            "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-b2abc345",
            "BackupName": "MusicCollectionBackup2",
            "BackupCreationDateTime": "2020-06-26T11:08:35.431000-07:00",
            "BackupStatus": "AVAILABLE",
            "BackupType": "USER",
            "BackupSizeBytes": 400
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 的隨需備份與還原。 DynamoDB

範例 2：列出使用者在特定時間範圍內建立的備份

下列範例只會列出由使用者（非由 DynamoDB 自動建立的）建立日期介於 2020 年 1 月 1 日至 2020 年 3 月 1 日的`MusicCollection`資料表備份。

```
aws dynamodb list-backups \
    --table-name MusicCollection \
    --time-range-lower-bound 1577836800 \
    --time-range-upper-bound 1583020800 \
    --backup-type USER
```

輸出：

```
{
    "BackupSummaries": [
        {
            "TableName": "MusicCollection",
            "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-a1bcd234",
            "BackupName": "MusicCollectionBackup1",
            "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
            "BackupStatus": "AVAILABLE",
            "BackupType": "USER",
```

```
            "BackupSizeBytes": 170
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 的隨需備份與還原。 DynamoDB

範例 3：限制頁面大小

下列範例會傳回所有現有備份的清單，但每個呼叫只會擷取一個項目，並視需要執行多個呼叫以取得整個清單。在大量資源上執行清單命令時，限制頁面大小非常有用，這可能會導致使用預設頁面大小 1000 時發生「逾時」錯誤。

```
aws dynamodb list-backups \
    --page-size 1
```

輸出：

```
{
    "BackupSummaries": [
        {
            "TableName": "MusicCollection",
            "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-a1bcd234",
            "BackupName": "MusicCollectionBackup1",
            "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
            "BackupStatus": "AVAILABLE",
            "BackupType": "USER",
            "BackupSizeBytes": 170
        },
        {
            "TableName": "MusicCollection",
            "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-b2abc345",
            "BackupName": "MusicCollectionBackup2",
```

```
            "BackupCreationDateTime": "2020-06-26T11:08:35.431000-07:00",
            "BackupStatus": "AVAILABLE",
            "BackupType": "USER",
            "BackupSizeBytes": 400
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 的隨需備份與還原。 DynamoDB

範例 4：限制傳回的項目數量

下列範例會將傳回的項目數限制為 1。回應包含要用來擷取結果下一頁NextToken的值。

```
aws dynamodb list-backups \
    --max-items 1
```

輸出：

```
{
    "BackupSummaries": [
        {
            "TableName": "MusicCollection",
            "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-a1bcd234",
            "BackupName": "MusicCollectionBackup1",
            "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
            "BackupStatus": "AVAILABLE",
            "BackupType": "USER",
            "BackupSizeBytes": 170
        }
    ],
    "NextToken":
 "abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 的隨需備份與還原。 DynamoDB

範例 5：擷取下一頁的結果

下列命令會使用先前呼叫list-backups命令的 NextToken值來擷取另一個結果頁面。由於此案例中的回應不包含NextToken值，我們知道我們已經到達結果的結尾。

```
aws dynamodb list-backups \
    --starting-
token abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

輸出

```
{
    "BackupSummaries": [
        {
            "TableName": "MusicCollection",
            "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-b2abc345",
            "BackupName": "MusicCollectionBackup2",
            "BackupCreationDateTime": "2020-06-26T11:08:35.431000-07:00",
            "BackupStatus": "AVAILABLE",
            "BackupType": "USER",
            "BackupSizeBytes": 400
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 的隨需備份與還原。 DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBackups。

list-contributor-insights

以下程式碼範例顯示如何使用 list-contributor-insights。

AWS CLI

範例 1：檢視 Contributor Insights 摘要清單

下列list-contributor-insights範例顯示 Contributor Insights 摘要的清單。

```
aws dynamodb list-contributor-insights
```

輸出：

```
{
    "ContributorInsightsSummaries": [
        {
            "TableName": "MusicCollection",
            "IndexName": "AlbumTitle-index",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "ProductCatalog",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "Forum",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "Reply",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "Thread",
            "ContributorInsightsStatus": "ENABLED"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的使用適用於 DynamoDB 的 CloudWatch Contributor Insights 分析資料存取。

範例 2：限制傳回的項目數量

下列範例會將傳回的項目數限制為 4。回應包含要用來擷取結果下一頁NextToken的值。

```
aws dynamodb list-contributor-insights \
    --max-results 4
```

輸出：

```
{
    "ContributorInsightsSummaries": [
        {
            "TableName": "MusicCollection",
            "IndexName": "AlbumTitle-index",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "ProductCatalog",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "Forum",
            "ContributorInsightsStatus": "ENABLED"
        }
    ],
    "NextToken":
 "abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的使用適用於 DynamoDB 的 CloudWatch Contributor Insights 分析資料存取。 DynamoDB

範例 3：擷取下一頁的結果

下列命令會使用先前呼叫`list-contributor-insights`命令`NextToken`的值來擷取另一個結果頁面。由於此案例中的回應不包含`NextToken`值，我們知道我們已經到達結果的結尾。

```
aws dynamodb list-contributor-insights \
    --max-results 4 \
    --next-
token abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

輸出：

```
{
    "ContributorInsightsSummaries": [
        {
            "TableName": "Reply",
            "ContributorInsightsStatus": "ENABLED"
```

```
        },
        {
            "TableName": "Thread",
            "ContributorInsightsStatus": "ENABLED"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的使用適用於 DynamoDB 的 CloudWatch Contributor Insights 分析資料存取。 DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListContributorInsights。

`list-global-tables`

以下程式碼範例顯示如何使用 `list-global-tables`。

AWS CLI

列出現有的 DynamoDB 全域資料表

下列`list-global-tables`範例列出所有現有的全域資料表。

```
aws dynamodb list-global-tables
```

輸出：

```
{
    "GlobalTables": [
        {
            "GlobalTableName": "MusicCollection",
            "ReplicationGroup": [
                {
                    "RegionName": "us-east-2"
                },
                {
                    "RegionName": "us-east-1"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 全域資料表。
DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListGlobalTables。

`list-tables`

以下程式碼範例顯示如何使用 `list-tables`。

AWS CLI

範例 1：列出資料表

下列`list-tables`範例列出與目前 AWS 帳戶和區域相關聯的所有資料表。

```
aws dynamodb list-tables
```

輸出：

```
{
    "TableNames": [
        "Forum",
        "ProductCatalog",
        "Reply",
        "Thread"
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的列出資料表名稱。

範例 2：限制頁面大小

下列範例會傳回所有現有資料表的清單，但每次呼叫只會擷取一個項目，並視需要執行多個呼叫以取得整個清單。在大量資源上執行清單命令時，限制頁面大小非常有用，這可能會導致使用預設頁面大小 1000 時發生「逾時」錯誤。

```
aws dynamodb list-tables \
    --page-size 1
```

輸出：

```
{
```

```
    "TableNames": [
        "Forum",
        "ProductCatalog",
        "Reply",
        "Thread"
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的列出資料表名稱。

範例 3：限制傳回的項目數量

下列範例會將傳回的項目數限制為 2。回應包含要用來擷取結果下一頁NextToken的值。

```
aws dynamodb list-tables \
    --max-items 2
```

輸出：

```
{
    "TableNames": [
        "Forum",
        "ProductCatalog"
    ],
    "NextToken":
 "abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的列出資料表名稱。

範例 4：擷取下一頁的結果

下列命令會使用先前呼叫list-tables命令NextToken的值來擷取另一個結果頁面。由於此案例中的回應不包含NextToken值，我們知道我們已經到達結果的結尾。

```
aws dynamodb list-tables \
    --starting-
token abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

輸出：

```
{
```

```
    "TableNames": [
        "Reply",
        "Thread"
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的列出資料表名稱。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTables。

`list-tags-of-resource`

以下程式碼範例顯示如何使用 `list-tags-of-resource`。

AWS CLI

範例 1：列出 DynamoDB 資源的標籤

下列`list-tags-of-resource`範例顯示`MusicCollection`資料表的標籤。

```
aws dynamodb list-tags-of-resource \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection
```

輸出：

```
{
    "Tags": [
        {
            "Key": "Owner",
            "Value": "blueTeam"
        },
        {
            "Key": "Environment",
            "Value": "Production"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 的標記。
DynamoDB

範例 2：限制傳回的標籤數量

下列範例會將傳回的標籤數量限制為 1。回應包含要用來擷取結果下一頁NextToken的值。

```
aws dynamodb list-tags-of-resource \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --max-items 1
```

輸出：

```
{
    "Tags": [
        {
            "Key": "Owner",
            "Value": "blueTeam"
        }
    ],
    "NextToken":
 "abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 的標記。DynamoDB

範例 3：擷取下一頁的結果

下列命令會使用先前呼叫list-tags-of-resource命令的 NextToken值來擷取另一個結果頁面。由於此案例中的回應不包含NextToken值，我們知道我們已經到達結果的結尾。

```
aws dynamodb list-tags-of-resource \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --starting-
token abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

輸出：

```
{
    "Tags": [
        {
            "Key": "Environment",
            "Value": "Production"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 的標記。DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTagsOfResource。

put-item

以下程式碼範例顯示如何使用 put-item。

AWS CLI

範例 1：將項目新增至資料表

下列put-item範例會將新項目新增至 MusicCollection 資料表。

```
aws dynamodb put-item \
    --table-name MusicCollection \
    --item file://item.json \
    --return-consumed-capacity TOTAL \
    --return-item-collection-metrics SIZE
```

item.json 的內容：

```
{
    "Artist": {"S": "No One You Know"},
    "SongTitle": {"S": "Call Me Today"},
    "AlbumTitle": {"S": "Greatest Hits"}
}
```

輸出：

```
{
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 1.0
    },
    "ItemCollectionMetrics": {
        "ItemCollectionKey": {
            "Artist": {
                "S": "No One You Know"
```

```
            }
        },
        "SizeEstimateRangeGB": [
            0.0,
            1.0
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的撰寫項目。

範例 2：有條件覆寫資料表中的項目

下列put-item範例只有在現有項目具有值為 的 AlbumTitle 屬性時，才會覆寫MusicCollection資料表中的現有項目Greatest Hits。命令會傳回項目的上一個值。

```
aws dynamodb put-item \
    --table-name MusicCollection \
    --item file://item.json \
    --condition-expression "#A = :A" \
    --expression-attribute-names file://names.json \
    --expression-attribute-values file://values.json \
    --return-values ALL_OLD
```

item.json 的內容：

```
{
    "Artist": {"S": "No One You Know"},
    "SongTitle": {"S": "Call Me Today"},
    "AlbumTitle": {"S": "Somewhat Famous"}
}
```

names.json 的內容：

```
{
    "#A": "AlbumTitle"
}
```

values.json 的內容：

```
{
    ":A": {"S": "Greatest Hits"}
```

```
}
```

輸出：

```
{
    "Attributes": {
        "AlbumTitle": {
            "S": "Greatest Hits"
        },
        "Artist": {
            "S": "No One You Know"
        },
        "SongTitle": {
            "S": "Call Me Today"
        }
    }
}
```

如果金鑰已存在，您應該會看到下列輸出：

```
A client error (ConditionalCheckFailedException) occurred when calling the PutItem
 operation: The conditional request failed.
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的撰寫項目。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutObject。

query

以下程式碼範例顯示如何使用 query。

AWS CLI

範例 1：查詢資料表

下列query範例會查詢 MusicCollection 資料表中的項目。資料表具有hash-and-range主索引鍵 (Artist 和 SongTitle)，但此查詢只會指定雜湊索引鍵值。它傳回名為 "No One You Know" 的藝術家的歌曲標題。

```
aws dynamodb query \
    --table-name MusicCollection \
    --projection-expression "SongTitle" \
```

```
    --key-condition-expression "Artist = :v1" \
    --expression-attribute-values file://expression-attributes.json \
    --return-consumed-capacity TOTAL
```

expression-attributes.json 的內容：

```
{
    ":v1": {"S": "No One You Know"}
}
```

輸出：

```
{
    "Items": [
        {
            "SongTitle": {
                "S": "Call Me Today"
            },
            "SongTitle": {
                "S": "Scared of My Shadow"
            }
        }
    ],
    "Count": 2,
    "ScannedCount": 2,
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 0.5
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的在 DynamoDB 中使用查詢。DynamoDB

範例 2：使用強式一致讀取查詢資料表，並以遞減順序周遊索引

下列範例會執行與第一個範例相同的查詢，但會以相反的順序傳回結果，並使用強式一致讀取。

```
aws dynamodb query \
    --table-name MusicCollection \
    --projection-expression "SongTitle" \
    --key-condition-expression "Artist = :v1" \
    --expression-attribute-values file://expression-attributes.json \
```

```
    --consistent-read \
    --no-scan-index-forward \
    --return-consumed-capacity TOTAL
```

expression-attributes.json 的內容：

```
{
    ":v1": {"S": "No One You Know"}
}
```

輸出：

```
{
    "Items": [
        {
            "SongTitle": {
                "S": "Scared of My Shadow"
            }
        },
        {
            "SongTitle": {
                "S": "Call Me Today"
            }
        }
    ],
    "Count": 2,
    "ScannedCount": 2,
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 1.0
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的在 DynamoDB 中使用查詢。DynamoDB

範例 3：篩選掉特定結果

下列範例會查詢 ，MusicCollection但會排除 AlbumTitle 屬性中具有特定值的結果。請注意，這不會影響 ScannedCount或 ConsumedCapacity，因為篩選條件會在項目讀取後套用。

```
aws dynamodb query \
```

```
    --table-name MusicCollection \
    --key-condition-expression "#n1 = :v1" \
    --filter-expression "NOT (#n2 IN (:v2, :v3))" \
    --expression-attribute-names file://names.json \
    --expression-attribute-values file://values.json \
    --return-consumed-capacity TOTAL
```

values.json 的內容：

```
{
    ":v1": {"S": "No One You Know"},
    ":v2": {"S": "Blue Sky Blues"},
    ":v3": {"S": "Greatest Hits"}
}
```

names.json 的內容：

```
{
    "#n1": "Artist",
    "#n2": "AlbumTitle"
}
```

輸出：

```
{
    "Items": [
        {
            "AlbumTitle": {
                "S": "Somewhat Famous"
            },
            "Artist": {
                "S": "No One You Know"
            },
            "SongTitle": {
                "S": "Call Me Today"
            }
        }
    ],
    "Count": 1,
    "ScannedCount": 2,
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
```

```
        "CapacityUnits": 0.5
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的在 DynamoDB 中使用查詢。DynamoDB

範例 4：僅擷取項目計數

下列範例會擷取符合查詢的項目計數，但不會自行擷取任何項目。

```
aws dynamodb query \
    --table-name MusicCollection \
    --select COUNT \
    --key-condition-expression "Artist = :v1" \
    --expression-attribute-values file://expression-attributes.json
```

expression-attributes.json 的內容：

```
{
    ":v1": {"S": "No One You Know"}
}
```

輸出：

```
{
    "Count": 2,
    "ScannedCount": 2,
    "ConsumedCapacity": null
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的在 DynamoDB 中使用查詢。DynamoDB

範例 5：查詢索引

下列範例會查詢本機次要索引 AlbumTitleIndex。查詢會從已投影至本機次要索引的基礎資料表傳回所有屬性。請注意，查詢本機次要索引或全域次要索引時，您還必須使用 table-name 參數提供基底資料表的名稱。

```
aws dynamodb query \
    --table-name MusicCollection \
```

```
    --index-name AlbumTitleIndex \
    --key-condition-expression "Artist = :v1" \
    --expression-attribute-values file://expression-attributes.json \
    --select ALL_PROJECTED_ATTRIBUTES \
    --return-consumed-capacity INDEXES
```

expression-attributes.json 的內容：

```
{
    ":v1": {"S": "No One You Know"}
}
```

輸出：

```
{
    "Items": [
        {
            "AlbumTitle": {
                "S": "Blue Sky Blues"
            },
            "Artist": {
                "S": "No One You Know"
            },
            "SongTitle": {
                "S": "Scared of My Shadow"
            }
        },
        {
            "AlbumTitle": {
                "S": "Somewhat Famous"
            },
            "Artist": {
                "S": "No One You Know"
            },
            "SongTitle": {
                "S": "Call Me Today"
            }
        }
    ],
    "Count": 2,
    "ScannedCount": 2,
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
```

```
        "CapacityUnits": 0.5,
        "Table": {
            "CapacityUnits": 0.0
        },
        "LocalSecondaryIndexes": {
            "AlbumTitleIndex": {
                "CapacityUnits": 0.5
            }
        }
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的在 DynamoDB 中使用查詢。DynamoDB

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 Query。

`restore-table-from-backup`

以下程式碼範例顯示如何使用 `restore-table-from-backup`。

AWS CLI

從現有備份還原 DynamoDB 資料表

下列`restore-table-from-backup`範例會從現有的備份還原指定的資料表。

```
aws dynamodb restore-table-from-backup \
    --target-table-name MusicCollection \
    --backup-arnarn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/
backup/01576616366715-b4e58d3a
```

輸出：

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
```

```
                "AttributeType": "S"
            }
        ],
        "TableName": "MusicCollection2",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": 1576618274.326,
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection2",
        "TableId": "114865c9-5ef3-496c-b4d1-c4cbdd2d44fb",
        "BillingModeSummary": {
            "BillingMode": "PROVISIONED"
        },
        "RestoreSummary": {
            "SourceBackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01576616366715-b4e58d3a",
            "SourceTableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "RestoreDateTime": 1576616366.715,
            "RestoreInProgress": true
        }
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 的隨需備份與還原。 DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreTableFromBackup。

restore-table-to-point-in-time

以下程式碼範例顯示如何使用 restore-table-to-point-in-time。

AWS CLI

將 DynamoDB 資料表還原至某個時間點

下列restore-table-to-point-in-time範例會將MusicCollection資料表還原至指定的時間點。

```
aws dynamodb restore-table-to-point-in-time \
    --source-table-name MusicCollection \
    --target-table-name MusicCollectionRestore \
    --restore-date-time 1576622404.0
```

輸出：

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "TableName": "MusicCollectionRestore",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": 1576623311.86,
```

```
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollectionRestore",
        "TableId": "befd9e0e-1843-4dc6-a147-d6d00e85cb1f",
        "BillingModeSummary": {
            "BillingMode": "PROVISIONED"
        },
        "RestoreSummary": {
            "SourceTableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
            "RestoreDateTime": 1576622404.0,
            "RestoreInProgress": true
        }
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB Point-in-Time復原。 DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreTableToPointInTime。

scan

以下程式碼範例顯示如何使用 scan。

AWS CLI

掃描資料表

下列scan範例會掃描整個MusicCollection資料表，然後將結果縮小為演出者「不知道」的歌曲。對於每個項目，只會傳回專輯標題和歌曲標題。

```
aws dynamodb scan \
    --table-name MusicCollection \
    --filter-expression "Artist = :a" \
    --projection-expression "#ST, #AT" \
```

```
    --expression-attribute-names file://expression-attribute-names.json \
    --expression-attribute-values file://expression-attribute-values.json
```

expression-attribute-names.json 的內容：

```
{
    "#ST": "SongTitle",
    "#AT":"AlbumTitle"
}
```

expression-attribute-values.json 的內容：

```
{
    ":a": {"S": "No One You Know"}
}
```

輸出：

```
{
    "Count": 2,
    "Items": [
        {
            "SongTitle": {
                "S": "Call Me Today"
            },
            "AlbumTitle": {
                "S": "Somewhat Famous"
            }
        },
        {
            "SongTitle": {
                "S": "Scared of My Shadow"
            },
            "AlbumTitle": {
                "S": "Blue Sky Blues"
            }
        }
    ],
    "ScannedCount": 3,
    "ConsumedCapacity": null
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的在 DynamoDB 中使用掃描。DynamoDB

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 Scan。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤新增至 DynamoDB 資源

下列tag-resource範例會將標籤鍵/值對新增至MusicCollection資料表。

```
aws dynamodb tag-resource \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --tags Key=Owner,Value=blueTeam
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 的標記。DynamoDB

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

transact-get-items

以下程式碼範例顯示如何使用 transact-get-items。

AWS CLI

從一或多個資料表以原子方式擷取多個項目

下列transact-get-items範例會以原子方式擷取多個項目。

```
aws dynamodb transact-get-items \
    --transact-items file://transact-items.json \
    --return-consumed-capacity TOTAL
```

transact-items.json 的內容：

```
[
    {
        "Get": {
            "Key": {
                "Artist": {"S": "Acme Band"},
                "SongTitle": {"S": "Happy Day"}
            },
            "TableName": "MusicCollection"
        }
    },
    {
        "Get": {
            "Key": {
                "Artist": {"S": "No One You Know"},
                "SongTitle": {"S": "Call Me Today"}
            },
            "TableName": "MusicCollection"
        }
    }
]
```

輸出：

```
{
    "ConsumedCapacity": [
        {
            "TableName": "MusicCollection",
            "CapacityUnits": 4.0,
            "ReadCapacityUnits": 4.0
        }
    ],
    "Responses": [
        {
            "Item": {
                "AlbumTitle": {
                    "S": "Songs About Life"
                },
                "Artist": {
                    "S": "Acme Band"
                },
                "SongTitle": {
                    "S": "Happy Day"
                }
```

```
            }
        },
        {
            "Item": {
                "AlbumTitle": {
                    "S": "Somewhat Famous"
                },
                "Artist": {
                    "S": "No One You Know"
                },
                "SongTitle": {
                    "S": "Call Me Today"
                }
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的使用 DynamoDB 交易管理複雜工作流程。 DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TransactGetItems。

`transact-write-items`

以下程式碼範例顯示如何使用 `transact-write-items`。

AWS CLI

範例 1：以原子方式將項目寫入一或多個資料表

下列`transact-write-items`範例會更新一個項目並刪除另一個項目。如果任一操作失敗，或任一項目包含`Rating`屬性，則操作會失敗。

```
aws dynamodb transact-write-items \
    --transact-items file://transact-items.json \
    --return-consumed-capacity TOTAL \
    --return-item-collection-metrics SIZE
```

`transact-items.json` 檔案的內容：

```
[
```

```
    {
        "Update": {
            "Key": {
                "Artist": {"S": "Acme Band"},
                "SongTitle": {"S": "Happy Day"}
            },
            "UpdateExpression": "SET AlbumTitle = :newval",
            "ExpressionAttributeValues": {
                ":newval": {"S": "Updated Album Title"}
            },
            "TableName": "MusicCollection",
            "ConditionExpression": "attribute_not_exists(Rating)"
        }
    },
    {
        "Delete": {
            "Key": {
                "Artist": {"S": "No One You Know"},
                "SongTitle": {"S": "Call Me Today"}
            },
            "TableName": "MusicCollection",
            "ConditionExpression": "attribute_not_exists(Rating)"
        }
    }
]
```

輸出：

```
{
    "ConsumedCapacity": [
        {
            "TableName": "MusicCollection",
            "CapacityUnits": 10.0,
            "WriteCapacityUnits": 10.0
        }
    ],
    "ItemCollectionMetrics": {
        "MusicCollection": [
            {
                "ItemCollectionKey": {
                    "Artist": {
                        "S": "No One You Know"
                    }
```

```
                },
                "SizeEstimateRangeGB": [
                    0.0,
                    1.0
                ]
            },
            {
                "ItemCollectionKey": {
                    "Artist": {
                        "S": "Acme Band"
                    }
                },
                "SizeEstimateRangeGB": [
                    0.0,
                    1.0
                ]
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的使用 DynamoDB 交易管理複雜工作流程。 DynamoDB

範例 2：使用用戶端請求字符以原子方式寫入項目

下列命令使用用戶端請求字符對`transact-write-items`等冪進行呼叫，這表示多個呼叫的效果與單一呼叫相同。

```
aws dynamodb transact-write-items \
    --transact-items file://transact-items.json \
    --client-request-token abc123
```

`transact-items.json` 檔案的內容：

```
[
    {
        "Update": {
            "Key": {
                "Artist": {"S": "Acme Band"},
                "SongTitle": {"S": "Happy Day"}
            },
```

```
            "UpdateExpression": "SET AlbumTitle = :newval",
            "ExpressionAttributeValues": {
                ":newval": {"S": "Updated Album Title"}
            },
            "TableName": "MusicCollection",
            "ConditionExpression": "attribute_not_exists(Rating)"
        }
    },
    {
        "Delete": {
            "Key": {
                "Artist": {"S": "No One You Know"},
                "SongTitle": {"S": "Call Me Today"}
            },
            "TableName": "MusicCollection",
            "ConditionExpression": "attribute_not_exists(Rating)"
        }
    }
]
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的使用 DynamoDB 交易管理複雜工作流程。 DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TransactWriteItems。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從 DynamoDB 資源移除標籤

下列untag-resource範例會從MusicCollection資料表中移除具有 金鑰Owner的標籤。

```
aws dynamodb untag-resource \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --tag-keys Owner
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 的標記。 DynamoDB

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-continuous-backups

以下程式碼範例顯示如何使用 update-continuous-backups。

AWS CLI

更新 DynamoDB 資料表的連續備份設定

下列update-continuous-backups範例會啟用MusicCollection資料表的point-in-time復原。

```
aws dynamodb update-continuous-backups \
    --table-name MusicCollection \
    --point-in-time-recovery-specification PointInTimeRecoveryEnabled=true
```

輸出：

```
{
    "ContinuousBackupsDescription": {
        "ContinuousBackupsStatus": "ENABLED",
        "PointInTimeRecoveryDescription": {
            "PointInTimeRecoveryStatus": "ENABLED",
            "EarliestRestorableDateTime": 1576622404.0,
            "LatestRestorableDateTime": 1576622404.0
        }
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB Point-in-Time復原。 DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateContinuousBackups。

update-contributor-insights

以下程式碼範例顯示如何使用 update-contributor-insights。

AWS CLI

在資料表上啟用 Contributor Insights

下列update-contributor-insights範例會在MusicCollection資料表和AlbumTitle-index全域次要索引上啟用 Contributor Insights。

```
aws dynamodb update-contributor-insights \
    --table-name MusicCollection \
    --index-name AlbumTitle-index \
    --contributor-insights-action ENABLE
```

輸出：

```
{
    "TableName": "MusicCollection",
    "IndexName": "AlbumTitle-index",
    "ContributorInsightsStatus": "ENABLING"
}
```

如需詳細資訊，請參閱《Amazon DynamoDB DynamoDB 開發人員指南》中的使用適用於 DynamoDB 的 CloudWatch Contributor Insights 分析資料存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateContributorInsights。

update-global-table-settings

以下程式碼範例顯示如何使用 update-global-table-settings。

AWS CLI

更新 DynamoDB 全域資料表上的佈建寫入容量設定

下列update-global-table-settings範例會將MusicCollection全域資料表的佈建寫入容量設定為 15。

```
aws dynamodb update-global-table-settings \
    --global-table-name MusicCollection \
    --global-table-provisioned-write-capacity-units 15
```

輸出：

```
{
    "GlobalTableName": "MusicCollection",
    "ReplicaSettings": [
        {
            "RegionName": "eu-west-1",
            "ReplicaStatus": "UPDATING",
            "ReplicaProvisionedReadCapacityUnits": 10,
            "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            },
            "ReplicaProvisionedWriteCapacityUnits": 10,
            "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            }
        },
        {
            "RegionName": "us-east-1",
            "ReplicaStatus": "UPDATING",
            "ReplicaProvisionedReadCapacityUnits": 10,
            "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            },
            "ReplicaProvisionedWriteCapacityUnits": 10,
            "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            }
        },
        {
            "RegionName": "us-east-2",
            "ReplicaStatus": "UPDATING",
            "ReplicaProvisionedReadCapacityUnits": 10,
            "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            },
            "ReplicaProvisionedWriteCapacityUnits": 10,
            "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                "AutoScalingDisabled": true
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 全域資料表。DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateGlobalTableSettings。

update-global-table

以下程式碼範例顯示如何使用 update-global-table。

AWS CLI

更新 DynamoDB 全域資料表

下列update-global-table範例會將指定區域中的複本新增至MusicCollection全域資料表。

```
aws dynamodb update-global-table \
    --global-table-name MusicCollection \
    --replica-updates Create={RegionName=eu-west-1}
```

輸出：

```
{
    "GlobalTableDescription": {
        "ReplicationGroup": [
            {
                "RegionName": "eu-west-1"
            },
            {
                "RegionName": "us-east-2"
            },
            {
                "RegionName": "us-east-1"
            }
        ],
        "GlobalTableArn": "arn:aws:dynamodb::123456789012:global-table/
MusicCollection",
        "CreationDateTime": 1576625818.532,
        "GlobalTableStatus": "ACTIVE",
        "GlobalTableName": "MusicCollection"
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 全域資料表。DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateGlobalTable。

update-item

以下程式碼範例顯示如何使用 update-item。

AWS CLI

範例 1：更新資料表中的項目

下列 update-item 範例會更新 MusicCollection 資料表中的項目。它會新增屬性 (Year) 並修改AlbumTitle屬性。在更新後出現的項目中的所有屬性都會傳回回應中。

```
aws dynamodb update-item \
    --table-name MusicCollection \
    --key file://key.json \
    --update-expression "SET #Y = :y, #AT = :t" \
    --expression-attribute-names file://expression-attribute-names.json \
    --expression-attribute-values file://expression-attribute-values.json  \
    --return-values ALL_NEW \
    --return-consumed-capacity TOTAL \
    --return-item-collection-metrics SIZE
```

key.json 的內容：

```
{
    "Artist": {"S": "Acme Band"},
    "SongTitle": {"S": "Happy Day"}
}
```

expression-attribute-names.json 的內容：

```
{
    "#Y":"Year", "#AT":"AlbumTitle"
}
```

expression-attribute-values.json 的內容：

```
{
```

```
    ":y":{"N": "2015"},
    ":t":{"S": "Louder Than Ever"}
}
```

輸出：

```
{
    "Attributes": {
        "AlbumTitle": {
            "S": "Louder Than Ever"
        },
        "Awards": {
            "N": "10"
        },
        "Artist": {
            "S": "Acme Band"
        },
        "Year": {
            "N": "2015"
        },
        "SongTitle": {
            "S": "Happy Day"
        }
    },
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 3.0
    },
    "ItemCollectionMetrics": {
        "ItemCollectionKey": {
            "Artist": {
                "S": "Acme Band"
            }
        },
        "SizeEstimateRangeGB": [
            0.0,
            1.0
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的撰寫項目。

範例 2：有條件更新項目

下列範例會更新 MusicCollection 資料表中的項目，但前提是現有項目尚未具有 Year 屬性。

```
aws dynamodb update-item \
    --table-name MusicCollection \
    --key file://key.json \
    --update-expression "SET #Y = :y, #AT = :t" \
    --expression-attribute-names file://expression-attribute-names.json \
    --expression-attribute-values file://expression-attribute-values.json  \
    --condition-expression "attribute_not_exists(#Y)"
```

key.json 的內容：

```
{
    "Artist": {"S": "Acme Band"},
    "SongTitle": {"S": "Happy Day"}
}
```

expression-attribute-names.json 的內容：

```
{
    "#Y":"Year",
    "#AT":"AlbumTitle"
}
```

expression-attribute-values.json 的內容：

```
{
    ":y":{"N": "2015"},
    ":t":{"S": "Louder Than Ever"}
}
```

如果項目已有Year屬性，DynamoDB 會傳回下列輸出。

```
An error occurred (ConditionalCheckFailedException) when calling the UpdateItem
 operation: The conditional request failed
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的撰寫項目。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateItem。

update-table-replica-auto-scaling

以下程式碼範例顯示如何使用 update-table-replica-auto-scaling。

AWS CLI

更新全域資料表複本之間的自動擴展設定

下列update-table-replica-auto-scaling範例會更新指定全域資料表複本之間的寫入容量自動擴展設定。

```
aws dynamodb update-table-replica-auto-scaling \
    --table-name MusicCollection \
    --provisioned-write-capacity-auto-scaling-update file://auto-scaling-policy.json
```

auto-scaling-policy.json 的內容：

```
{
    "MinimumUnits": 10,
    "MaximumUnits": 100,
    "AutoScalingDisabled": false,
    "ScalingPolicyUpdate": {
        "PolicyName": "DynamoDBWriteCapacityUtilization:table/MusicCollection",
        "TargetTrackingScalingPolicyConfiguration": {
            "TargetValue": 80
        }
    }
}
```

輸出：

```
{
    "TableAutoScalingDescription": {
        "TableName": "MusicCollection",
        "TableStatus": "ACTIVE",
        "Replicas": [
            {
                "RegionName": "eu-central-1",
                "GlobalSecondaryIndexes": [],
                "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                    "MinimumUnits": 5,
                    "MaximumUnits": 40000,
```

```
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 70.0
                            }
                        }
                    ]
                },
                "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                    "MinimumUnits": 10,
                    "MaximumUnits": 100,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 80.0
                            }
                        }
                    ]
                },
                "ReplicaStatus": "ACTIVE"
            },
            {
                "RegionName": "us-east-1",
                "GlobalSecondaryIndexes": [],
                "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                    "MinimumUnits": 5,
                    "MaximumUnits": 40000,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
```

```
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 70.0
                            }
                        }
                    ]
                },
                "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
                    "MinimumUnits": 10,
                    "MaximumUnits": 100,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 80.0
                            }
                        }
                    ]
                },
                "ReplicaStatus": "ACTIVE"
            },
            {
                "RegionName": "us-east-2",
                "GlobalSecondaryIndexes": [],
                "ReplicaProvisionedReadCapacityAutoScalingSettings": {
                    "MinimumUnits": 5,
                    "MaximumUnits": 40000,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 70.0
                            }
                        }
                    ]
                },
                "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
```

```
                    "MinimumUnits": 10,
                    "MaximumUnits": 100,
                    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
                    "ScalingPolicies": [
                        {
                            "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
                            "TargetTrackingScalingPolicyConfiguration": {
                                "TargetValue": 80.0
                            }
                        }
                    ]
                },
                "ReplicaStatus": "ACTIVE"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 全域資料表。DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateTableReplicaAutoScaling。

update-table

以下程式碼範例顯示如何使用 update-table。

AWS CLI

範例 1：修改資料表的計費模式

下列update-table範例會增加MusicCollection資料表上的佈建讀取和寫入容量。

```
aws dynamodb update-table \
    --table-name MusicCollection \
    --billing-mode PROVISIONED \
    --provisioned-throughput ReadCapacityUnits=15,WriteCapacityUnits=10
```

輸出：

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "AlbumTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "TableName": "MusicCollection",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "UPDATING",
        "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
        "ProvisionedThroughput": {
            "LastIncreaseDateTime": "2020-07-28T13:18:18.921000-07:00",
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 15,
            "WriteCapacityUnits": 10
        },
        "TableSizeBytes": 182,
        "ItemCount": 2,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
        "TableId": "abcd0123-01ab-23cd-0123-abcdef123456",
        "BillingModeSummary": {
            "BillingMode": "PROVISIONED",
            "LastUpdateToPayPerRequestDateTime": "2020-07-28T13:14:48.366000-07:00"
        }
```

```
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的更新資料表。

範例 2：建立全域次要索引

下列範例會將全域次要索引新增至MusicCollection資料表。

```
aws dynamodb update-table \
    --table-name MusicCollection \
    --attribute-definitions AttributeName=AlbumTitle,AttributeType=S \
    --global-secondary-index-updates file://gsi-updates.json
```

gsi-updates.json 的內容：

```
[
    {
        "Create": {
            "IndexName": "AlbumTitle-index",
            "KeySchema": [
                {
                    "AttributeName": "AlbumTitle",
                    "KeyType": "HASH"
                }
            ],
            "ProvisionedThroughput": {
                "ReadCapacityUnits": 10,
                "WriteCapacityUnits": 10
            },
            "Projection": {
                "ProjectionType": "ALL"
            }
        }
    }
]
```

輸出：

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
```

```
                "AttributeName": "AlbumTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "TableName": "MusicCollection",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "UPDATING",
        "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
        "ProvisionedThroughput": {
            "LastIncreaseDateTime": "2020-07-28T12:59:17.537000-07:00",
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 15,
            "WriteCapacityUnits": 10
        },
        "TableSizeBytes": 182,
        "ItemCount": 2,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
        "TableId": "abcd0123-01ab-23cd-0123-abcdef123456",
        "BillingModeSummary": {
            "BillingMode": "PROVISIONED",
            "LastUpdateToPayPerRequestDateTime": "2020-07-28T13:14:48.366000-07:00"
        },
        "GlobalSecondaryIndexes": [
            {
                "IndexName": "AlbumTitle-index",
                "KeySchema": [
                    {
```

```
                        "AttributeName": "AlbumTitle",
                        "KeyType": "HASH"
                    }
                ],
                "Projection": {
                    "ProjectionType": "ALL"
                },
                "IndexStatus": "CREATING",
                "Backfilling": false,
                "ProvisionedThroughput": {
                    "NumberOfDecreasesToday": 0,
                    "ReadCapacityUnits": 10,
                    "WriteCapacityUnits": 10
                },
                "IndexSizeBytes": 0,
                "ItemCount": 0,
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitle-index"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的更新資料表。

範例 3：在資料表上啟用 DynamoDB 串流

下列命令會在MusicCollection資料表上啟用 DynamoDB Streams。

```
aws dynamodb update-table \
    --table-name MusicCollection \
    --stream-specification StreamEnabled=true,StreamViewType=NEW_IMAGE
```

輸出：

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "AlbumTitle",
                "AttributeType": "S"
            },
            {
```

```
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "TableName": "MusicCollection",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "UPDATING",
        "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
        "ProvisionedThroughput": {
            "LastIncreaseDateTime": "2020-07-28T12:59:17.537000-07:00",
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 15,
            "WriteCapacityUnits": 10
        },
        "TableSizeBytes": 182,
        "ItemCount": 2,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
        "TableId": "abcd0123-01ab-23cd-0123-abcdef123456",
        "BillingModeSummary": {
            "BillingMode": "PROVISIONED",
            "LastUpdateToPayPerRequestDateTime": "2020-07-28T13:14:48.366000-07:00"
        },
        "LocalSecondaryIndexes": [
            {
                "IndexName": "AlbumTitleIndex",
                "KeySchema": [
                    {
                        "AttributeName": "Artist",
                        "KeyType": "HASH"
                    },
                    {
```

```
                        "AttributeName": "AlbumTitle",
                        "KeyType": "RANGE"
                    }
                ],
                "Projection": {
                    "ProjectionType": "INCLUDE",
                    "NonKeyAttributes": [
                        "Year",
                        "Genre"
                    ]
                },
                "IndexSizeBytes": 139,
                "ItemCount": 2,
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitleIndex"
            }
        ],
        "GlobalSecondaryIndexes": [
            {
                "IndexName": "AlbumTitle-index",
                "KeySchema": [
                    {
                        "AttributeName": "AlbumTitle",
                        "KeyType": "HASH"
                    }
                ],
                "Projection": {
                    "ProjectionType": "ALL"
                },
                "IndexStatus": "ACTIVE",
                "ProvisionedThroughput": {
                    "NumberOfDecreasesToday": 0,
                    "ReadCapacityUnits": 10,
                    "WriteCapacityUnits": 10
                },
                "IndexSizeBytes": 0,
                "ItemCount": 0,
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitle-index"
            }
        ],
        "StreamSpecification": {
            "StreamEnabled": true,
            "StreamViewType": "NEW_IMAGE"
```

```
        },
        "LatestStreamLabel": "2020-07-28T21:53:39.112",
        "LatestStreamArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/stream/2020-07-28T21:53:39.112"
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的更新資料表。

範例 4：啟用伺服器端加密

下列範例會在MusicCollection資料表上啟用伺服器端加密。

```
aws dynamodb update-table \
    --table-name MusicCollection \
    --sse-specification Enabled=true,SSEType=KMS
```

輸出：

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "AlbumTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "TableName": "MusicCollection",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
```

```
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "ACTIVE",
        "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
        "ProvisionedThroughput": {
            "LastIncreaseDateTime": "2020-07-28T12:59:17.537000-07:00",
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 15,
            "WriteCapacityUnits": 10
        },
        "TableSizeBytes": 182,
        "ItemCount": 2,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
        "TableId": "abcd0123-01ab-23cd-0123-abcdef123456",
        "BillingModeSummary": {
            "BillingMode": "PROVISIONED",
            "LastUpdateToPayPerRequestDateTime": "2020-07-28T13:14:48.366000-07:00"
        },
        "LocalSecondaryIndexes": [
            {
                "IndexName": "AlbumTitleIndex",
                "KeySchema": [
                    {
                        "AttributeName": "Artist",
                        "KeyType": "HASH"
                    },
                    {
                        "AttributeName": "AlbumTitle",
                        "KeyType": "RANGE"
                    }
                ],
                "Projection": {
                    "ProjectionType": "INCLUDE",
                    "NonKeyAttributes": [
                        "Year",
                        "Genre"
                    ]
                },
                "IndexSizeBytes": 139,
                "ItemCount": 2,
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitleIndex"
            }
```

```
        ],
        "GlobalSecondaryIndexes": [
            {
                "IndexName": "AlbumTitle-index",
                "KeySchema": [
                    {
                        "AttributeName": "AlbumTitle",
                        "KeyType": "HASH"
                    }
                ],
                "Projection": {
                    "ProjectionType": "ALL"
                },
                "IndexStatus": "ACTIVE",
                "ProvisionedThroughput": {
                    "NumberOfDecreasesToday": 0,
                    "ReadCapacityUnits": 10,
                    "WriteCapacityUnits": 10
                },
                "IndexSizeBytes": 0,
                "ItemCount": 0,
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitle-index"
            }
        ],
        "StreamSpecification": {
            "StreamEnabled": true,
            "StreamViewType": "NEW_IMAGE"
        },
        "LatestStreamLabel": "2020-07-28T21:53:39.112",
        "LatestStreamArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/stream/2020-07-28T21:53:39.112",
        "SSEDescription": {
            "Status": "UPDATING"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的更新資料表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateTable。

update-time-to-live

以下程式碼範例顯示如何使用 update-time-to-live。

AWS CLI

更新資料表上的存留時間設定

下列update-time-to-live範例會啟用指定資料表上的存留時間。

```
aws dynamodb update-time-to-live \
    --table-name MusicCollection \
    --time-to-live-specification Enabled=true,AttributeName=ttl
```

輸出：

```
{
    "TimeToLiveSpecification": {
        "Enabled": true,
        "AttributeName": "ttl"
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的存留時間。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateTimeToLive。

使用 的 DynamoDB Streams 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 DynamoDB Streams 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`describe-stream`

以下程式碼範例顯示如何使用 `describe-stream`。

AWS CLI

取得 DynamoDB 串流的相關資訊

下列`describe-stream`命令會顯示特定 DynamoDB 串流的相關資訊。

```
aws dynamodbstreams describe-stream \
    --stream-arn arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576
```

輸出：

```
{
    "StreamDescription": {
        "StreamArn": "arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576",
        "StreamLabel": "2019-10-22T18:02:01.576",
        "StreamStatus": "ENABLED",
        "StreamViewType": "NEW_AND_OLD_IMAGES",
        "CreationRequestDateTime": 1571767321.571,
        "TableName": "Music",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "Shards": [
            {
                "ShardId": "shardId-00000001571767321804-697ce3d2",
                "SequenceNumberRange": {
                    "StartingSequenceNumber": "400000000000642977831",
                    "EndingSequenceNumber": "400000000000642977831"
                }
```

```
            },
            {
                "ShardId": "shardId-00000001571780995058-40810d86",
                "SequenceNumberRange": {
                    "StartingSequenceNumber": "757400000000005655171150"
                },
                "ParentShardId": "shardId-00000001571767321804-697ce3d2"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的使用 DynamoDB Streams 擷取資料表活動。 DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStream。

get-records

以下程式碼範例顯示如何使用 get-records。

AWS CLI

從 Dynamodb 串流取得記錄

下列get-records命令會使用指定的 Amazon DynamoDB 碎片迭代器擷取記錄。

```
aws dynamodbstreams get-records \
    --shard-iterator "arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576|1|
AAAAAAAAAAGgM3YZ89vLZZxjmoQeo33r9M4x3+zmmTLsiL86MfrF4+B4EbsByi52InVmiONmy6xVW4IRcIIbs1zO7MNI
+CjNPlqQjnyRSAnfOwWmKhL1/KNParWSfz2odf780oOObIDIWRRMkt7+Hyzh9SD
+hFxFAWR5C7QIlOXPc8mRBfNIazfrVCjJK8/jsjCzsqNyXKzJbhh+GXCoxYN
+Kpmg4nyj1EAsYhbGL35muvHFoHjcyuynbsczbWaXNfThDwRAyvoTmc8XhHKtAWUbJiaVd8ZPtQwDsThCrmDRPIdmTRG
+w/lEGS05ha1qNP+Vl4+tuhz2TRnhnJo/pny9GI/yGpce97mWvSPr5KPwy+Dtcm5BHayBs
+PVYHITaTliInFlT
+LCwvaz1QH3MY3b8A05Z800wjpktm60iQqtMeDwN4NX6FrcxR34JoFKGsgR8XkHVJzz2xr1xqSJ12ycpNTyHnndusw==
```

輸出：

```
{
    "Records": [
        {
```

```
            "eventID": "c3b5d798eef6215d42f8137b19a88e50",
            "eventName": "INSERT",
            "eventVersion": "1.1",
            "eventSource": "aws:dynamodb",
            "awsRegion": "us-west-1",
            "dynamodb": {
                "ApproximateCreationDateTime": 1571849028.0,
                "Keys": {
                    "Artist": {
                        "S": "No One You Know"
                    },
                    "SongTitle": {
                        "S": "Call Me Today"
                    }
                },
                "NewImage": {
                    "AlbumTitle": {
                        "S": "Somewhat Famous"
                    },
                    "Artist": {
                        "S": "No One You Know"
                    },
                    "Awards": {
                        "N": "1"
                    },
                    "SongTitle": {
                        "S": "Call Me Today"
                    }
                },
                "SequenceNumber": "700000000013256296913",
                "SizeBytes": 119,
                "StreamViewType": "NEW_AND_OLD_IMAGES"
            }
        },
        {
            "eventID": "878960a6967867e2da16b27380a27328",
            "eventName": "INSERT",
            "eventVersion": "1.1",
            "eventSource": "aws:dynamodb",
            "awsRegion": "us-west-1",
            "dynamodb": {
                "ApproximateCreationDateTime": 1571849029.0,
                "Keys": {
                    "Artist": {
```

```
                    "S": "Acme Band"
                },
                "SongTitle": {
                    "S": "Happy Day"
                }
            },
            "NewImage": {
                "AlbumTitle": {
                    "S": "Songs About Life"
                },
                "Artist": {
                    "S": "Acme Band"
                },
                "Awards": {
                    "N": "10"
                },
                "SongTitle": {
                    "S": "Happy Day"
                }
            },
            "SequenceNumber": "800000000013256297217",
            "SizeBytes": 100,
            "StreamViewType": "NEW_AND_OLD_IMAGES"
        }
    },
    {
        "eventID": "520fabde080e159fc3710b15ee1d4daa",
        "eventName": "MODIFY",
        "eventVersion": "1.1",
        "eventSource": "aws:dynamodb",
        "awsRegion": "us-west-1",
        "dynamodb": {
            "ApproximateCreationDateTime": 1571849734.0,
            "Keys": {
                "Artist": {
                    "S": "Acme Band"
                },
                "SongTitle": {
                    "S": "Happy Day"
                }
            },
            "NewImage": {
                "AlbumTitle": {
                    "S": "Updated Album Title"
```

```
                    },
                    "Artist": {
                        "S": "Acme Band"
                    },
                    "Awards": {
                        "N": "10"
                    },
                    "SongTitle": {
                        "S": "Happy Day"
                    }
                },
                "OldImage": {
                    "AlbumTitle": {
                        "S": "Songs About Life"
                    },
                    "Artist": {
                        "S": "Acme Band"
                    },
                    "Awards": {
                        "N": "10"
                    },
                    "SongTitle": {
                        "S": "Happy Day"
                    }
                },
                "SequenceNumber": "900000000013256687845",
                "SizeBytes": 170,
                "StreamViewType": "NEW_AND_OLD_IMAGES"
            }
        }
    ],
    "NextShardIterator": "arn:aws:dynamodb:us-west-1:123456789012:table/
Music/stream/2019-10-23T16:41:08.740|1|AAAAAAAAAAEhEI04jkFLW
+LKOwivjT8d/IHEh3iExV2xK00aTxEzVy1C1C7Kbb5+ZOW6bT9VQ2n1/
mrs7+PRiaOZCHJu7JHJVW7zlsqOi/ges3fw8GYEymyL+piEk35cx67rQqwKKyq
+Q6w9JyjreIOj4F2lWLV26lBwRTrIYC4IB7C3BZZK4715QwYdDxNdVHiSBRZX8UqoS6WOt0F87xZLNB9F/
NhYBLXi/wcGvAcBcC0TNIOH+N0NqwtoB/
FGCkNrf8YZ0xRoNN6RgGuVWHF3pxOhxEJeFZoSoJTIKeG9YcYxzi5Ci/
mhdtm7tBXnbw5c6xmsGsBqTirNjlDyJLcWl8Cl0UOLX63Ufo/5QliztcjEbKsQe28x8LM8o7VH1Is0fF/
ITt8awSA4igyJS0P87GN8Qri8kj8iaE35805jBHWF2wvwT6Iy2xGrR2r2HzYps9dwGOarVdEITaJfWzNoL4HajMhmREZ
+V04i1YIeHMXJfcwetNRuIbdQXfJht2NQZa4PVV6iknY6d19MrdbSTMKoqAuvp6g3Q2jH4t7GKCLWgodcPAn8g5+43Da
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的使用 DynamoDB Streams 擷取資料表活動。 DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRecords。

get-shard-iterator

以下程式碼範例顯示如何使用 get-shard-iterator。

AWS CLI

若要取得碎片迭代器

下列get-shard-iterator命令會擷取指定碎片的碎片疊代運算。

```
aws dynamodbstreams get-shard-iterator \
    --stream-arn arn:aws:dynamodb:us-west-1:12356789012:table/Music/
stream/2019-10-22T18:02:01.576 \
    --shard-id shardId-00000001571780995058-40810d86 \
    --shard-iterator-type LATEST
```

輸出：

```
{
    "ShardIterator": "arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576|1|
AAAAAAAAAAGgM3YZ89vLZZxjmoQeo33r9M4x3+zmmTLsiL86MfrF4+B4EbsByi52InVmiONmy6xVW4IRcIIbs1zO7MNI
+CjNPlqQjnyRSAnfOwWmKhL1/KNParWSfz2odf780oOObIDIWRRMkt7+Hyzh9SD
+hFxFAWR5C7QIlOXPc8mRBfNIazfrVCjJK8/jsjCzsqNyXKzJbhh+GXCoxYN
+Kpmg4nyj1EAsYhbGL35muvHFoHjcyuynbsczbWaXNfThDwRAyvoTmc8XhHKtAWUbJiaVd8ZPtQwDsThCrmDRPIdmTRG
+w/lEGS05ha1qNP+Vl4+tuhz2TRnhnJo/pny9GI/yGpce97mWvSPr5KPwy+Dtcm5BHayBs
+PVYHITaTliInFlT
+LCwvaz1QH3MY3b8A05Z800wjpktm60iQqtMeDwN4NX6FrcxR34JoFKGsgR8XkHVJzz2xr1xqSJ12ycpNTyHnndusw==
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的使用 DynamoDB Streams 擷取資料表活動。 DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetShardIterator。

`list-streams`

以下程式碼範例顯示如何使用 `list-streams`。

AWS CLI

列出 DynamoDB 串流

下列`list-streams`命令會列出預設 AWS 區域中所有現有的 Amazon DynamoDB 串流。

```
aws dynamodbstreams list-streams
```

輸出：

```
{
    "Streams": [
        {
            "StreamArn": "arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576",
            "TableName": "Music",
            "StreamLabel": "2019-10-22T18:02:01.576"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的使用 DynamoDB Streams 擷取資料表活動。 DynamoDB

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListStreams。

使用 的 Amazon EC2 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon EC2 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

accept-address-transfer

以下程式碼範例顯示如何使用 accept-address-transfer。

AWS CLI

接受轉移至您帳戶的彈性 IP 地址

下列accept-address-transfer範例接受將指定的彈性 IP 地址轉移到您的帳戶。

```
aws ec2 accept-address-transfer \
    --address 100.21.184.216
```

輸出：

```
{
    "AddressTransfer": {
        "PublicIp": "100.21.184.216",
        "AllocationId": "eipalloc-09ad461b0d03f6aaf",
        "TransferAccountId": "123456789012",
        "TransferOfferExpirationTimestamp": "2023-02-22T20:51:10.000Z",
        "TransferOfferAcceptedTimestamp": "2023-02-22T22:52:54.000Z",
        "AddressTransferStatus": "accepted"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的轉移彈性 IP 地址。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptAddressTransfer。

accept-reserved-instances-exchange-quote

以下程式碼範例顯示如何使用 accept-reserved-instances-exchange-quote。

AWS CLI

執行可轉換預留執行個體交換

此範例會執行指定可轉換預留執行個體的交換。

命令：

```
aws ec2 accept-reserved-instances-exchange-quote --reserved-
instance-ids 7b8750c3-397e-4da4-bbcb-a45ebexample --target-
configurations OfferingId=b747b472-423c-48f3-8cee-679bcexample
```

輸出：

```
{
  "ExchangeId": "riex-e68ed3c1-8bc8-4c17-af77-811afexample"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptReservedInstancesExchangeQuote。

accept-transit-gateway-peering-attachment

以下程式碼範例顯示如何使用 accept-transit-gateway-peering-attachment。

AWS CLI

接受傳輸閘道對等互連附件

下列accept-transit-gateway-peering-attachment範例接受指定的傳輸閘道對等互連附件。--region 參數指定接受者傳輸閘道所在的區域。

```
aws ec2 accept-transit-gateway-peering-attachment \
    --transit-gateway-attachment-id tgw-attach-4455667788aabbccd \
    --region us-east-2
```

輸出：

```
{
    "TransitGatewayPeeringAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
        "RequesterTgwInfo": {
            "TransitGatewayId": "tgw-123abc05e04123abc",
            "OwnerId": "123456789012",
            "Region": "us-west-2"
        },
        "AccepterTgwInfo": {
            "TransitGatewayId": "tgw-11223344aabbcc112",
            "OwnerId": "123456789012",
```

```
            "Region": "us-east-2"
        },
        "State": "pending",
        "CreationTime": "2019-12-09T11:38:31.000Z"
    }
}
```

如需詳細資訊，請參閱《傳輸閘道指南》中的傳輸閘道對等連接。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptTransitGatewayPeeringAttachment。

accept-transit-gateway-vpc-attachment

以下程式碼範例顯示如何使用 accept-transit-gateway-vpc-attachment。

AWS CLI

接受將 VPC 連接至傳輸閘道的請求。

下列accept-transit-gateway-vpc-attachment範例接受請求存放區指定的附件。

```
aws ec2 accept-transit-gateway-vpc-attachment \
    --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE
```

輸出：

```
{
    "TransitGatewayVpcAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-0a34fe6b4fEXAMPLE",
        "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
        "VpcId": "vpc-07e8ffd50fEXAMPLE",
        "VpcOwnerId": "123456789012",
        "State": "pending",
        "SubnetIds": [
            "subnet-0752213d59EXAMPLE"
        ],
        "CreationTime": "2019-07-10T17:33:46.000Z",
        "Options": {
            "DnsSupport": "enable",
            "Ipv6Support": "disable"
        }
```

```
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的 Transit Gateway Attachments to a VPC。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptTransitGatewayVpcAttachment。

accept-vpc-endpoint-connections

以下程式碼範例顯示如何使用 accept-vpc-endpoint-connections。

AWS CLI

接受介面端點連線請求

此範例接受指定端點服務的指定端點連線請求。

命令：

```
aws ec2 accept-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --
vpc-endpoint-ids vpce-0c1308d7312217abc
```

輸出：

```
{
  "Unsuccessful": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptVpcEndpointConnections。

accept-vpc-peering-connection

以下程式碼範例顯示如何使用 accept-vpc-peering-connection。

AWS CLI

接受 VPC 對等互連

此範例接受指定的 VPC 互連連線請求。

命令：

```
aws ec2 accept-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

輸出：

```
{
  "VpcPeeringConnection": {
    "Status": {
      "Message": "Provisioning",
      "Code": "provisioning"
    },
    "Tags": [],
    "AccepterVpcInfo": {
      "OwnerId": "444455556666",
      "VpcId": "vpc-44455566",
      "CidrBlock": "10.0.1.0/28"
    },
    "VpcPeeringConnectionId": "pcx-1a2b3c4d",
    "RequesterVpcInfo": {
      "OwnerId": "444455556666",
      "VpcId": "vpc-111abc45",
      "CidrBlock": "10.0.0.0/28"
    }
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptVpcPeeringConnection。

advertise-byoip-cidr

以下程式碼範例顯示如何使用 advertise-byoip-cidr。

AWS CLI

公告地址範圍

下列advertise-byoip-cidr範例會公告指定的公有 IPv4 地址範圍。

```
aws ec2 advertise-byoip-cidr \
    --cidr 203.0.113.25/24
```

輸出：

```
{
    "ByoipCidr": {
        "Cidr": "203.0.113.25/24",
        "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
        "State": "provisioned"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdvertiseByoipCidr。

`allocate-address`

以下程式碼範例顯示如何使用 `allocate-address`。

AWS CLI

範例 1：從 Amazon 的地址集區配置彈性 IP 地址

以下 `allocate-address` 範例會配置彈性 IP 地址。Amazon EC2 會從 Amazon 的地址集區中選取地址。

```
aws ec2 allocate-address
```

輸出：

```
{
    "PublicIp": "70.224.234.241",
    "AllocationId": "eipalloc-01435ba59eEXAMPLE",
    "PublicIpv4Pool": "amazon",
    "NetworkBorderGroup": "us-west-2",
    "Domain": "vpc"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的彈性 IP 地址。

範例 2：配置彈性 IP 地址並將其與網路邊界群組建立關聯

下列 `allocate-address` 範例會配置彈性 IP 地址，並將其與指定的網路邊界群組建立關聯。

```
aws ec2 allocate-address \
    --network-border-group us-west-2-lax-1
```

輸出：

```
{
    "PublicIp": "70.224.234.241",
    "AllocationId": "eipalloc-e03dd489ceEXAMPLE",
    "PublicIpv4Pool": "amazon",
    "NetworkBorderGroup": "us-west-2-lax-1",
    "Domain": "vpc"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的彈性 IP 地址。

範例 3：從您擁有的地址集區配置彈性 IP 地址

以下 `allocate-address` 範例會從您已用於 Amazon Web Services 帳戶的地址集區配置彈性 IP 地址。Amazon EC2 會從此地址集區中選取地址。

```
aws ec2 allocate-address \
    --public-ipv4-pool ipv4pool-ec2-1234567890abcdef0
```

輸出：

```
{
    "AllocationId": "eipalloc-02463d08ceEXAMPLE",
    "NetworkBorderGroup": "us-west-2",
    "CustomerOwnedIp": "18.218.95.81",
    "CustomerOwnedIpv4Pool": "ipv4pool-ec2-1234567890abcdef0",
    "Domain": "vpc"
    "NetworkBorderGroup": "us-west-2",
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的彈性 IP 地址。

範例 4：從 IPAM 集區配置彈性 IP 地址

下列`allocate-address`範例會從 Amazon VPC IP Address Manager (IPAM) 集區配置特定的 /32 彈性 IP 地址。

```
aws ec2 allocate-address \
    --region us-east-1 \
    --ipam-pool-id ipam-pool-1234567890abcdef0 \
```

```
    --address 192.0.2.0
```

輸出：

```
{
    "PublicIp": "192.0.2.0",
    "AllocationId": "eipalloc-abcdef01234567890",
    "PublicIpv4Pool": "ipam-pool-1234567890abcdef0",
    "NetworkBorderGroup": "us-east-1",
    "Domain": "vpc"
}
```

如需詳細資訊，請參閱「Amazon VPC IPAM 使用者指南」中的從 IPAM 集區配置循序彈性 IP 位址。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AllocateAddress。

allocate-hosts

以下程式碼範例顯示如何使用 allocate-hosts。

AWS CLI

範例 1：配置專用主機

下列allocate-hosts範例會在eu-west-1a可用區域中配置單一專用主機，您可以在其上啟動m5.large執行個體。根據預設，專用主機僅接受目標執行個體啟動，且不支援主機復原。

```
aws ec2 allocate-hosts \
    --instance-type m5.large \
    --availability-zone eu-west-1a \
    --quantity 1
```

輸出：

```
{
    "HostIds": [
        "h-07879acf49EXAMPLE"
    ]
}
```

範例 2：配置已啟用自動置放和主機復原的專用主機

下列allocate-hosts範例會在啟用自動置放和主機復原的eu-west-1a可用區域中配置單一專用主機。

```
aws ec2 allocate-hosts \
    --instance-type m5.large \
    --availability-zone eu-west-1a \
    --auto-placement on \
    --host-recovery on \
    --quantity 1
```

輸出：

```
{
     "HostIds": [
         "h-07879acf49EXAMPLE"
     ]
}
```

範例 3：使用標籤配置專用主機

下列allocate-hosts範例會配置單一專用主機，並使用名為 purpose且值為 的金鑰來套用標籤production。

```
aws ec2 allocate-hosts \
    --instance-type m5.large \
    --availability-zone eu-west-1a \
    --quantity 1 \
    --tag-specifications 'ResourceType=dedicated-
host,Tags={Key=purpose,Value=production}'
```

輸出：

```
{
    "HostIds": [
        "h-07879acf49EXAMPLE"
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的配置專用主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AllocateHosts。

allocate-ipam-pool-cidr

以下程式碼範例顯示如何使用 allocate-ipam-pool-cidr。

AWS CLI

從 IPAM 集區配置 CIDR

下列allocate-ipam-pool-cidr範例會從 IPAM 集區配置 CIDR。

(Linux)：

```
aws ec2 allocate-ipam-pool-cidr \
    --ipam-pool-id ipam-pool-0533048da7d823723 \
    --netmask-length 24
```

(Windows)：

```
aws ec2 allocate-ipam-pool-cidr ^
   --ipam-pool-id ipam-pool-0533048da7d823723 ^
   --netmask-length 24
```

輸出：

```
{
    "IpamPoolAllocation": {
        "Cidr": "10.0.0.0/24",
        "IpamPoolAllocationId": "ipam-pool-alloc-018ecc28043b54ba38e2cd99943cebfbd",
        "ResourceType": "custom",
        "ResourceOwner": "123456789012"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的手動將 CIDR 配置到集區以預留 IP 地址空間。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AllocateIpamPoolCidr。

apply-security-groups-to-client-vpn-target-network

以下程式碼範例顯示如何使用 apply-security-groups-to-client-vpn-target-network。

AWS CLI

將安全群組套用至 Client VPN 端點的目標網路

下列apply-security-groups-to-client-vpn-target-network範例會將安全群組套用至指定目標網路與 Client VPN 端點之間的sg-01f6e627a89f4db32關聯。

```
aws ec2 apply-security-groups-to-client-vpn-target-network \
    --security-group-ids sg-01f6e627a89f4db32 \
    --vpc-id vpc-0e2110c2f324332e0 \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

輸出：

```
{
    "SecurityGroupIds": [
        "sg-01f6e627a89f4db32"
    ]
}
```

如需詳細資訊，請參閱 AWS Client VPN 管理員指南中的目標網路。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ApplySecurityGroupsToClientVpnTargetNetwork。

assign-ipv6-addresses

以下程式碼範例顯示如何使用 assign-ipv6-addresses。

AWS CLI

將特定 IPv6 地址指派給網路界面

此範例會將指定的 IPv6 地址指派給指定的網路介面。

命令：

```
aws ec2 assign-ipv6-addresses --network-interface-id eni-38664473 --ipv6-
addresses 2001:db8:1234:1a00:3304:8879:34cf:4071 2001:db8:1234:1a00:9691:9503:25ad:1761
```

輸出：

```
{
  "AssignedIpv6Addresses": [
      "2001:db8:1234:1a00:3304:8879:34cf:4071",
      "2001:db8:1234:1a00:9691:9503:25ad:1761"
  ],
  "NetworkInterfaceId": "eni-38664473"
}
```

將 Amazon 選取的 IPv6 地址指派給網路介面

此範例會將兩個 IPv6 地址指派給指定的網路介面。Amazon 會自動從子網路 IPv6 CIDR 區塊範圍中的可用 IPv6 地址指派這些 IPv6 地址。

命令：

```
aws ec2 assign-ipv6-addresses --network-interface-id eni-38664473 --ipv6-address-
count 2
```

輸出：

```
{
  "AssignedIpv6Addresses": [
      "2001:db8:1234:1a00:3304:8879:34cf:4071",
      "2001:db8:1234:1a00:9691:9503:25ad:1761"
  ],
  "NetworkInterfaceId": "eni-38664473"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssignIpv6Addresses。

assign-private-ip-addresses

以下程式碼範例顯示如何使用 assign-private-ip-addresses。

AWS CLI

指派網路界面的特定次要私有 IP 地址

此範例會將指定的次要私有 IP 地址指派給指定的網路介面。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 assign-private-ip-addresses --network-interface-id eni-e5aa89a3 --private-ip-addresses 10.0.0.82
```

將 Amazon EC2 選取的次要私有 IP 地址指派給網路介面

此範例會將兩個次要私有 IP 地址指派給指定的網路介面。Amazon EC2 會自動從與網路介面相關聯的子網路 CIDR 區塊範圍內的可用 IP 地址指派這些 IP 地址。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 assign-private-ip-addresses --network-interface-id eni-e5aa89a3 --secondary-private-ip-address-count 2
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssignPrivateIpAddresses。

assign-private-nat-gateway-address

以下程式碼範例顯示如何使用 assign-private-nat-gateway-address。

AWS CLI

將私有 IP 地址指派給私有 NAT 閘道

下列assign-private-nat-gateway-address範例會將兩個私有 IP 地址指派給指定的私有 NAT 閘道。

```
aws ec2 assign-private-nat-gateway-address \
    --nat-gateway-id nat-1234567890abcdef0 \
    --private-ip-address-count 2
```

輸出：

```
{
    "NatGatewayId": "nat-1234567890abcdef0",
    "NatGatewayAddresses": [
        {
            "NetworkInterfaceId": "eni-0065a61b324d1897a",
            "IsPrimary": false,
            "Status": "assigning"
```

```
        },
        {
            "NetworkInterfaceId": "eni-0065a61b324d1897a",
            "IsPrimary": false,
            "Status": "assigning"
        }
    ]
}
```

如需詳細資訊，請參閱 《Amazon VPC 使用者指南》中的 NAT 閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssignPrivateNatGatewayAddress。

associate-address

以下程式碼範例顯示如何使用 associate-address。

AWS CLI

範例 1：將彈性 IP 地址與執行個體建立關聯

下列associate-address範例會將彈性 IP 地址與指定的 EC2 執行個體建立關聯。

```
aws ec2 associate-address \
    --instance-id i-0b263919b6498b123 \
    --allocation-id eipalloc-64d5890a
```

輸出：

```
{
    "AssociationId": "eipassoc-2bebb745"
}
```

範例 2：將彈性 IP 地址與網路介面建立關聯

下列associate-address範例會將指定的彈性 IP 地址與指定的網路介面建立關聯。

```
aws ec2 associate-address
    --allocation-id eipalloc-64d5890a \
    --network-interface-id eni-1a2b3c4d
```

輸出：

```
{
    "AssociationId": "eipassoc-2bebb745"
}
```

範例 3：將彈性 IP 地址與私有 IP 地址建立關聯

下列associate-address範例會將指定的彈性 IP 地址與指定網路介面中指定的私有 IP 地址建立關聯。

```
aws ec2 associate-address \
    --allocation-id eipalloc-64d5890a \
    --network-interface-id eni-1a2b3c4d \
    --private-ip-address 10.0.0.85
```

輸出：

```
{
    "AssociationId": "eipassoc-2bebb745"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的彈性 IP 地址。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AssociateAddress。

associate-client-vpn-target-network

以下程式碼範例顯示如何使用 associate-client-vpn-target-network。

AWS CLI

將目標網路與 Client VPN 端點建立關聯

下列associate-client-vpn-target-network範例會將子網路與指定的 Client VPN 端點建立關聯。

```
aws ec2 associate-client-vpn-target-network \
    --subnet-id subnet-0123456789abcabca \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

輸出：

```
{
    "AssociationId": "cvpn-assoc-12312312312312312",
    "Status": {
        "Code": "associating"
    }
}
```

如需詳細資訊，請參閱 AWS Client VPN 管理員指南中的目標網路。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateClientVpnTargetNetwork。

associate-dhcp-options

以下程式碼範例顯示如何使用 associate-dhcp-options。

AWS CLI

將 DHCP 選項集與您的 VPC 建立關聯

此範例會將指定的 DHCP 選項集與指定的 VPC 建立關聯。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 associate-dhcp-options --dhcp-options-id dopt-d9070ebb --vpc-id vpc-a01106c2
```

將預設 DHCP 選項集與您的 VPC 建立關聯

此範例會將預設 DHCP 選項集與指定的 VPC 建立關聯。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 associate-dhcp-options --dhcp-options-id default --vpc-id vpc-a01106c2
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateDhcpOptions。

associate-iam-instance-profile

以下程式碼範例顯示如何使用 associate-iam-instance-profile。

AWS CLI

將 IAM 執行個體描述檔與執行個體建立關聯

此範例會將名為 的 IAM 執行個體描述檔admin-role與執行個體 建立關聯i-123456789abcde123。

命令：

```
aws ec2 associate-iam-instance-profile --instance-id i-123456789abcde123 --iam-instance-profile Name=admin-role
```

輸出：

```
{
  "IamInstanceProfileAssociation": {
      "InstanceId": "i-123456789abcde123",
      "State": "associating",
      "AssociationId": "iip-assoc-0e7736511a163c209",
      "IamInstanceProfile": {
          "Id": "AIPAJBLK7RKJKWDXVHIEC",
          "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"
      }
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateIamInstanceProfile。

associate-instance-event-window

以下程式碼範例顯示如何使用 associate-instance-event-window。

AWS CLI

範例 1：將一或多個執行個體與事件時段建立關聯

下列associate-instance-event-window範例會將一或多個執行個體與事件時段建立關聯。

```
aws ec2 associate-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
```

```
    --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

輸出：

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [
                "i-1234567890abcdef0",
                "i-0598c7d356eba48d7"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

如需事件時段限制，請參閱《Amazon EC2 使用者指南》的排程事件一節中的考量事項。

範例 2：將執行個體標籤與事件時段建立關聯

下列associate-instance-event-window範例會將執行個體標籤與事件時段建立關聯。輸入 instance-event-window-id 參數以指定事件時段。若要關聯執行個體標籤，請指定 association-target 參數，並為參數值指定一或多個標籤。

```
aws ec2 associate-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"
```

輸出：

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
```

```
            "InstanceIds": [],
            "Tags": [
                {
                    "Key": "k2",
                    "Value": "v2"
                },
                {
                    "Key": "k1",
                    "Value": "v1"
                }
            ],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

如需事件時段限制，請參閱《Amazon EC2 使用者指南》的排程事件一節中的考量事項。

範例 3：將專用主機與事件時段建立關聯

下列associate-instance-event-window範例會將專用主機與事件時段建立關聯。輸入 instance-event-window-id 參數以指定事件時段。若要建立專用主機的關聯，請指定 --association-target 參數，並為參數值指定其中一個專用主機 IDs。

```
aws ec2 associate-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "DedicatedHostIds=h-029fa35a02b99801d"
```

輸出：

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": [
                "h-029fa35a02b99801d"
```

```
            ]
        },
        "State": "creating"
    }
}
```

如需事件時段限制，請參閱《Amazon EC2 使用者指南》的排程事件一節中的考量事項。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateInstanceEventWindow。

associate-ipam-resource-discovery

以下程式碼範例顯示如何使用 associate-ipam-resource-discovery。

AWS CLI

將資源探索與 IPAM 建立關聯

在此範例中，您是 IPAM 委派管理員，並且資源探索已由另一個 AWS 帳戶建立並與您共用，因此您可以使用 IPAM 來管理和監控另一個帳戶擁有的資源 CIDRs。

注意

若要完成此請求，您將需要可透過 describe-ipam-resource-discoveries 取得的資源探索 ID，以及可透過 describe-ipams 取得的 IPAM ID。您要建立關聯的資源探索必須先使用 RAM AWS 與您的帳戶共用。--region您輸入的 必須與您要建立關聯的 IPAM 主區域相符。

下列associate-ipam-resource-discovery範例會將資源探索與 IPAM 建立關聯。

```
aws ec2 associate-ipam-resource-discovery \
    --ipam-id ipam-005f921c17ebd5107 \
    --ipam-resource-discovery-id ipam-res-disco-03e0406de76a044ee \
    --tag-specifications 'ResourceType=ipam-resource-discovery,Tags=[{Key=cost-
center,Value=cc123}]' \
    --region us-east-1
```

輸出：

```
{
    {
        "IpamResourceDiscoveryAssociation": {
            "OwnerId": "320805250157",
```

```
            "IpamResourceDiscoveryAssociationId": "ipam-res-disco-
assoc-04382a6346357cf82",
            "IpamResourceDiscoveryAssociationArn": "arn:aws:ec2::320805250157:ipam-
resource-discovery-association/ipam-res-disco-assoc-04382a6346357cf82",
            "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
            "IpamId": "ipam-005f921c17ebd5107",
            "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
            "IpamRegion": "us-east-1",
            "IsDefault": false,
            "ResourceDiscoveryStatus": "active",
            "State": "associate-in-progress",
            "Tags": []
        }
    }
}
```

建立資源探索的關聯後，您可以監控和/或管理其他帳戶所建立資源的 IP 地址。如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的將 IPAM 與組織外部的帳戶整合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateIpamResourceDiscovery。

associate-nat-gateway-address

以下程式碼範例顯示如何使用 associate-nat-gateway-address。

AWS CLI

將彈性 IP 地址與公有 NAT 閘道建立關聯

下列associate-nat-gateway-address範例會將指定的彈性 IP 地址與指定的公有 NAT 閘道建立關聯。 AWS 會自動指派次要私有 IPv4 地址。

```
aws ec2 associate-nat-gateway-address \
    --nat-gateway-id nat-1234567890abcdef0 \
    --allocation-ids eipalloc-0be6ecac95EXAMPLE
```

輸出：

```
{
    "NatGatewayId": "nat-1234567890abcdef0",
    "NatGatewayAddresses": [
        {
```

```
            "AllocationId": "eipalloc-0be6ecac95EXAMPLE",
            "NetworkInterfaceId": "eni-09cc4b2558794f7f9",
            "IsPrimary": false,
            "Status": "associating"
        }
    ]
}
```

如需詳細資訊，請參閱 《Amazon VPC 使用者指南》中的 NAT 閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateNatGatewayAddress。

associate-route-table

以下程式碼範例顯示如何使用 associate-route-table。

AWS CLI

將路由表與子網路建立關聯

此範例會將指定的路由表與指定的子網路建立關聯。

命令：

```
aws ec2 associate-route-table --route-table-id rtb-22574640 --subnet-
id subnet-9d4a7b6c
```

輸出：

```
{
    "AssociationId": "rtbassoc-781d0d1a"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateRouteTable。

associate-security-group-vpc

以下程式碼範例顯示如何使用 associate-security-group-vpc。

AWS CLI

將安全群組與另一個 VPC 建立關聯

下列associate-security-group-vpc範例會將指定的安全群組與指定的 VPC 建立關聯。

```
aws ec2 associate-security-group-vpc \
    --group-id sg-04dbb43907d3f8a78 \
    --vpc-id vpc-0bf4c2739bc05a694
```

輸出：

```
{
    "State": "associating"
}
```

如需詳細資訊，請參閱《Amazon VPCs》中的將安全群組與多個 VPC 建立關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateSecurityGroupVpc。

associate-subnet-cidr-block

以下程式碼範例顯示如何使用 associate-subnet-cidr-block。

AWS CLI

將 IPv6 CIDR 區塊與子網路建立關聯

此範例會將 IPv6 CIDR 區塊與指定的子網路建立關聯。

命令：

```
aws ec2 associate-subnet-cidr-block --subnet-id subnet-5f46ec3b --ipv6-cidr-
block 2001:db8:1234:1a00::/64
```

輸出：

```
{
  "SubnetId": "subnet-5f46ec3b",
  "Ipv6CidrBlockAssociation": {
      "Ipv6CidrBlock": "2001:db8:1234:1a00::/64",
      "AssociationId": "subnet-cidr-assoc-3aa54053",
      "Ipv6CidrBlockState": {
          "State": "associating"
```

```
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateSubnetCidrBlock。

associate-transit-gateway-multicast-domain

以下程式碼範例顯示如何使用 associate-transit-gateway-multicast-domain。

AWS CLI

將傳輸閘道與多點傳送網域建立關聯

下列associate-transit-gateway-multicast-domain範例會將指定的子網路和附件與指定的多點傳送網域建立關聯。

```
aws ec2 associate-transit-gateway-multicast-domain \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \
    --transit-gateway-attachment-id tgw-attach-028c1dd0f8f5cbe8e \
    --subnet-ids subnet-000de86e3b49c932a \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

輸出：

```
{
    "Associations": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
        "TransitGatewayAttachmentId": "tgw-attach-028c1dd0f8f5cbe8e",
        "ResourceId": "vpc-01128d2c240c09bd5",
        "ResourceType": "vpc",
        "Subnets": [
            {
                "SubnetId": "subnet-000de86e3b49c932a",
                "State": "associating"
            }
        ]
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的多點傳送網域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateTransitGatewayMulticastDomain。

associate-transit-gateway-route-table

以下程式碼範例顯示如何使用 associate-transit-gateway-route-table。

AWS CLI

將傳輸閘道路由表與傳輸閘道連接建立關聯

下列範例會將指定的傳輸閘道路由表與指定的 VPC 連接建立關聯。

```
aws ec2 associate-transit-gateway-route-table \
    --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE \
    --transit-gateway-attachment-id tgw-attach-0b5968d3b6EXAMPLE
```

輸出：

```
{
    "Association": {
        "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
        "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
        "ResourceId": "vpc-0065acced4EXAMPLE",
        "ResourceType": "vpc",
        "State": "associating"
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的關聯 Transit Gateway Route Table。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateTransitGatewayRouteTable。

associate-vpc-cidr-block

以下程式碼範例顯示如何使用 associate-vpc-cidr-block。

AWS CLI

範例 1：將 Amazon 提供的 IPv6 CIDR 區塊與 VPC 建立關聯

下列associate-vpc-cidr-block範例會將 IPv6 CIDR 區塊與指定的 VPC 建立關聯：

```
aws ec2 associate-vpc-cidr-block \
    --amazon-provided-ipv6-cidr-block \
    --ipv6-cidr-block-network-border-group us-west-2-lax-1  \
    --vpc-id vpc-8EXAMPLE
```

輸出：

```
{
    "Ipv6CidrBlockAssociation": {
        "AssociationId": "vpc-cidr-assoc-0838ce7d9dEXAMPLE",
        "Ipv6CidrBlockState": {
            "State": "associating"
        },
        "NetworkBorderGroup": "us-west-2-lax-1"
    },
    "VpcId": "vpc-8EXAMPLE"
}
```

範例 2：將其他 IPv4 CIDR 區塊與 VPC 建立關聯

下列associate-vpc-cidr-block範例會將 IPv4 CIDR 區塊10.2.0.0/16與指定的 VPC 建立關聯。

```
aws ec2 associate-vpc-cidr-block \
    --vpc-id vpc-1EXAMPLE \
    --cidr-block 10.2.0.0/16
```

輸出：

```
{
    "CidrBlockAssociation": {
        "AssociationId": "vpc-cidr-assoc-2EXAMPLE",
        "CidrBlock": "10.2.0.0/16",
        "CidrBlockState": {
            "State": "associating"
        }
    },
    "VpcId": "vpc-1EXAMPLE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateVpcCidrBlock。

attach-classic-link-vpc

以下程式碼範例顯示如何使用 attach-classic-link-vpc。

AWS CLI

將 EC2-Classic 執行個體連結至（連接）VPC

此範例透過 VPC 安全群組 sg-88888888 將執行個體 i-1234567890abcdef0 連結至 VPC vpc-12312312。

命令：

```
aws ec2 attach-classic-link-vpc --instance-id  i-1234567890abcdef0 --vpc-
id vpc-88888888 --groups sg-12312312
```

輸出：

```
{
  "Return": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachClassicLinkVpc。

attach-internet-gateway

以下程式碼範例顯示如何使用 attach-internet-gateway。

AWS CLI

將網際網路閘道連接至 VPC

下列attach-internet-gateway範例會將指定的網際網路閘道連接至特定的 VPC。

```
aws ec2 attach-internet-gateway \
    --internet-gateway-id igw-0d0fb496b3EXAMPLE \
    --vpc-id vpc-0a60eb65b4EXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的網際網路閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachInternetGateway。

attach-network-interface

以下程式碼範例顯示如何使用 attach-network-interface。

AWS CLI

範例 1：將網路界面連接至執行個體

下列attach-network-interface範例會將指定的網路界面連接至指定的執行個體。

```
aws ec2 attach-network-interface \
    --network-interface-id eni-0dc56a8d4640ad10a \
    --instance-id i-1234567890abcdef0 \
    --device-index 1
```

輸出：

```
{
    "AttachmentId": "eni-attach-01a8fc87363f07cf9"
}
```

如需詳細，請參閱《Amazon EC2 使用者指南》中的彈性網絡介面。

範例 2：將網路介面連接至具有多個網路卡的執行個體

下列attach-network-interface範例會將指定的網路介面連接到指定的執行個體和網路卡。

```
aws ec2 attach-network-interface \
    --network-interface-id eni-07483b1897541ad83 \
    --instance-id i-01234567890abcdef \
    --network-card-index 1 \
    --device-index 1
```

輸出：

```
{
    "AttachmentId": "eni-attach-0fbd7ee87a88cd06c"
```

```
}
```

如需詳細，請參閱《Amazon EC2 使用者指南》中的彈性網絡介面。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachNetworkInterface。

attach-verified-access-trust-provider

以下程式碼範例顯示如何使用 attach-verified-access-trust-provider。

AWS CLI

將信任提供者連接至執行個體

下列attach-verified-access-trust-provider範例會將指定的 Verified Access 信任提供者連接至指定的 Verified Access 執行個體。

```
aws ec2 attach-verified-access-trust-provider \
    --verified-access-instance-id vai-0ce000c0b7643abea \
    --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

輸出：

```
{
    "VerifiedAccessTrustProvider": {
        "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
        "Description": "",
        "TrustProviderType": "user",
        "UserTrustProviderType": "iam-identity-center",
        "PolicyReferenceName": "idc",
        "CreationTime": "2023-08-25T19:00:38",
        "LastUpdatedTime": "2023-08-25T19:00:38"
    },
    "VerifiedAccessInstance": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "Description": "",
        "VerifiedAccessTrustProviders": [
            {
                "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
                "TrustProviderType": "user",
                "UserTrustProviderType": "iam-identity-center"
            }
```

```
        ],
        "CreationTime": "2023-08-25T18:27:56",
        "LastUpdatedTime": "2023-08-25T18:27:56"
    }
}
```

如需詳細資訊，請參閱 Verified Access 使用者指南中的 Verified Access 執行個體。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachVerifiedAccessTrustProvider。

attach-volume

以下程式碼範例顯示如何使用 attach-volume。

AWS CLI

將磁碟區連接至執行個體

此範例命令會將磁碟區 (vol-1234567890abcdef0) 連接至執行個體 (i-01474ef662b89480)，做為 /dev/sdf。

命令：

```
aws ec2 attach-volume --volume-id vol-1234567890abcdef0 --instance-
id i-01474ef662b89480 --device /dev/sdf
```

輸出：

```
{
    "AttachTime": "YYYY-MM-DDTHH:MM:SS.000Z",
    "InstanceId": "i-01474ef662b89480",
    "VolumeId": "vol-1234567890abcdef0",
    "State": "attaching",
    "Device": "/dev/sdf"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachVolume。

attach-vpn-gateway

以下程式碼範例顯示如何使用 attach-vpn-gateway。

AWS CLI

將虛擬私有閘道連接至 VPC

下列attach-vpn-gateway範例會將指定的虛擬私有閘道連接至指定的 VPC。

```
aws ec2 attach-vpn-gateway \
    --vpn-gateway-id vgw-9a4cacf3 \
    --vpc-id vpc-a01106c2
```

輸出：

```
{
    "VpcAttachment": {
        "State": "attaching",
        "VpcId": "vpc-a01106c2"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachVpnGateway。

authorize-client-vpn-ingress

以下程式碼範例顯示如何使用 authorize-client-vpn-ingress。

AWS CLI

新增 Client VPN 端點的授權規則

下列authorize-client-vpn-ingress範例新增了輸入授權規則，允許所有用戶端存取網際網路 (0.0.0.0/0)。

```
aws ec2 authorize-client-vpn-ingress \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --target-network-cidr 0.0.0.0/0 \
    --authorize-all-groups
```

輸出：

```
{
```

```
    "Status": {
        "Code": "authorizing"
    }
}
```

如需詳細資訊，請參閱 AWS Client VPN 管理員指南中的授權規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AuthorizeClientVpnIngress。

authorize-security-group-egress

以下程式碼範例顯示如何使用 authorize-security-group-egress。

AWS CLI

範例 1：新增允許傳出流量至特定地址範圍的規則

下列authorize-security-group-egress範例新增規則，授予 TCP 連接埠 80 上指定地址範圍的存取權。

```
aws ec2 authorize-security-group-egress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges=[{CidrIp=10.0.0.0/16}]'
```

輸出：

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0b15794cdb17bf29c",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": true,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "10.0.0.0/16"
        }
    ]
}
```

範例 2：新增允許傳出流量至特定安全群組的規則

下列authorize-security-group-egress範例新增規則，將存取權授予 TCP 連接埠 80 上指定的安全群組。

```
aws ec2 authorize-security-group-egress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=80,ToPort=80,UserIdGroupPairs=[{GroupId=sg-0aad1c26bbeec5c22}]'
```

輸出：

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0b5dd815afcea9cc3",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": true,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "ReferencedGroupInfo": {
                "GroupId": "sg-0aad1c26bbeec5c22",
                "UserId": "123456789012"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的安全群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AuthorizeSecurityGroupEgress。

authorize-security-group-ingress

以下程式碼範例顯示如何使用 authorize-security-group-ingress。

AWS CLI

範例 1：新增規則，以允許傳入 SSH 流量

以下 authorize-security-group-ingress 範例會新增規則，以允許 TCP 連接埠 22 (SSH) 上的傳入流量。

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --protocol tcp \
    --port 22 \
    --cidr 203.0.113.0/24
```

輸出：

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-01afa97ef3e1bedfc",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "203.0.113.0/24"
        }
    ]
}
```

範例 2：新增規則，以允許來自其他安全群組的傳入 HTTP 流量

下列 authorize-security-group-ingress 範例會新增規則，以允許 TCP 連接埠 80 上來自來源安全群組 sg-1a2b3c4d 的傳入存取。來源群組必須在相同 VPC 或對等 VPC 中 (需要 VPC 對等互連)。傳入流量會根據與來源安全群組相關聯之執行個體的私有 IP 地址允許 (而非公有 IP 地址或彈性 IP 地址)。

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --protocol tcp \
    --port 80 \
    --source-group sg-1a2b3c4d
```

輸出：

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-01f4be99110f638a7",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "ReferencedGroupInfo": {
                "GroupId": "sg-1a2b3c4d",
                "UserId": "123456789012"
            }
        }
    ]
}
```

範例 3：在相同的呼叫中新增多個規則

下列 `authorize-security-group-ingress` 範例會使用 `ip-permissions` 參數來新增兩個傳入規則；一個規則可在 TCP 連接埠 3389 (RDP) 上啟用傳入存取，另一個規則可啟用 Ping/ICMP。

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=3389,ToPort=3389,IpRanges=[{CidrIp=172.31.0.0/16}]"
 "IpProtocol=icmp,FromPort=-1,ToPort=-1,IpRanges=[{CidrIp=172.31.0.0/16}]'
```

輸出：

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-00e06e5d3690f29f3",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "tcp",
```

```
            "FromPort": 3389,
            "ToPort": 3389,
            "CidrIpv4": "172.31.0.0/16"
        },
        {
            "SecurityGroupRuleId": "sgr-0a133dd4493944b87",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": -1,
            "ToPort": -1,
            "CidrIpv4": "172.31.0.0/16"
        }
    ]
}
```

範例 4：為 ICMP 流量新增規則

下列 authorize-security-group-ingress 範例會使用 ip-permissions 參數來新增傳入規則，以允許來自任何地方的 ICMP 訊息 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set (類型 3，代碼 4)。

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=icmp,FromPort=3,ToPort=4,IpRanges=[{CidrIp=0.0.0.0/0}]'
```

輸出：

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0de3811019069b787",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "icmp",
            "FromPort": 3,
            "ToPort": 4,
            "CidrIpv4": "0.0.0.0/0"
```

```
        }
    ]
}
```

範例 5：為 IPv6 流量新增規則

下列 authorize-security-group-ingress 範例會使用 ip-permissions 參數新增傳入規則，以允許來自 IPv6 範圍 2001:db8:1234:1a00::/64 的 SSH 存取 (連接埠 22)。

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=22,ToPort=22,Ipv6Ranges=[{CidrIpv6=2001:db8:1234:1a00::/64}]'
```

輸出：

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0455bc68b60805563",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv6": "2001:db8:1234:1a00::/64"
        }
    ]
}
```

範例 6：為 ICMPv6 流量新增規則

下列 authorize-security-group-ingress 範例會使用 ip-permissions 參數來新增傳入規則，以允許來自任何地方的 ICMPv6 流量。

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions 'IpProtocol=icmpv6,Ipv6Ranges=[{CidrIpv6=::/0}]'
```

輸出：

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-04b612d9363ab6327",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "icmpv6",
            "FromPort": -1,
            "ToPort": -1,
            "CidrIpv6": "::/0"
        }
    ]
}
```

範例 7：新增具有描述的規則

下列 authorize-security-group-ingress 範例會使用 ip-permissions 參數來新增傳入規則，以允許來自指定 IPv4 地址範圍的 RDP 流量。此規則提供描述，可於稍後協助識別。

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=3389,ToPort=3389,IpRanges=[{CidrIp=203.0.113.0/24,Description='RDP
 office'}]'
```

輸出：

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0397bbcc01e974db3",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 3389,
            "ToPort": 3389,
```

```
            "CidrIpv4": "203.0.113.0/24",
            "Description": "RDP access from NY office"
        }
    ]
}
```

範例 8：新增使用字首清單的傳入規則

下列 authorize-security-group-ingress 範例會使用 ip-permissions 參數來新增傳入規則，以允許指定字首清單中 CIDR 範圍適用的所有流量。

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-04a351bfe432d4e71 \
    --ip-permissions
 'IpProtocol=all,PrefixListIds=[{PrefixListId=pl-002dc3ec097de1514}]'
```

輸出：

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-09c74b32f677c6c7c",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "-1",
            "FromPort": -1,
            "ToPort": -1,
            "PrefixListId": "pl-0721453c7ac4ec009"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的安全群組。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AuthorizeSecurityGroupIngress。

bundle-instance

以下程式碼範例顯示如何使用 bundle-instance。

AWS CLI

綁定執行個體

此範例會將執行個體綁定i-1234567890abcdef0至名為 的儲存貯體bundletasks。在您指定存取金鑰 IDs的值之前，請檢閱並遵循管理 AWS 存取金鑰最佳實務中的指引。

命令：

```
aws ec2 bundle-instance --instance-id i-1234567890abcdef0 --bucket bundletasks --
prefix winami --owner-akid AK12AJEXAMPLE --owner-sak example123example
```

輸出：

```
{
  "BundleTask": {
    "UpdateTime": "2015-09-15T13:30:35.000Z",
    "InstanceId": "i-1234567890abcdef0",
    "Storage": {
      "S3": {
        "Prefix": "winami",
        "Bucket": "bundletasks"
      }
    },
    "State": "pending",
    "StartTime": "2015-09-15T13:30:35.000Z",
    "BundleId": "bun-294e041f"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BundleInstance。

cancel-bundle-task

以下程式碼範例顯示如何使用 cancel-bundle-task。

AWS CLI

取消套件任務

此範例會取消套件任務 bun-2a4e041c。

命令：

```
aws ec2 cancel-bundle-task --bundle-id bun-2a4e041c
```

輸出：

```
{
  "BundleTask": {
    "UpdateTime": "2015-09-15T13:27:40.000Z",
    "InstanceId": "i-1234567890abcdef0",
    "Storage": {
      "S3": {
        "Prefix": "winami",
        "Bucket": "bundletasks"
      }
    },
    "State": "cancelling",
    "StartTime": "2015-09-15T13:24:35.000Z",
    "BundleId": "bun-2a4e041c"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelBundleTask。

cancel-capacity-reservation-fleets

以下程式碼範例顯示如何使用 cancel-capacity-reservation-fleets。

AWS CLI

取消容量保留機群

下列cancel-capacity-reservation-fleets範例會取消指定的容量保留機群及其保留的容量。當您取消機群時，其狀態會變更為 cancelled，且無法再建立新的容量保留。此外，機群中的所有個別容量預留都會取消，先前在預留容量中執行的執行個體仍會繼續以共用容量正常執行。

```
aws ec2 cancel-capacity-reservation-fleets \
    --capacity-reservation-fleet-ids crf-abcdef01234567890
```

輸出：

```
{
    "SuccessfulFleetCancellations": [
        {
            "CurrentFleetState": "cancelling",
            "PreviousFleetState": "active",
            "CapacityReservationFleetId": "crf-abcdef01234567890"
        }
    ],
    "FailedFleetCancellations": []
}
```

如需容量保留機群的詳細資訊，請參閱《Amazon EC2 使用者指南》中的容量保留機群。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelCapacityReservationFleets。

cancel-capacity-reservation

以下程式碼範例顯示如何使用 cancel-capacity-reservation。

AWS CLI

取消容量保留

下列cancel-capacity-reservation範例會取消指定的容量保留。

```
aws ec2 cancel-capacity-reservation \
    --capacity-reservation-id cr-1234abcd56EXAMPLE
```

輸出：

```
{
    "Return": true
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的取消容量保留。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelCapacityReservation。

cancel-conversion-task

以下程式碼範例顯示如何使用 cancel-conversion-task。

AWS CLI

取消執行個體或磁碟區的作用中轉換

此範例會取消與任務 ID import-i-fh95npoc 相關聯的上傳。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 cancel-conversion-task --conversion-task-id import-i-fh95npoc
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelConversionTask。

cancel-export-task

以下程式碼範例顯示如何使用 cancel-export-task。

AWS CLI

取消作用中的匯出任務

此範例會取消任務 ID export-i-fgelt0i7 的作用中匯出任務。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 cancel-export-task --export-task-id export-i-fgelt0i7
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelExportTask。

cancel-image-launch-permission

以下程式碼範例顯示如何使用 cancel-image-launch-permission。

AWS CLI

取消與 Amazon Web Services 帳戶共用 AMI

下列cancel-image-launch-permission範例會從指定的 AMI 啟動許可中移除您的帳戶。

```
aws ec2 cancel-image-launch-permission \
    --image-id ami-0123456789example \
```

```
    --region us-east-1
```

輸出：

```
{
    "Return": true
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的取消與 Amazon Web Services 帳戶共用 AMI。 Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelImageLaunchPermission。

cancel-import-task

以下程式碼範例顯示如何使用 cancel-import-task。

AWS CLI

取消匯入任務

下列cancel-import-task範例會取消指定的匯入映像任務。

```
aws ec2 cancel-import-task \
    --import-task-id import-ami-1234567890abcdef0
```

輸出：

```
{
    "ImportTaskId": "import-ami-1234567890abcdef0",
    "PreviousState": "active",
    "State": "deleting"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelImportTask。

cancel-reserved-instances-listing

以下程式碼範例顯示如何使用 cancel-reserved-instances-listing。

AWS CLI

取消預留執行個體清單

下列cancel-reserved-instances-listing範例會取消指定的預留執行個體清單。

```
aws ec2 cancel-reserved-instances-listing \
    --reserved-instances-listing-id 5ec28771-05ff-4b9b-aa31-9e57dexample
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelReservedInstancesListing。

cancel-spot-fleet-requests

以下程式碼範例顯示如何使用 cancel-spot-fleet-requests。

AWS CLI

範例 1：取消 Spot 機群請求並終止相關聯的執行個體

下列cancel-spot-fleet-requests範例會取消 Spot Fleet 請求，並終止相關聯的隨需執行個體和 Spot 執行個體。

```
aws ec2 cancel-spot-fleet-requests \
    --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --terminate-instances
```

輸出：

```
{
    "SuccessfulFleetRequests": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
            "CurrentSpotFleetRequestState": "cancelled_terminating",
            "PreviousSpotFleetRequestState": "active"
        }
    ],
    "UnsuccessfulFleetRequests": []
}
```

範例 2：取消 Spot 機群請求而不終止相關聯的執行個體

下列cancel-spot-fleet-requests範例會取消 Spot Fleet 請求，而不終止相關聯的隨需執行個體和 Spot 執行個體。

```
aws ec2 cancel-spot-fleet-requests \
    --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --no-terminate-instances
```

輸出：

```
{
    "SuccessfulFleetRequests": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
            "CurrentSpotFleetRequestState": "cancelled_running",
            "PreviousSpotFleetRequestState": "active"
        }
    ],
    "UnsuccessfulFleetRequests": []
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的取消 Spot 機群請求。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelSpotFleetRequests。

cancel-spot-instance-requests

以下程式碼範例顯示如何使用 cancel-spot-instance-requests。

AWS CLI

取消 Spot 執行個體請求

此範例命令會取消 Spot 執行個體請求。

命令：

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

輸出：

```
{
```

```
    "CancelledSpotInstanceRequests": [
        {
            "State": "cancelled",
            "SpotInstanceRequestId": "sir-08b93456"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelSpotInstanceRequests。

`confirm-product-instance`

以下程式碼範例顯示如何使用 `confirm-product-instance`。

AWS CLI

確認產品執行個體

此範例會判斷指定的產品程式碼是否與指定的執行個體相關聯。

命令：

```
aws ec2 confirm-product-instance --product-code 774F4FF8 --instance-
id i-1234567890abcdef0
```

輸出：

```
{
  "OwnerId": "123456789012"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ConfirmProductInstance。

`copy-fpga-image`

以下程式碼範例顯示如何使用 `copy-fpga-image`。

AWS CLI

複製 Amazon FPGA 映像

此範例會將指定的 AFI 從 us-east-1區域複製到目前區域 (eu-west-1)。

命令：

```
aws ec2 copy-fpga-image --name copy-afi --source-fpga-image-id afi-0d123e123bfc85abc
 --source-region us-east-1 --region eu-west-1
```

輸出：

```
{
  "FpgaImageId": "afi-06b12350a123fbabc"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CopyFpgaImage。

copy-image

以下程式碼範例顯示如何使用 copy-image。

AWS CLI

範例 1：將 AMI 複製到另一個區域

下列copy-image範例命令會將指定的 AMI 從 us-west-2 區域複製到 us-east-1區域，並新增簡短的描述。

```
aws ec2 copy-image \
    --region us-east-1 \
    --name ami-name \
    --source-region us-west-2 \
    --source-image-id ami-066877671789bd71b \
    --description "This is my copied image."
```

輸出：

```
{
    "ImageId": "ami-0123456789abcdefg"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的複製 AMI。

範例 2：將 AMI 複製到另一個區域並加密備份快照

下列copy-image命令會將指定的 AMI 從 us-west-2 區域複製到目前的 區域，並使用指定的 KMS 金鑰加密備份快照。

```
aws ec2 copy-image \
    --source-region us-west-2 \
    --name ami-name \
    --source-image-id ami-066877671789bd71b \
    --encrypted \
    --kms-key-id alias/my-kms-key
```

輸出：

```
{
    "ImageId": "ami-0123456789abcdefg"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的複製 AMI。

範例 3：複製 AMI 時包含使用者定義的 AMI 標籤

下列copy-image命令使用 --copy-image-tags 參數，在複製 AMI 時複製使用者定義的 AMI 標籤。

```
aws ec2 copy-image \
    --region us-east-1 \
    --name ami-name \
    --source-region us-west-2 \
    --source-image-id ami-066877671789bd71b \
    --description "This is my copied image."
    --copy-image-tags
```

輸出：

```
{
    "ImageId": "ami-0123456789abcdefg"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的複製 AMI。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CopyImage。

copy-snapshot

以下程式碼範例顯示如何使用 copy-snapshot。

AWS CLI

範例 1：將快照複製到另一個區域

下列copy-snapshot範例命令會將指定的快照從 us-west-2 區域複製到 us-east-1區域，並新增簡短的描述。

```
aws ec2 copy-snapshot \
    --region us-east-1 \
    --source-region us-west-2 \
    --source-snapshot-id snap-066877671789bd71b \
    --description 'This is my copied snapshot.'
```

輸出：

```
{
    "SnapshotId": "snap-066877671789bd71b"
}
```

範例 2：複製未加密快照並加密新快照

下列copy-snapshot命令會將指定的未加密快照從 us-west-2區域複製到目前的 區域，並使用指定的 KMS 金鑰加密新的快照。

```
aws ec2 copy-snapshot \
    --source-region us-west-2 \
    --source-snapshot-id snap-066877671789bd71b \
    --encrypted \
    --kms-key-id alias/my-kms-key
```

輸出：

```
{
    "SnapshotId": "snap-066877671789bd71b"
}
```

如需詳細資訊，請參閱《Amazon EBS 使用者指南》中的複製 Amazon EBS 快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CopySnapshot。

create-capacity-reservation-fleet

以下程式碼範例顯示如何使用 create-capacity-reservation-fleet。

AWS CLI

建立容量保留機群

下列create-capacity-reservation-fleet範例會為請求中指定的執行個體類型建立容量保留機群，最多可達指定的總目標容量。容量預留機群預留容量的執行個體數量，取決於您在請求中指定的總目標容量和執行個體類型權重。指定要使用的執行個體類型，以及每個指定執行個體類型的優先順序。

```
aws ec2 create-capacity-reservation-fleet \
--total-target-capacity 24 \
--allocation-strategy prioritized \
--instance-match-criteria open \
--tenancy default \
--end-date 2022-12-31T23:59:59.000Z \
--instance-type-specifications file://instanceTypeSpecification.json
```

instanceTypeSpecification.json 的內容：

```
[
    {
        "InstanceType": "m5.xlarge",
        "InstancePlatform": "Linux/UNIX",
        "Weight": 3.0,
        "AvailabilityZone":"us-east-1a",
        "EbsOptimized": true,
        "Priority" : 1
    }
]
```

輸出：

```
{
    "Status": "submitted",
    "TotalFulfilledCapacity": 0.0,
```

```
    "CapacityReservationFleetId": "crf-abcdef01234567890",
    "TotalTargetCapacity": 24
}
```

如需容量保留機群的詳細資訊，請參閱《Amazon EC2 使用者指南》中的容量保留機群。

如需執行個體類型權重和總目標容量的詳細資訊，請參閱《Amazon EC2 使用者指南》中的執行個體類型權重和總目標容量。

如需指定指定指定執行個體類型優先順序的詳細資訊，請參閱《Amazon EC2 使用者指南》中的配置策略和執行個體類型優先順序。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCapacityReservationFleet。

`create-capacity-reservation`

以下程式碼範例顯示如何使用 `create-capacity-reservation`。

AWS CLI

範例 1：建立容量保留

下列`create-capacity-reservation`範例會在`eu-west-1a`可用區域中建立容量保留，您可以在其中啟動三個執行 Linux/Unix 作業系統的`t2.medium`執行個體。根據預設，容量保留是使用開啟的執行個體比對條件建立，且不支援暫時性儲存，且會保持作用中狀態，直到您手動取消為止。

```
aws ec2 create-capacity-reservation \
    --availability-zone eu-west-1a \
    --instance-type t2.medium \
    --instance-platform Linux/UNIX \
    --instance-count 3
```

輸出：

```
{
    "CapacityReservation": {
        "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
        "EndDateType": "unlimited",
        "AvailabilityZone": "eu-west-1a",
        "InstanceMatchCriteria": "open",
        "EphemeralStorage": false,
```

```
        "CreateDate": "2019-08-16T09:27:35.000Z",
        "AvailableInstanceCount": 3,
        "InstancePlatform": "Linux/UNIX",
        "TotalInstanceCount": 3,
        "State": "active",
        "Tenancy": "default",
        "EbsOptimized": false,
        "InstanceType": "t2.medium"
    }
}
```

範例 2：建立在指定日期/時間自動結束的容量保留

下列create-capacity-reservation範例會在eu-west-1a可用區域中建立容量保留，您可以在其中啟動三個執行 Linux/Unix 作業系統的m5.large執行個體。此容量保留會在 08/31/2019 23：59：59 自動結束。

```
aws ec2 create-capacity-reservation \
    --availability-zone eu-west-1a \
    --instance-type m5.large \
    --instance-platform Linux/UNIX \
    --instance-count 3 \
    --end-date-type limited \
    --end-date 2019-08-31T23:59:59Z
```

輸出：

```
{
    "CapacityReservation": {
        "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
        "EndDateType": "limited",
        "AvailabilityZone": "eu-west-1a",
        "EndDate": "2019-08-31T23:59:59.000Z",
        "InstanceMatchCriteria": "open",
        "EphemeralStorage": false,
        "CreateDate": "2019-08-16T10:15:53.000Z",
        "AvailableInstanceCount": 3,
        "InstancePlatform": "Linux/UNIX",
        "TotalInstanceCount": 3,
        "State": "active",
        "Tenancy": "default",
        "EbsOptimized": false,
```

```
        "InstanceType": "m5.large"
    }
}
```

範例 3：建立只接受目標執行個體啟動的容量保留

下列create-capacity-reservation範例會建立只接受目標執行個體啟動的容量保留。

```
aws ec2 create-capacity-reservation \
    --availability-zone eu-west-1a \
    --instance-type m5.large \
    --instance-platform Linux/UNIX \
    --instance-count 3 \
    --instance-match-criteria targeted
```

輸出：

```
{
    "CapacityReservation": {
        "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
        "EndDateType": "unlimited",
        "AvailabilityZone": "eu-west-1a",
        "InstanceMatchCriteria": "targeted",
        "EphemeralStorage": false,
        "CreateDate": "2019-08-16T10:21:57.000Z",
        "AvailableInstanceCount": 3,
        "InstancePlatform": "Linux/UNIX",
        "TotalInstanceCount": 3,
        "State": "active",
        "Tenancy": "default",
        "EbsOptimized": false,
        "InstanceType": "m5.large"
    }
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的建立容量保留。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCapacityReservation。

create-carrier-gateway

以下程式碼範例顯示如何使用 create-carrier-gateway。

AWS CLI

建立電信業者閘道

下列create-carrier-gateway範例會為指定的 VPC 建立電信業者閘道。

```
aws ec2 create-carrier-gateway \
    --vpc-id vpc-0c529aEXAMPLE1111
```

輸出：

```
{
    "CarrierGateway": {
        "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
        "VpcId": "vpc-0c529aEXAMPLE1111",
        "State": "pending",
        "OwnerId": "123456789012"
    }
}
```

如需詳細資訊，請參閱《AWS Wavelength 使用者指南》中的電信業者閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCarrierGateway。

create-client-vpn-endpoint

以下程式碼範例顯示如何使用 create-client-vpn-endpoint。

AWS CLI

建立 Client VPN 端點

下列create-client-vpn-endpoint範例會建立使用交互身分驗證的 Client VPN 端點，並指定用戶端 CIDR 區塊的值。

```
aws ec2 create-client-vpn-endpoint \
    --client-cidr-block "172.31.0.0/16" \
    --server-certificate-arn arn:aws:acm:ap-south-1:123456789012:certificate/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --authentication-options Type=certificate-
authentication,MutualAuthentication={ClientRootCertificateChainArn=arn:aws:acm:ap-
south-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE} \
```

```
    --connection-log-options Enabled=false
```

輸出：

```
{
    "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
    "Status": {
        "Code": "pending-associate"
    },
    "DnsName": "cvpn-endpoint-123456789123abcde.prod.clientvpn.ap-
south-1.amazonaws.com"
}
```

如需詳細資訊，請參閱 Client VPN 管理員指南中的 Client VPN 端點。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateClientVpnEndpoint。

`create-client-vpn-route`

以下程式碼範例顯示如何使用 `create-client-vpn-route`。

AWS CLI

建立 Client VPN 端點的路由

下列`create-client-vpn-route`範例會將路由新增至 Client VPN 端點指定子網路的網際網路 (`0.0.0.0/0`)。

```
aws ec2 create-client-vpn-route \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --destination-cidr-block 0.0.0.0/0  \
    --target-vpc-subnet-id subnet-0123456789abcabca
```

輸出：

```
{
    "Status": {
        "Code": "creating"
    }
}
```

如需詳細資訊，請參閱 AWS Client VPN 管理員指南中的路由。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateClientVpnRoute。

create-coip-cidr

以下程式碼範例顯示如何使用 create-coip-cidr。

AWS CLI

建立客戶擁有的 IP (CoIP) 地址範圍

下列create-coip-cidr範例會在指定的 CoIP 集區中建立指定的 CoIP 地址範圍。

```
aws ec2 create-coip-cidr \
    --cidr 15.0.0.0/24 \
    --coip-pool-id ipv4pool-coip-1234567890abcdefg
```

輸出：

```
{
    "CoipCidr": {
        "Cidr": "15.0.0.0/24",
        "CoipPoolId": "ipv4pool-coip-1234567890abcdefg",
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890"
    }
}
```

如需詳細資訊，請參閱 AWS Outposts 使用者指南中的客戶擁有的 IP 地址。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCoipCidr。

create-coip-pool

以下程式碼範例顯示如何使用 create-coip-pool。

AWS CLI

建立客戶擁有的 IP (CoIP) 地址集區

下列create-coip-pool範例會為指定本機閘道路由表中的 CoIP 地址建立 CoIP CoIP 集區。

```
aws ec2 create-coip-pool \
```

```
    --local-gateway-route-table-id lgw-rtb-abcdefg1234567890
```

輸出：

```
{
    "CoipPool": {
        "PoolId": "ipv4pool-coip-1234567890abcdefg",
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
        "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-
coip-1234567890abcdefg"
    }
}
```

如需詳細資訊，請參閱 AWS Outposts 使用者指南中的客戶擁有的 IP 地址。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCoipPool。

create-customer-gateway

以下程式碼範例顯示如何使用 create-customer-gateway。

AWS CLI

建立客戶閘道

此範例會建立具有其外部界面之指定 IP 地址的客戶閘道。

命令：

```
aws ec2 create-customer-gateway --type ipsec.1 --public-ip 12.1.2.3 --bgp-asn 65534
```

輸出：

```
{
    "CustomerGateway": {
        "CustomerGatewayId": "cgw-0e11f167",
        "IpAddress": "12.1.2.3",
        "State": "available",
        "Type": "ipsec.1",
        "BgpAsn": "65534"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCustomerGateway。

create-default-subnet

以下程式碼範例顯示如何使用 create-default-subnet。

AWS CLI

建立預設子網路

此範例會在可用區域 中建立預設子網路us-east-2a。

命令：

```
aws ec2 create-default-subnet --availability-zone us-east-2a

{
   "Subnet": {
       "AvailabilityZone": "us-east-2a",
       "Tags": [],
       "AvailableIpAddressCount": 4091,
       "DefaultForAz": true,
       "Ipv6CidrBlockAssociationSet": [],
       "VpcId": "vpc-1a2b3c4d",
       "State": "available",
       "MapPublicIpOnLaunch": true,
       "SubnetId": "subnet-1122aabb",
       "CidrBlock": "172.31.32.0/20",
       "AssignIpv6AddressOnCreation": false
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDefaultSubnet。

create-default-vpc

以下程式碼範例顯示如何使用 create-default-vpc。

AWS CLI

建立預設 VPC

此範例會建立預設 VPC。

命令：

```
aws ec2 create-default-vpc
```

輸出：

```
{
   "Vpc": {
       "VpcId": "vpc-8eaae5ea",
       "InstanceTenancy": "default",
       "Tags": [],
       "Ipv6CidrBlockAssociationSet": [],
       "State": "pending",
       "DhcpOptionsId": "dopt-af0c32c6",
       "CidrBlock": "172.31.0.0/16",
       "IsDefault": true
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDefaultVpc。

create-dhcp-options

以下程式碼範例顯示如何使用 create-dhcp-options。

AWS CLI

建立一組 DHCP 選項

下列create-dhcp-options範例會建立一組 DHCP 選項，指定網域名稱、網域名稱伺服器和 NetBIOS 節點類型。

```
aws ec2 create-dhcp-options \
    --dhcp-configuration \
        "Key=domain-name-servers,Values=10.2.5.1,10.2.5.2" \
        "Key=domain-name,Values=example.com" \
        "Key=netbios-node-type,Values=2"
```

輸出：

```
{
```

```
    "DhcpOptions": {
        "DhcpConfigurations": [
            {
                "Key": "domain-name",
                "Values": [
                    {
                        "Value": "example.com"
                    }
                ]
            },
            {
                "Key": "domain-name-servers",
                "Values": [
                    {
                        "Value": "10.2.5.1"
                    },
                    {
                        "Value": "10.2.5.2"
                    }
                ]
            },
            {
                "Key": "netbios-node-type",
                "Values": [
                    {
                        "Value": "2"
                    }
                ]
            }
        ],
        "DhcpOptionsId": "dopt-06d52773eff4c55f3"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDhcpOptions。

create-egress-only-internet-gateway

以下程式碼範例顯示如何使用 create-egress-only-internet-gateway。

AWS CLI

建立輸出限定網際網路閘道

此範例會為指定的 VPC 建立輸出限定網際網路閘道。

命令：

```
aws ec2 create-egress-only-internet-gateway --vpc-id vpc-0c62a468
```

輸出：

```
{
  "EgressOnlyInternetGateway": {
      "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
      "Attachments": [
          {
              "State": "attached",
              "VpcId": "vpc-0c62a468"
          }
      ]
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateEgressOnlyInternetGateway。

create-fleet

以下程式碼範例顯示如何使用 create-fleet。

AWS CLI

範例 1：建立啟動 Spot 執行個體做為預設購買模型的 EC2 機群

下列create-fleet範例使用啟動機群所需的最低參數來建立 EC2 機群：啟動範本、目標容量和預設購買模型。啟動範本由其啟動範本 ID 和版本編號識別。機群的目標容量為 2 個執行個體，預設購買模式為 spot，這會導致機群啟動 2 個 Spot 執行個體。

當您建立 EC2 機群時，請使用 JSON 檔案指定要啟動的執行個體相關資訊。

```
aws ec2 create-fleet \
    --cli-input-json file://file_name.json
```

file_name.json 的內容：

```
{
    "LaunchTemplateConfigs": [
    {
        "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
        }
    }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "spot"
    }
}
```

輸出：

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

範例 2：建立 EC2 機群，以啟動隨需執行個體做為預設購買模式

下列create-fleet範例使用啟動機群所需的最低參數來建立 EC2 機群：啟動範本、目標容量和預設購買模型。啟動範本由其啟動範本 ID 和版本編號識別。機群的目標容量為 2 個執行個體，預設購買模式為 on-demand，這會導致機群啟動 2 個隨需執行個體。

當您建立 EC2 機群時，請使用 JSON 檔案指定要啟動的執行個體相關資訊。

```
aws ec2 create-fleet \
    --cli-input-json file://file_name.json
```

file_name.json 的內容：

```
{
    "LaunchTemplateConfigs": [
    {
        "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
        }
```

```
    }
    ],
    "TargetCapacitySpecification": {
    "TotalTargetCapacity": 2,
    "DefaultTargetCapacityType": "on-demand"
    }
}
```

輸出：

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

範例 3：建立啟動隨需執行個體做為主要容量的 EC2 機群

下列create-fleet範例會建立 EC2 機群，指定機群 2 個執行個體的總目標容量，以及 1 個隨需執行個體的目標容量。預設購買模型為 spot。機群會依指定啟動 1 個隨需執行個體，但需要再啟動一個執行個體才能滿足總目標容量。差異的購買模型計算方式為 TotalTargetCapacity - OnDemandTargetCapacity = DefaultTargetCapacityType，這會導致機群啟動 1 個 Spot 執行個體。

當您建立 EC2 機群時，請使用 JSON 檔案指定要啟動的執行個體相關資訊。

```
aws ec2 create-fleet \
    --cli-input-json file://file_name.json
```

file_name.json 的內容：

```
{
    "LaunchTemplateConfigs": [
    {
        "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
        }
    }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "OnDemandTargetCapacity":1,
```

```
        "DefaultTargetCapacityType": "spot"
    }
}
```

輸出：

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

範例 4：建立使用最低價格配置策略啟動 Spot 執行個體的 EC2 機群

如果未指定 競價型執行個體 的分配策略，則會使用 `lowest-price` 的預設分配策略。下列`create-fleet`範例使用`lowest-price`配置策略建立 EC2 機群。覆寫啟動範本的三個啟動規格具有不同的執行個體類型，但具有相同的權重後容量和子網。總目標容量為 2 個執行個體，預設購買模式為 `spot`。EC2 Fleet 會以最低價格之啟動規格的執行個體類型啟動 2 個 競價型執行個體。

當您建立 EC2 機群時，請使用 JSON 檔案指定要啟動的執行個體相關資訊。

```
aws ec2 create-fleet \
    --cli-input-json file://file_name.jsonContents of file_name.json::

{
    "LaunchTemplateConfigs": [
    {
        "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
        },
        "Overrides": [
            {
                "InstanceType": "c4.large",
                "WeightedCapacity": 1,
                "SubnetId": "subnet-a4f6c5d3"
            },
            {
                "InstanceType": "c3.large",
                "WeightedCapacity": 1,
                "SubnetId": "subnet-a4f6c5d3"
            },
            {
```

```
                "InstanceType": "c5.large",
                "WeightedCapacity": 1,
                "SubnetId": "subnet-a4f6c5d3"
            }
        ]
    }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "spot"

  }
}
```

輸出：

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFleet。

create-flow-logs

以下程式碼範例顯示如何使用 create-flow-logs。

AWS CLI

範例 1：建立流程日誌

下列create-flow-logs範例會建立流程日誌，擷取指定網路介面的所有拒絕流量。流程日誌會使用指定 IAM 角色中的許可，傳送至 CloudWatch Logs 中的日誌群組。

```
aws ec2 create-flow-logs \
    --resource-type NetworkInterface \
    --resource-ids eni-11223344556677889 \
    --traffic-type REJECT \
    --log-group-name my-flow-logs \
    --deliver-logs-permission-arn arn:aws:iam::123456789101:role/publishFlowLogs
```

輸出：

```
{
    "ClientToken": "so0eNA2uSHUNlHI0S2cJ305GuIX1CezaRdGtexample",
    "FlowLogIds": [
        "fl-12345678901234567"
    ],
    "Unsuccessful": []
}
```

如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的 VPC 流程日誌。

範例 2：使用自訂格式建立流程日誌

下列create-flow-logs範例會建立流程日誌，擷取指定 VPC 的所有流量，並將流程日誌交付至 Amazon S3 儲存貯體。--log-format 參數會指定流量日誌記錄的自訂格式。若要在 Windows 上執行此命令，請將單引號 (') 變更為雙引號 (")。

```
aws ec2 create-flow-logs \
    --resource-type VPC \
    --resource-ids vpc-00112233344556677 \
    --traffic-type ALL \
    --log-destination-type s3 \
    --log-destination arn:aws:s3:::flow-log-bucket/my-custom-flow-logs/ \
    --log-format '${version} ${vpc-id} ${subnet-id} ${instance-id} ${srcaddr}
 ${dstaddr} ${srcport} ${dstport} ${protocol} ${tcp-flags} ${type} ${pkt-srcaddr}
 ${pkt-dstaddr}'
```

如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的 VPC 流程日誌。

範例 3：建立具有一分鐘最大彙總間隔的流程日誌

下列create-flow-logs範例會建立流程日誌，擷取指定 VPC 的所有流量，並將流程日誌交付至 Amazon S3 儲存貯體。--max-aggregation-interval 參數指定最大彙總間隔為 60 秒 (1 分鐘) 。

```
aws ec2 create-flow-logs \
    --resource-type VPC \
    --resource-ids vpc-00112233344556677 \
    --traffic-type ALL \
    --log-destination-type s3 \
    --log-destination arn:aws:s3:::flow-log-bucket/my-custom-flow-logs/ \
    --max-aggregation-interval 60
```

如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的 VPC 流程日誌。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFlowLogs。

create-fpga-image

以下程式碼範例顯示如何使用 create-fpga-image。

AWS CLI

建立 Amazon FPGA 映像

此範例會從指定儲存貯體中指定的 tarball 建立 AFI。

命令：

```
aws ec2 create-fpga-image --name my-afi --description test-afi --input-storage-location Bucket=my-fpga-bucket,Key=dcp/17_12_22-103226.Developer_CL.tar --logs-storage-location Bucket=my-fpga-bucket,Key=logs
```

輸出：

```
{
  "FpgaImageId": "afi-0d123e123bfc85abc",
  "FpgaImageGlobalId": "agfi-123cb27b5e84a0abc"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFpgaImage。

create-image

以下程式碼範例顯示如何使用 create-image。

AWS CLI

範例 1：從 Amazon EBS 支援的執行個體建立 AMI

下列create-image範例會從指定的執行個體建立 AMI。

```
aws ec2 create-image \
```

```
    --instance-id i-1234567890abcdef0 \
    --name "My server" \
    --description "An AMI for my server"
```

輸出：

```
{
    "ImageId": "ami-abcdef01234567890"
}
```

如需為 AMI 指定區塊型設備映射的詳細資訊，請參閱《Amazon EC2 使用者指南》中的為 AMI 指定區塊型設備映射。

範例 2：從 Amazon EBS 後端執行個體建立 AMI，而不重新啟動

下列create-image範例會建立 AMI 並設定 --no-reboot 參數，讓執行個體不會在建立映像之前重新啟動。

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "My server" \
    --no-reboot
```

輸出：

```
{
    "ImageId": "ami-abcdef01234567890"
}
```

如需為 AMI 指定區塊型設備映射的詳細資訊，請參閱《Amazon EC2 使用者指南》中的為 AMI 指定區塊型設備映射。

範例 3：在建立時標記 AMI 和快照

下列create-image範例會建立 AMI，並使用相同的標籤來標記 AMI 和快照 cost-center=cc123

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "My server" \
```

```
    --tag-specifications "ResourceType=image,Tags=[{Key=cost-
center,Value=cc123}]" "ResourceType=snapshot,Tags=[{Key=cost-center,Value=cc123}]"
```

輸出：

```
{
    "ImageId": "ami-abcdef01234567890"
}
```

如需在建立時標記資源的詳細資訊，請參閱《Amazon EC2 使用者指南》中的在資源建立時新增標籤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateImage。

create-instance-connect-endpoint

以下程式碼範例顯示如何使用 create-instance-connect-endpoint。

AWS CLI

建立 EC2 Instance Connect Endpoint

下列create-instance-connect-endpoint範例會在指定的子網路中建立 EC2 Instance Connect Endpoint。

```
aws ec2 create-instance-connect-endpoint \
    --region us-east-1 \
    --subnet-id subnet-0123456789example
```

輸出：

```
{
    "VpcId": "vpc-0123abcd",
    "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-
connect-endpoint/eice-0123456789example",
    "AvailabilityZone": "us-east-1a",
    "NetworkInterfaceIds": [
        "eni-0123abcd"
    ],
    "PreserveClientIp": true,
    "Tags": [],
```

```
    "FipsDnsName": "eice-0123456789example.0123abcd.fips.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com",
    "StateMessage": "",
    "State": "create-complete",
    "DnsName": "eice-0123456789example.0123abcd.ec2-instance-connect-endpoint.us-
east-1.amazonaws.com",
    "SubnetId": "subnet-0123abcd",
    "OwnerId": "111111111111",
    "SecurityGroupIds": [
        "sg-0123abcd"
    ],
    "InstanceConnectEndpointId": "eice-0123456789example",
    "CreatedAt": "2023-04-07T15:43:53.000Z"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的建立 EC2 Instance Connect Endpoint。Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateInstanceConnectEndpoint。

create-instance-event-window

以下程式碼範例顯示如何使用 create-instance-event-window。

AWS CLI

範例 1：建立具有時間範圍的事件時段

下列create-instance-event-window範例會建立具有時間範圍的事件時段。您不能指定 cron-expression 參數。

```
aws ec2 create-instance-event-window \
    --region us-east-1 \
    --time-range StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8 \
    --tag-specifications "ResourceType=instance-event-
window,Tags=[{Key=K1,Value=V1}]" \
    --name myEventWindowName
```

輸出：

```
{
    "InstanceEventWindow": {
```

```
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "TimeRanges": [
            {
                "StartWeekDay": "monday",
                "StartHour": 2,
                "EndWeekDay": "wednesday",
                "EndHour": 8
            }
        ],
        "Name": "myEventWindowName",
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

如需事件時段限制，請參閱《Amazon EC2 使用者指南》的排程事件一節中的考量事項。

範例 2：使用 Cron 表達式建立事件時段

下列create-instance-event-window範例會建立具有 Cron 表達式的事件視窗。您不能指定 time-range 參數。

```
aws ec2 create-instance-event-window \
    --region us-east-1 \
    --cron-expression "* 21-23 * * 2,3" \
    --tag-specifications "ResourceType=instance-event-
window,Tags=[{Key=K1,Value=V1}]" \
    --name myEventWindowName
```

輸出：

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "State": "creating",
```

```
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

如需事件時段限制，請參閱《Amazon EC2 使用者指南》的排程事件一節中的考量事項。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateInstanceEventWindow。

create-instance-export-task

以下程式碼範例顯示如何使用 create-instance-export-task。

AWS CLI

匯出執行個體

此範例命令會建立任務，將執行個體 i-1234567890abcdef0 匯出至 Amazon S3 儲存貯體 myexportbucket。

命令：

```
aws ec2 create-instance-export-task --description "RHEL5 instance" --
instance-id i-1234567890abcdef0 --target-environment vmware --export-to-s3-
task DiskImageFormat=vmdk,ContainerFormat=ova,S3Bucket=myexportbucket,S3Prefix=RHEL5
```

輸出：

```
{
    "ExportTask": {
        "State": "active",
        "InstanceExportDetails": {
            "InstanceId": "i-1234567890abcdef0",
            "TargetEnvironment": "vmware"
        },
        "ExportToS3Task": {
            "S3Bucket": "myexportbucket",
            "S3Key": "RHEL5export-i-fh8sjjsq.ova",
```

```
            "DiskImageFormat": "vmdk",
            "ContainerFormat": "ova"
        },
        "Description": "RHEL5 instance",
        "ExportTaskId": "export-i-fh8sjjsq"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateInstanceExportTask。

create-internet-gateway

以下程式碼範例顯示如何使用 create-internet-gateway。

AWS CLI

建立網際網路閘道

下列create-internet-gateway範例會建立標籤為 的網際網路閘道Name=my-igw。

```
aws ec2 create-internet-gateway \
    --tag-specifications ResourceType=internet-gateway,Tags=[{Key=Name,Value=my-igw}]
```

輸出：

```
{
    "InternetGateway": {
        "Attachments": [],
        "InternetGatewayId": "igw-0d0fb496b3994d755",
        "OwnerId": "123456789012",
        "Tags": [
            {
                "Key": "Name",
                "Value": "my-igw"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的網際網路閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateInternetGateway。

create-ipam-pool

以下程式碼範例顯示如何使用 create-ipam-pool。

AWS CLI

建立 IPAM 集區

下列create-ipam-pool範例會建立 IPAM 集區。

(Linux)：

```
aws ec2 create-ipam-pool \
    --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 \
    --address-family ipv4 \
    --auto-import \
    --allocation-min-netmask-length 16 \
    --allocation-max-netmask-length 26 \
    --allocation-default-netmask-length 24 \
    --allocation-resource-tags "Key=Environment,Value=Preprod" \
    --tag-specifications 'ResourceType=ipam-pool,Tags=[{Key=Name,Value="Preprod
 pool"}]'
```

(Windows)：

```
aws ec2 create-ipam-pool ^
    --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 ^
    --address-family ipv4 ^
    --auto-import ^
    --allocation-min-netmask-length 16 ^
    --allocation-max-netmask-length 26 ^
    --allocation-default-netmask-length 24 ^
    --allocation-resource-tags "Key=Environment,Value=Preprod" ^
    --tag-specifications ResourceType=ipam-pool,Tags=[{Key=Name,Value="Preprod
 pool"}]
```

輸出：

```
{
    "IpamPool": {
```

```
        "OwnerId": "123456789012",
        "IpamPoolId": "ipam-pool-0533048da7d823723",
        "IpamPoolArn": "arn:aws:ec2::123456789012:ipam-pool/ipam-
pool-0533048da7d823723",
        "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-02fc38cd4c48e7d38",
        "IpamScopeType": "private",
        "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
        "IpamRegion": "us-east-1",
        "Locale": "None",
        "PoolDepth": 1,
        "State": "create-in-progress",
        "AutoImport": true,
        "AddressFamily": "ipv4",
        "AllocationMinNetmaskLength": 16,
        "AllocationMaxNetmaskLength": 26,
        "AllocationDefaultNetmaskLength": 24,
        "AllocationResourceTags": [
            {
                "Key": "Environment",
                "Value": "Preprod"
            }
        ],
        "Tags": [
            {
                "Key": "Name",
                "Value": "Preprod pool"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的規劃 IP 地址佈建。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateIpamPool。

create-ipam-resource-discovery

以下程式碼範例顯示如何使用 create-ipam-resource-discovery。

AWS CLI

建立資源探索

在此範例中，您是委派 IPAM 管理員，想要建立資源探索並與另一個 AWS 組織中的 IPAM 管理員共用，以便其他組織中的管理員可以管理和監控組織中資源的 IP 地址。

重要

此範例同時包含 --region和 --operating-regions選項，因為雖然它們是選用的，但必須以特定方式設定它們，才能成功整合資源探索與 IPAM。* --operating-regions 必須符合您要 IPAM 探索資源的區域。如果有您不希望 IPAM 管理 IP 地址的區域（例如，基於合規原因），請不要包含這些地址。* --region必須符合您要與之建立關聯的 IPAM 主區域。您必須在建立 IPAM 的相同區域中建立資源探索。例如，如果您要與之建立關聯的 IPAM 是在 us-east-1 中建立的，請在 請求--region us-east-1中包含 。如果您未指定，則 --region和 --operating-regions選項都會預設為您執行命令的區域。

在此範例中，與我們整合的 IPAM 操作區域包括 us-west-1、 us-west-2和 ap-south-1。當我們建立資源探索時，我們希望 IPAM 探索 us-west-1和 中的資源 IP 地址，us-west-2而不是 ap-south-1。因此，我們只會在請求--operating-regions RegionName='us-west-1' RegionName='us-west-2'中包含 。

下列create-ipam-resource-discovery範例會建立 IPAM 資源探索。

```
aws ec2 create-ipam-resource-discovery \
    --description 'Example-resource-discovery' \
    --tag-specifications 'ResourceType=ipam-resource-discovery,Tags=[{Key=cost-
center,Value=cc123}]' \
    --operating-regions RegionName='us-west-1' RegionName='us-west-2' \
    --region us-east-1
```

輸出：

```
{
    "IpamResourceDiscovery":{
        "OwnerId": "149977607591",
        "IpamResourceDiscoveryId": "ipam-res-disco-0257046d8aa78b8bc",
        "IpamResourceDiscoveryArn": "arn:aws:ec2::149977607591:ipam-resource-
discovery/ipam-res-disco-0257046d8aa78b8bc",
        "IpamResourceDiscoveryRegion": "us-east-1",
        "Description": "'Example-resource-discovery'",
        "OperatingRegions":[
            {"RegionName": "us-west-1"},
            {"RegionName": "us-west-2"},
            {"RegionName": "us-east-1"}
```

```
        ],
        "IsDefault": false,
        "State": "create-in-progress",
        "Tags": [
            {
                "Key": "cost-center",
                "Value": "cc123"
            }
        ]
}
```

建立資源探索之後，您可能想要與另一個 IPAM 委派管理員共用，您可以使用 create-resource-share 來執行此操作。如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的將 IPAM 與組織外部的帳戶整合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateIpamResourceDiscovery。

`create-ipam-scope`

以下程式碼範例顯示如何使用 `create-ipam-scope`。

AWS CLI

建立 IPAM 範圍

下列`create-ipam-scope`範例會建立 IPAM 範圍。

(Linux)：

```
aws ec2 create-ipam-scope \
    --ipam-id ipam-08440e7a3acde3908 \
    --description "Example description" \
    --tag-specifications 'ResourceType=ipam-scope,Tags=[{Key=Name,Value="Example
 name value"}]'
```

(Windows)：

```
aws ec2 create-ipam-scope ^
    --ipam-id ipam-08440e7a3acde3908 ^
    --description "Example description" ^
    --tag-specifications ResourceType=ipam-scope,Tags=[{Key=Name,Value="Example name
 value"}]
```

輸出：

```
{
    "IpamScope": {
        "OwnerId": "123456789012",
        "IpamScopeId": "ipam-scope-01c1ebab2b63bd7e4",
        "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-01c1ebab2b63bd7e4",
        "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
        "IpamRegion": "us-east-1",
        "IpamScopeType": "private",
        "IsDefault": false,
        "Description": "Example description",
        "PoolCount": 0,
        "State": "create-in-progress",
        "Tags": [
            {
                "Key": "Name",
                "Value": "Example name value"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的建立其他範圍。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateIpamScope。

create-ipam

以下程式碼範例顯示如何使用 create-ipam。

AWS CLI

建立 IPAM

下列create-ipam範例會建立 IPAM。

(Linux)：

```
aws ec2 create-ipam \
    --description "Example description" \
    --operating-regions "RegionName=us-east-2" "RegionName=us-west-1" \
```

```
    --tag-specifications 'ResourceType=ipam,Tags=[{Key=Name,Value=ExampleIPAM}]'
```

(Windows)：

```
aws ec2 create-ipam ^
    --description "Example description" ^
    --operating-regions "RegionName=us-east-2" "RegionName=us-west-1" ^
    --tag-specifications ResourceType=ipam,Tags=[{Key=Name,Value=ExampleIPAM}]
```

輸出：

```
{
    "Ipam": {
        "OwnerId": "123456789012",
        "IpamId": "ipam-036486dfa6af58ee0",
        "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-036486dfa6af58ee0",
        "IpamRegion": "us-east-1",
        "PublicDefaultScopeId": "ipam-scope-071b8042b0195c183",
        "PrivateDefaultScopeId": "ipam-scope-0807405dece705a30",
        "ScopeCount": 2,
        "OperatingRegions": [
            {
                "RegionName": "us-east-2"
            },
            {
                "RegionName": "us-west-1"
            },
            {
                "RegionName": "us-east-1"
            }
        ],
        "State": "create-in-progress",
        "Tags": [
            {
                "Key": "Name",
                "Value": "ExampleIPAM"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的建立 IPAM。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateIpam。

create-key-pair

以下程式碼範例顯示如何使用 create-key-pair。

AWS CLI

建立一組金鑰對

此範例會建立名稱為 MyKeyPair 的金鑰對。

命令：

```
aws ec2 create-key-pair --key-name MyKeyPair
```

輸出是 ASCII 版本的私有金鑰和金鑰指紋。您需要將金鑰儲存到檔案。

如需詳細資訊，請參閱《AWS 命令行介面使用者指南》中的「使用金鑰對」。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateKeyPair。

create-launch-template-version

以下程式碼範例顯示如何使用 create-launch-template-version。

AWS CLI

建立啟動範本版本

此範例會根據啟動範本的第 1 版建立新的啟動範本版本，並指定不同的 AMI ID。

命令：

```
aws ec2 create-launch-template-version --launch-template-id lt-0abcd290751193123
 --version-description WebVersion2 --source-version 1 --launch-template-data
 '{"ImageId":"ami-c998b6b2"}'
```

輸出：

```
{
```

```
  "LaunchTemplateVersion": {
      "VersionDescription": "WebVersion2",
      "LaunchTemplateId": "lt-0abcd290751193123",
      "LaunchTemplateName": "WebServers",
      "VersionNumber": 2,
      "CreatedBy": "arn:aws:iam::123456789012:root",
      "LaunchTemplateData": {
          "ImageId": "ami-c998b6b2",
          "InstanceType": "t2.micro",
          "NetworkInterfaces": [
              {
                  "Ipv6Addresses": [
                      {
                          "Ipv6Address": "2001:db8:1234:1a00::123"
                      }
                  ],
                  "DeviceIndex": 0,
                  "SubnetId": "subnet-7b16de0c",
                  "AssociatePublicIpAddress": true
              }
          ]
      },
      "DefaultVersion": false,
      "CreateTime": "2017-12-01T13:35:46.000Z"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLaunchTemplateVersion。

create-launch-template

以下程式碼範例顯示如何使用 create-launch-template。

AWS CLI

範例 1：建立啟動範本

以下 create-launch-template 範例會建立啟動範本，而此範本可指定執行個體啟動所在的子網路、將公有 IP 地址和 IPv6 地址指派給執行個體，並為執行個體建立標籤。

```
aws ec2 create-launch-template \
    --launch-template-name TemplateForWebServer \
    --version-description WebVersion1 \
```

```
    --launch-template-data '{"NetworkInterfaces":
[{"AssociatePublicIpAddress":true,"DeviceIndex":0,"Ipv6AddressCount":1,"SubnetId":"subnet-7b
[{"ResourceType":"instance","Tags":[{"Key":"purpose","Value":"webserver"}]}]}'
```

輸出：

```
{
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-01238c059e3466abc",
        "LaunchTemplateName": "TemplateForWebServer",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "CreateTime": "2019-01-27T09:13:24.000Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的「從啟動範本啟動執行個體」。如需有關引用 JSON 格式參數的詳細資訊，請參閱《AWS 命令行介面使用者指南》中的「引用字串」。

範例 2：為 Amazon EC2 Auto Scaling 建立啟動範本

以下 `create-launch-template` 範例會建立具備多個標籤和區塊型裝置映射的啟動範本，以指定執行個體啟動時的額外 EBS 磁碟區。請為 `Groups`，即對應至 Auto Scaling 群組在其中啟動執行個體的 VPC 安全群組指定數值。將 VPC 和子網路指定為 Auto Scaling 群組的屬性。

```
aws ec2 create-launch-template \
    --launch-template-name TemplateForAutoScaling \
    --version-description AutoScalingVersion1 \
    --launch-template-data '{"NetworkInterfaces":
[{"DeviceIndex":0,"AssociatePublicIpAddress":true,"Groups":
["sg-7c227019,sg-903004f8"],"DeleteOnTermination":true}],"ImageId":"ami-
b42209de","InstanceType":"m4.large","TagSpecifications":
[{"ResourceType":"instance","Tags":[{"Key":"environment","Value":"production"},
{"Key":"purpose","Value":"webserver"}]},{"ResourceType":"volume","Tags":
[{"Key":"environment","Value":"production"},{"Key":"cost-
center","Value":"cc123"}]}],"BlockDeviceMappings":[{"DeviceName":"/dev/sda1","Ebs":
{"VolumeSize":100}}]}' --region us-east-1
```

輸出：

```
{
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-0123c79c33a54e0abc",
        "LaunchTemplateName": "TemplateForAutoScaling",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "CreateTime": "2019-04-30T18:16:06.000Z"
    }
}
```

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的「建立 Auto Scaling 群組的啟動範本」。如需有關引用 JSON 格式參數的詳細資訊，請參閱《AWS 命令行介面使用者指南》中的「引用字串」。

範例 3：建立指定 EBS 磁碟區加密的啟動範本

下列 `create-launch-template` 範例會建立啟動範本，其中包含從未加密快照中建立的已加密 EBS 磁碟區。此範本也會在建立期間標記磁碟區。如果預設為停用加密，則您必須指定 `"Encrypted"` 選項，如下列範例所示。如果您使用 `"KmsKeyId"` 選項來指定客戶受管的 CMK，即使預設為啟用加密，您也必須指定 `"Encrypted"` 選項。

```
aws ec2 create-launch-template \
  --launch-template-name TemplateForEncryption \
  --launch-template-data file://config.json
```

`config.json` 的內容：

```
{
    "BlockDeviceMappings":[
        {
            "DeviceName":"/dev/sda1",
            "Ebs":{
                "VolumeType":"gp2",
                "DeleteOnTermination":true,
                "SnapshotId":"snap-066877671789bd71b",
                "Encrypted":true,
                "KmsKeyId":"arn:aws:kms:us-east-1:012345678910:key/abcd1234-
a123-456a-a12b-a123b4cd56ef"
            }
        }
```

```
    ],
    "ImageId":"ami-00068cd7555f543d5",
    "InstanceType":"c5.large",
    "TagSpecifications":[
        {
            "ResourceType":"volume",
            "Tags":[
                {
                    "Key":"encrypted",
                    "Value":"yes"
                }
            ]
        }
    ]
}
```

輸出：

```
{
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-0d5bd51bcf8530abc",
        "LaunchTemplateName": "TemplateForEncryption",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "CreateTime": "2020-01-07T19:08:36.000Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的「從快照還原 Amazon EBS 磁碟區」和「預設加密」。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateLaunchTemplate。

create-local-gateway-route-table-virtual-interface-group-association

以下程式碼範例顯示如何使用 create-local-gateway-route-table-virtual-interface-group-association。

AWS CLI

將本機閘道路由表與虛擬介面 (VIFs群組建立關聯

下列create-local-gateway-route-table-virtual-interface-group-association範例會在指定的本機閘道路由表和 VIF 群組之間建立關聯。

```
aws ec2 create-local-gateway-route-table-virtual-interface-group-association \
    --local-gateway-route-table-id lgw-rtb-exampleidabcd1234 \
    --local-gateway-virtual-interface-group-id lgw-vif-grp-exampleid0123abcd
```

輸出：

```
{
    "LocalGatewayRouteTableVirtualInterfaceGroupAssociation": {
        "LocalGatewayRouteTableVirtualInterfaceGroupAssociationId": "lgw-vif-grp-
assoc-exampleid12345678",
        "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-exampleid0123abcd",
        "LocalGatewayId": "lgw-exampleid11223344",
        "LocalGatewayRouteTableId": "lgw-rtb-exampleidabcd1234",
        "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-
gateway-route-table/lgw-rtb-exampleidabcd1234",
        "OwnerId": "111122223333",
        "State": "pending",
        "Tags": []
    }
}
```

如需詳細資訊，請參閱 AWS Outposts 使用者指南中的 VIF 群組關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLocalGatewayRouteTableVirtualInterfaceGroupAssociation。

create-local-gateway-route-table-vpc-association

以下程式碼範例顯示如何使用 create-local-gateway-route-table-vpc-association。

AWS CLI

將 VPC 與路由表建立關聯

下列create-local-gateway-route-table-vpc-association範例會將指定的 VPC 與指定的本機閘道路由表建立關聯。

```
aws ec2 create-local-gateway-route-table-vpc-association \
    --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \
```

```
    --vpc-id vpc-07ef66ac71EXAMPLE
```

輸出：

```
{
    "LocalGatewayRouteTableVpcAssociation": {
        "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-0ee765bcc8EXAMPLE",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
        "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
        "VpcId": "vpc-07ef66ac71EXAMPLE",
        "State": "associated"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLocalGatewayRouteTableVpcAssociation。

create-local-gateway-route-table

以下程式碼範例顯示如何使用 create-local-gateway-route-table。

AWS CLI

建立本機閘道路由表

下列create-local-gateway-route-table範例會使用直接 VPC 路由模式建立本機閘道路由表。

```
aws ec2 create-local-gateway-route-table \
    --local-gateway-id lgw-1a2b3c4d5e6f7g8h9 \
    --mode direct-vpc-routing
```

輸出：

```
{
    "LocalGatewayRouteTable": {
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
        "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-
gateway-route-table/lgw-rtb-abcdefg1234567890",
        "LocalGatewayId": "lgw-1a2b3c4d5e6f7g8h9",
        "OutpostArn": "arn:aws:outposts:us-west-2:111122223333:outpost/
op-021345abcdef67890",
```

```
        "OwnerId": "111122223333",
        "State": "pending",
        "Tags": [],
        "Mode": "direct-vpc-routing"
    }
}
```

如需詳細資訊，請參閱《AWS Outposts 使用者指南》中的本機閘道路油表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLocalGatewayRouteTable。

create-local-gateway-route

以下程式碼範例顯示如何使用 create-local-gateway-route。

AWS CLI

為本機閘道路由表建立靜態路由

下列create-local-gateway-route範例會在指定的本機閘道路由表中建立指定的路由。

```
aws ec2 create-local-gateway-route \
    --destination-cidr-block 0.0.0.0/0 \
    --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE
```

輸出：

```
{
    "Route": {
        "DestinationCidrBlock": "0.0.0.0/0",
        "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
        "Type": "static",
        "State": "deleted",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLocalGatewayRoute。

create-managed-prefix-list

以下程式碼範例顯示如何使用 create-managed-prefix-list。

AWS CLI

建立字首清單

下列create-managed-prefix-list範例會建立最多 10 個項目的 IPv4 字首清單，並在字首清單中建立 2 個項目。

```
aws ec2 create-managed-prefix-list \
    --address-family IPv4 \
    --max-entries 10 \
    --entries Cidr=10.0.0.0/16,Description=vpc-a Cidr=10.2.0.0/16,Description=vpc-b
 \
    --prefix-list-name vpc-cidrs
```

輸出：

```
{
    "PrefixList": {
        "PrefixListId": "pl-0123456abcabcabc1",
        "AddressFamily": "IPv4",
        "State": "create-in-progress",
        "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-0123456abcabcabc1",
        "PrefixListName": "vpc-cidrs",
        "MaxEntries": 10,
        "Version": 1,
        "Tags": [],
        "OwnerId": "123456789012"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的受管字首清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateManagedPrefixList。

create-nat-gateway

以下程式碼範例顯示如何使用 create-nat-gateway。

AWS CLI

範例 1：建立公有 NAT 閘道

下列create-nat-gateway範例會在指定的子網路中建立公有 NAT 閘道，並將彈性 IP 地址與指定的配置 ID 建立關聯。建立公有 NAT 閘道時，您必須關聯彈性 IP 地址。

```
aws ec2 create-nat-gateway \
    --subnet-id subnet-0250c25a1fEXAMPLE \
    --allocation-id eipalloc-09ad461b0dEXAMPLE
```

輸出：

```
{
    "NatGateway": {
        "CreateTime": "2021-12-01T22:22:38.000Z",
        "NatGatewayAddresses": [
            {
                "AllocationId": "eipalloc-09ad461b0dEXAMPLE"
            }
        ],
        "NatGatewayId": "nat-0c61bf8a12EXAMPLE",
        "State": "pending",
        "SubnetId": "subnet-0250c25a1fEXAMPLE",
        "VpcId": "vpc-0a60eb65b4EXAMPLE",
        "ConnectivityType": "public"
    }
}
```

如需詳細資訊，請參閱 《Amazon VPC 使用者指南》中的 NAT 閘道。

範例 2：建立私有 NAT 閘道

下列create-nat-gateway範例會在指定的子網路中建立私有 NAT 閘道。私有 NAT 閘道沒有相關聯的彈性 IP 地址。

```
aws ec2 create-nat-gateway \
    --subnet-id subnet-0250c25a1fEXAMPLE \
    --connectivity-type private
```

輸出：

```
{
    "NatGateway": {
```

```
        "CreateTime": "2021-12-01T22:26:00.000Z",
        "NatGatewayAddresses": [
            {}
        ],
        "NatGatewayId": "nat-011b568379EXAMPLE",
        "State": "pending",
        "SubnetId": "subnet-0250c25a1fEXAMPLE",
        "VpcId": "vpc-0a60eb65b4EXAMPLE",
        "ConnectivityType": "private"
    }
}
```

如需詳細資訊，請參閱 《Amazon VPC 使用者指南》中的 NAT 閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateNatGateway。

`create-network-acl-entry`

以下程式碼範例顯示如何使用 `create-network-acl-entry`。

AWS CLI

建立網路 ACL 項目

此範例會為指定的網路 ACL 建立項目。此規則允許將流量從 UDP 連接埠 53 (DNS) 上的任何 IPv4 地址 (0.0.0.0/0) 傳入任何相關聯的子網路。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 create-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 100 --protocol udp --port-range From=53,To=53 --cidr-block 0.0.0.0/0 --rule-action allow
```

此範例會為指定的網路 ACL 建立規則，允許從 TCP 連接埠 80 (HTTP) 上的任何 IPv6 地址 (：：/0) 傳入流量。

命令：

```
aws ec2 create-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 120 --protocol tcp --port-range From=80,To=80 --ipv6-cidr-block ::/0 --rule-action allow
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateNetworkAclEntry。

create-network-acl

以下程式碼範例顯示如何使用 create-network-acl。

AWS CLI

建立網路 ACL

此範例會為指定的 VPC 建立網路 ACL。

命令：

```
aws ec2 create-network-acl --vpc-id vpc-a01106c2
```

輸出：

```
{
    "NetworkAcl": {
        "Associations": [],
        "NetworkAclId": "acl-5fb85d36",
        "VpcId": "vpc-a01106c2",
        "Tags": [],
        "Entries": [
            {
                "CidrBlock": "0.0.0.0/0",
                "RuleNumber": 32767,
                "Protocol": "-1",
                "Egress": true,
                "RuleAction": "deny"
            },
            {
                "CidrBlock": "0.0.0.0/0",
                "RuleNumber": 32767,
                "Protocol": "-1",
                "Egress": false,
                "RuleAction": "deny"
            }
        ],
        "IsDefault": false
    }
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateNetworkAcl。

create-network-insights-access-scope

以下程式碼範例顯示如何使用 create-network-insights-access-scope。

AWS CLI

建立網路存取範圍

下列create-network-insights-access-scope範例會建立網路存取範圍。

```
aws ec2 create-network-insights-access-scope \
    --cli-input-json file://access-scope-file.json
```

access-scope-file.json 的內容：

```
{
    "MatchPaths": [
        {
            "Source": {
                "ResourceStatement": {
                    "Resources": [
                        "vpc-abcd12e3"
                    ]
                }
            }
         }
    ],
    "ExcludePaths": [
        {
            "Source": {
                "ResourceStatement": {
                    "ResourceTypes": [
                        "AWS::EC2::InternetGateway"
                    ]
                }
            }
        }
    ]
```

```
}
```

輸出：

```
{
    "NetworkInsightsAccessScope": {
        "NetworkInsightsAccessScopeId": "nis-123456789abc01234",
        "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope/nis-123456789abc01234",
        "CreatedDate": "2022-01-25T19:20:28.796000+00:00",
        "UpdatedDate": "2022-01-25T19:20:28.797000+00:00"
    },
    "NetworkInsightsAccessScopeContent": {
        "NetworkInsightsAccessScopeId": "nis-123456789abc01234",
        "MatchPaths": [
            {
                "Source": {
                    "ResourceStatement": {
                        "Resources": [
                            "vpc-abcd12e3"
                        ]
                    }
                }
            }
        ],
        "ExcludePaths": [
            {
                "Source": {
                    "ResourceStatement": {
                        "ResourceTypes": [
                            "AWS::EC2::InternetGateway"
                        ]
                    }
                }
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Network Access Analyzer 指南》中的使用 AWS CLI 開始使用 Network Access Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateNetworkInsightsAccessScope。

create-network-insights-path

以下程式碼範例顯示如何使用 create-network-insights-path。

AWS CLI

建立路徑

下列create-network-insights-path範例會建立路徑。來源是指定的網際網路閘道，而目的地是指定的 EC2 執行個體。若要判斷是否可使用指定的通訊協定和連接埠來連接目的地，請使用start-network-insights-analysis命令來分析路徑。

```
aws ec2 create-network-insights-path \
    --source igw-0797cccdc9d73b0e5 \
    --destination i-0495d385ad28331c7 \
    --destination-port 22 \
    --protocol TCP
```

輸出：

```
{
    "NetworkInsightsPaths": {
        "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",
        "NetworkInsightsPathArn": "arn:aws:ec2:us-east-1:123456789012:network-
insights-path/nip-0b26f224f1d131fa8",
        "CreatedDate": "2021-01-20T22:43:46.933Z",
        "Source": "igw-0797cccdc9d73b0e5",
        "Destination": "i-0495d385ad28331c7",
        "Protocol": "tcp"
    }
}
```

如需詳細資訊，請參閱 Reachability Analyzer 指南中的開始使用 AWS CLI。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateNetworkInsightsPath。

create-network-interface-permission

以下程式碼範例顯示如何使用 create-network-interface-permission。

AWS CLI

建立網路介面許可

此範例授予 帳戶123456789012將網路介面連接至執行個體eni-1a2b3c4d的許可。

命令：

```
aws ec2 create-network-interface-permission --network-interface-id eni-1a2b3c4d --
aws-account-id 123456789012 --permission INSTANCE-ATTACH
```

輸出：

```
{
  "InterfacePermission": {
      "PermissionState": {
          "State": "GRANTED"
      },
      "NetworkInterfacePermissionId": "eni-perm-06fd19020ede149ea",
      "NetworkInterfaceId": "eni-1a2b3c4d",
      "Permission": "INSTANCE-ATTACH",
      "AwsAccountId": "123456789012"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateNetworkInterfacePermission。

create-network-interface

以下程式碼範例顯示如何使用 create-network-interface。

AWS CLI

範例 1：指定網路介面的 IPv4 地址

下列create-network-interface範例會為具有指定主要 IPv4 地址的指定子網路建立網路介面。

```
aws ec2 create-network-interface \
    --subnet-id subnet-00a24d0d67acf6333 \
    --description "my network interface" \
    --groups sg-09dfba7ed20cda78b \
    --private-ip-address 10.0.8.17
```

輸出：

```
{
    "NetworkInterface": {
        "AvailabilityZone": "us-west-2a",
        "Description": "my network interface",
        "Groups": [
            {
                "GroupName": "my-security-group",
                "GroupId": "sg-09dfba7ed20cda78b"
            }
        ],
        "InterfaceType": "interface",
        "Ipv6Addresses": [],
        "MacAddress": "06:6a:0f:9a:49:37",
        "NetworkInterfaceId": "eni-0492b355f0cf3b3f8",
        "OwnerId": "123456789012",
        "PrivateDnsName": "ip-10-0-8-18.us-west-2.compute.internal",
        "PrivateIpAddress": "10.0.8.17",
        "PrivateIpAddresses": [
            {
                "Primary": true,
                "PrivateDnsName": "ip-10-0-8-17.us-west-2.compute.internal",
                "PrivateIpAddress": "10.0.8.17"
            }
        ],
        "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
        "RequesterManaged": false,
        "SourceDestCheck": true,
        "Status": "pending",
        "SubnetId": "subnet-00a24d0d67acf6333",
        "TagSet": [],
        "VpcId": "vpc-02723a0feeeb9d57b"
    }
}
```

範例 2：使用 IPv4 地址和 IPv6 地址建立網路介面

下列`create-network-interface`範例會使用 Amazon EC2 選取的 IPv4 地址和 IPv6 地址，為指定的子網路建立網路介面。

```
aws ec2 create-network-interface \
    --subnet-id subnet-00a24d0d67acf6333 \
    --description "my dual stack network interface" \
    --ipv6-address-count 1 \
```

```
    --groups sg-09dfba7ed20cda78b
```

輸出：

```
{
    "NetworkInterface": {
        "AvailabilityZone": "us-west-2a",
        "Description": "my dual stack network interface",
        "Groups": [
            {
                "GroupName": "my-security-group",
                "GroupId": "sg-09dfba7ed20cda78b"
            }
        ],
        "InterfaceType": "interface",
        "Ipv6Addresses": [
            {
                "Ipv6Address": "2600:1f13:cfe:3650:a1dc:237c:393a:4ba7",
                "IsPrimaryIpv6": false
            }
        ],
        "MacAddress": "06:b8:68:d2:b2:2d",
        "NetworkInterfaceId": "eni-05da417453f9a84bf",
        "OwnerId": "123456789012",
        "PrivateDnsName": "ip-10-0-8-18.us-west-2.compute.internal",
        "PrivateIpAddress": "10.0.8.18",
        "PrivateIpAddresses": [
            {
                "Primary": true,
                "PrivateDnsName": "ip-10-0-8-18.us-west-2.compute.internal",
                "PrivateIpAddress": "10.0.8.18"
            }
        ],
        "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
        "RequesterManaged": false,
        "SourceDestCheck": true,
        "Status": "pending",
        "SubnetId": "subnet-00a24d0d67acf6333",
        "TagSet": [],
        "VpcId": "vpc-02723a0feeeb9d57b",
        "Ipv6Address": "2600:1f13:cfe:3650:a1dc:237c:393a:4ba7"
    }
}
```

範例 3：使用連線追蹤組態選項建立網路介面

下列create-network-interface範例會建立網路介面，並設定閒置連線追蹤逾時。

```
aws ec2 create-network-interface \
    --subnet-id subnet-00a24d0d67acf6333 \
    --groups sg-02e57dbcfe0331c1b \
    --connection-tracking-specification TcpEstablishedTimeout=86400,UdpTimeout=60
```

輸出：

```
{
    "NetworkInterface": {
        "AvailabilityZone": "us-west-2a",
        "ConnectionTrackingConfiguration": {
            "TcpEstablishedTimeout": 86400,
            "UdpTimeout": 60
        },
        "Description": "",
        "Groups": [
            {
                "GroupName": "my-security-group",
                "GroupId": "sg-02e57dbcfe0331c1b"
            }
        ],
        "InterfaceType": "interface",
        "Ipv6Addresses": [],
        "MacAddress": "06:4c:53:de:6d:91",
        "NetworkInterfaceId": "eni-0c133586e08903d0b",
        "OwnerId": "123456789012",
        "PrivateDnsName": "ip-10-0-8-94.us-west-2.compute.internal",
        "PrivateIpAddress": "10.0.8.94",
        "PrivateIpAddresses": [
            {
                "Primary": true,
                "PrivateDnsName": "ip-10-0-8-94.us-west-2.compute.internal",
                "PrivateIpAddress": "10.0.8.94"
            }
        ],
        "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
        "RequesterManaged": false,
        "SourceDestCheck": true,
        "Status": "pending",
```

```
        "SubnetId": "subnet-00a24d0d67acf6333",
        "TagSet": [],
        "VpcId": "vpc-02723a0feeeb9d57b"
    }
}
```

範例 4：建立 Elastic Fabric Adapter

下列create-network-interface範例會建立 EFA。

```
aws ec2 create-network-interface \
    --interface-type efa \
    --subnet-id subnet-00a24d0d67acf6333 \
    --description "my efa" \
    --groups sg-02e57dbcfe0331c1b
```

輸出：

```
{
    "NetworkInterface": {
        "AvailabilityZone": "us-west-2a",
        "Description": "my efa",
        "Groups": [
            {
                "GroupName": "my-efa-sg",
                "GroupId": "sg-02e57dbcfe0331c1b"
            }
        ],
        "InterfaceType": "efa",
        "Ipv6Addresses": [],
        "MacAddress": "06:d7:a4:f7:4d:57",
        "NetworkInterfaceId": "eni-034acc2885e862b65",
        "OwnerId": "123456789012",
        "PrivateDnsName": "ip-10-0-8-180.us-west-2.compute.internal",
        "PrivateIpAddress": "10.0.8.180",
        "PrivateIpAddresses": [
            {
                "Primary": true,
                "PrivateDnsName": "ip-10-0-8-180.us-west-2.compute.internal",
                "PrivateIpAddress": "10.0.8.180"
            }
        ],
        "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
```

```
        "RequesterManaged": false,
        "SourceDestCheck": true,
        "Status": "pending",
        "SubnetId": "subnet-00a24d0d67acf6333",
        "TagSet": [],
        "VpcId": "vpc-02723a0feeeb9d57b"
    }
}
```

如需詳細，請參閱《Amazon EC2 使用者指南》中的彈性網絡介面。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateNetworkInterface。

create-placement-group

以下程式碼範例顯示如何使用 create-placement-group。

AWS CLI

建立置放群組

此範例命令會建立具有指定名稱的置放群組。

命令：

```
aws ec2 create-placement-group --group-name my-cluster --strategy cluster
```

建立分割區置放群組

此範例命令會建立名為 的分割區置放群組，HDFS-Group-A其中包含五個分割區。

命令：

```
aws ec2 create-placement-group --group-name HDFS-Group-A --strategy partition --
partition-count 5
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePlacementGroup。

create-replace-root-volume-task

以下程式碼範例顯示如何使用 create-replace-root-volume-task。

AWS CLI

範例 1：將根磁碟區還原為其初始啟動狀態

下列create-replace-root-volume-task範例會將執行個體 i-0123456789abcdefa 的根磁碟區還原為其初始啟動狀態。

```
aws ec2 create-replace-root-volume-task \
    --instance-id i-0123456789abcdefa
```

輸出：

```
{
    "ReplaceRootVolumeTask":
    {
        "InstanceId": "i-0123456789abcdefa",
            "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",
            "TaskState": "pending",
            "StartTime": "2022-03-14T15:06:38Z",
            "Tags": []
    }
}
```

範例 2：將根磁碟區還原至特定快照

下列create-replace-root-volume-task範例會將執行個體 i-0123456789abcdefa 的根磁碟區還原至快照 snap-0abcdef1234567890。

```
aws ec2 create-replace-root-volume-task \
    --instance-id i-0123456789abcdefa \
    --snapshot-id  snap-0abcdef1234567890
```

輸出：

```
{
    "ReplaceRootVolumeTask":
    {
        "InstanceId": "i-0123456789abcdefa",
        "ReplaceRootVolumeTaskId": "replacevol-0555566667777abcd",
        "TaskState": "pending",
        "StartTime": "2022-03-14T15:16:28Z",
```

```
        "Tags": []
    }
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的取代根磁碟區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateReplaceRootVolumeTask。

create-reserved-instances-listing

以下程式碼範例顯示如何使用 create-reserved-instances-listing。

AWS CLI

在預留執行個體 Marketplace 中列出預留執行個體

下列create-reserved-instances-listing範例會在預留執行個體 Marketplace 中為指定的預留執行個體建立清單。

```
aws ec2 create-reserved-instances-listing \
    --reserved-instances-id 5ec28771-05ff-4b9b-aa31-9e57dexample \
    --instance-count 3 \
    --price-schedules CurrencyCode=USD,Price=25.50 \
    --client-token 550e8400-e29b-41d4-a716-446655440000
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateReservedInstancesListing。

create-restore-image-task

以下程式碼範例顯示如何使用 create-restore-image-task。

AWS CLI

從 S3 儲存貯體還原 AMI

下列create-restore-image-task範例會從 S3 儲存貯體還原 AMI。使用describe-store-image-tasks輸出S3ObjectKey `` and ``Bucket中的 值，指定 AMI 的物件索引鍵和 AMI 複製的 S3 儲存貯體名稱，並指定還原 AMI 的名稱。此帳戶的區域中，AMI 的名稱必須是唯一的。還原的 AMI 會收到新的 AMI ID。

```
aws ec2 create-restore-image-task \
```

```
    --object-key ami-1234567890abcdef0.bin \
    --bucket my-ami-bucket \
    --name 'New AMI Name'
```

輸出：

```
{
    "ImageId": "ami-0eab20fe36f83e1a8"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的使用 S3 存放和還原 AMI。 Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRestoreImageTask。

create-route-table

以下程式碼範例顯示如何使用 create-route-table。

AWS CLI

建立路由表

此範例會建立指定 VPC 的路由表。

命令：

```
aws ec2 create-route-table --vpc-id vpc-a01106c2
```

輸出：

```
{
    "RouteTable": {
        "Associations": [],
        "RouteTableId": "rtb-22574640",
        "VpcId": "vpc-a01106c2",
        "PropagatingVgws": [],
        "Tags": [],
        "Routes": [
            {
                "GatewayId": "local",
                "DestinationCidrBlock": "10.0.0.0/16",
                "State": "active"
```

```
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateRouteTable。

`create-route`

以下程式碼範例顯示如何使用 `create-route`。

AWS CLI

建立路由

此範例會為指定的路由表建立路由。路由符合所有 IPv4 流量 (`0.0.0.0/0`)，並將其路由至指定的網際網路閘道。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 create-route --route-table-id rtb-22574640 --destination-cidr-
block 0.0.0.0/0 --gateway-id igw-c0a643a9
```

此範例命令會在路由表 rtb-g8ff4ea2 中建立路由。路由符合 IPv4 CIDR 區塊 10.0.0.0/16 的流量，並將其路由至 VPC 互連連線 pcx-111aa22。此路由可讓流量導向至 VPC 對等互連連線中的對等 VPC。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 create-route --route-table-id rtb-g8ff4ea2 --destination-cidr-
block 10.0.0.0/16 --vpc-peering-connection-id pcx-1a2b3c4d
```

此範例會在符合所有 IPv6 流量 (`::/0`) 的指定路由表中建立路由，並將其路由至指定的輸出限定網際網路閘道。

命令：

```
aws ec2 create-route --route-table-id rtb-dce620b8 --destination-ipv6-cidr-
block ::/0 --egress-only-internet-gateway-id eigw-01eadbd45ecd7943f
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRoute。

create-security-group

以下程式碼範例顯示如何使用 create-security-group。

AWS CLI

為 EC2-Classic 建立安全群組

此範例會建立名為 MySecurityGroup 的安全群組。

命令：

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My
 security group"
```

輸出：

```
{
    "GroupId": "sg-903004f8"
}
```

為 EC2-VPC 建立安全群組

此範例會為指定 VPC 建立名為 MySecurityGroup 的安全群組。

命令：

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My
 security group" --vpc-id vpc-1a2b3c4d
```

輸出：

```
{
    "GroupId": "sg-903004f8"
}
```

如需詳細資訊，請參閱《AWS 命令行介面使用者指南》中的「使用安全群組」。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateSecurityGroup。

create-snapshot

以下程式碼範例顯示如何使用 create-snapshot。

AWS CLI

建立快照

此範例命令會建立磁碟區 ID 為 的磁碟區快照，vol-1234567890abcdef0以及識別快照的簡短描述。

命令：

```
aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description "This is my root volume snapshot"
```

輸出：

```
{
    "Description": "This is my root volume snapshot",
    "Tags": [],
    "Encrypted": false,
    "VolumeId": "vol-1234567890abcdef0",
    "State": "pending",
    "VolumeSize": 8,
    "StartTime": "2018-02-28T21:06:01.000Z",
    "Progress": "",
    "OwnerId": "012345678910",
    "SnapshotId": "snap-066877671789bd71b"
}
```

使用標籤建立快照

此範例命令會建立快照並套用兩個標籤： purpose=prod 和 costcenter=123。

命令：

```
aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description 'Prod backup' --tag-specifications 'ResourceType=snapshot,Tags=[{Key=purpose,Value=prod},{Key=costcenter,Value=123}]'
```

輸出：

```
{
    "Description": "Prod backup",
    "Tags": [
```

```
        {
            "Value": "prod",
            "Key": "purpose"
        },
        {
            "Value": "123",
            "Key": "costcenter"
        }
    ],
    "Encrypted": false,
    "VolumeId": "vol-1234567890abcdef0",
    "State": "pending",
    "VolumeSize": 8,
    "StartTime": "2018-02-28T21:06:06.000Z",
    "Progress": "",
    "OwnerId": "012345678910",
    "SnapshotId": "snap-09ed24a70bc19bbe4"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSnapshot。

create-snapshots

以下程式碼範例顯示如何使用 create-snapshots。

AWS CLI

範例 1：建立多磁碟區快照

下列create-snapshots範例會建立連接至指定執行個體之所有磁碟區的快照。

```
aws ec2 create-snapshots \
    --instance-specification InstanceId=i-1234567890abcdef0 \
    --description "This is snapshot of a volume from my-instance"
```

輸出：

```
{
    "Snapshots": [
        {
            "Description": "This is a snapshot of a volume from my-instance",
            "Tags": [],
            "Encrypted": false,
```

```
            "VolumeId": "vol-0a01d2d5a34697479",
            "State": "pending",
            "VolumeSize": 16,
            "StartTime": "2019-08-05T16:58:19.000Z",
            "Progress": "",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-07f30e3909aa0045e"
        },
        {
            "Description": "This is a snapshot of a volume from my-instance",
            "Tags": [],
            "Encrypted": false,
            "VolumeId": "vol-02d0d4947008cb1a2",
            "State": "pending",
            "VolumeSize": 20,
            "StartTime": "2019-08-05T16:58:19.000Z",
            "Progress": "",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-0ec20b602264aad48"
        },
        ...
    ]
}
```

範例 2：從來源磁碟區建立具有標籤的多磁碟區快照

下列create-snapshots範例會建立連接至指定執行個體之所有磁碟區的快照，並將標籤從每個磁碟區複製到其對應的快照。

```
aws ec2 create-snapshots \
    --instance-specification InstanceId=i-1234567890abcdef0 \
    --copy-tags-from-source volume \
    --description "This is snapshot of a volume from my-instance"
```

輸出：

```
{
    "Snapshots": [
        {
            "Description": "This is a snapshot of a volume from my-instance",
            "Tags": [
                {
                    "Key": "Name",
```

```
                    "Value": "my-volume"
                }
            ],
            "Encrypted": false,
            "VolumeId": "vol-02d0d4947008cb1a2",
            "State": "pending",
            "VolumeSize": 20,
            "StartTime": "2019-08-05T16:53:04.000Z",
            "Progress": "",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-053bfaeb821a458dd"
        }
        ...
    ]
}
```

範例 3：建立不包含根磁碟區的多磁碟區快照

下列create-snapshots範例會建立連接至指定執行個體之所有磁碟區的快照，根磁碟區除外。

```
aws ec2 create-snapshots \
    --instance-specification InstanceId=i-1234567890abcdef0,ExcludeBootVolume=true
```

如需範例輸出，請參閱範例 1。

範例 4：建立多磁碟區快照並新增標籤

下列create-snapshots範例會建立連接至指定執行個體之所有磁碟區的快照，並為每個快照新增兩個標籤。

```
aws ec2 create-snapshots \
    --instance-specification InstanceId=i-1234567890abcdef0 \
    --tag-specifications 'ResourceType=snapshot,Tags=[{Key=Name,Value=backup},
{Key=costcenter,Value=123}]'
```

如需範例輸出，請參閱範例 1。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSnapshots。

create-spot-datafeed-subscription

以下程式碼範例顯示如何使用 create-spot-datafeed-subscription。

AWS CLI

建立 Spot 執行個體資料饋送

下列create-spot-datafeed-subscription範例會建立 Spot 執行個體資料饋送。

```
aws ec2 create-spot-datafeed-subscription \
    --bucket amzn-s3-demo-bucket \
    --prefix spot-data-feed
```

輸出：

```
{
    "SpotDatafeedSubscription": {
        "Bucket": "amzn-s3-demo-bucket",
        "OwnerId": "123456789012",
        "Prefix": "spot-data-feed",
        "State": "Active"
    }
}
```

資料饋送會存放在您指定的 Amazon S3 儲存貯體中。此資料饋送的檔案名稱格式如下。

```
amzn-s3-demo-bucket.s3.amazonaws.com/spot-data-feed/123456789012.YYYY-MM-DD-
HH.n.abcd1234.gz
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Spot 執行個體資料饋送。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSpotDatafeedSubscription。

create-store-image-task

以下程式碼範例顯示如何使用 create-store-image-task。

AWS CLI

將 AMI 存放在 S3 儲存貯體

下列create-store-image-task範例會將 AMI 存放在 S3 儲存貯體中。指定 AMI 的 ID 以及要存放 AMI 的 S3 儲存貯體名稱。

```
aws ec2 create-store-image-task \
```

```
  --image-id ami-1234567890abcdef0 \
  --bucket my-ami-bucket
```

輸出：

```
{
    "ObjectKey": "ami-1234567890abcdef0.bin"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的使用 S3 存放和還原 AMI。 Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateStoreImageTask。

create-subnet-cidr-reservation

以下程式碼範例顯示如何使用 create-subnet-cidr-reservation。

AWS CLI

建立子網路 CIDR 保留

下列create-subnet-cidr-reservation範例會為指定的子網路和 CIDR 範圍建立子網路 CIDR 保留。

```
aws ec2 create-subnet-cidr-reservation \
    --subnet-id subnet-03c51e2eEXAMPLE \
    --reservation-type prefix \
    --cidr 10.1.0.20/26
```

輸出：

```
{
    "SubnetCidrReservation": {
        "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
        "SubnetId": "subnet-03c51e2e6cEXAMPLE",
        "Cidr": "10.1.0.16/28",
        "ReservationType": "prefix",
        "OwnerId": "123456789012"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 子網 CIDR 保留。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSubnetCidrReservation。

create-subnet

以下程式碼範例顯示如何使用 create-subnet。

AWS CLI

範例 1：建立僅具有 IPv4 CIDR 區塊的子網路

以下 create-subnet 範例會在指定 VPC 內建立具有指定 IPv4 CIDR 區塊的子網路。

```
aws ec2 create-subnet \
    --vpc-id vpc-081ec835f3EXAMPLE \
    --cidr-block 10.0.0.0/24 \
    --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv4-only-
subnet}]
```

輸出：

```
{
    "Subnet": {
        "AvailabilityZone": "us-west-2a",
        "AvailabilityZoneId": "usw2-az2",
        "AvailableIpAddressCount": 251,
        "CidrBlock": "10.0.0.0/24",
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false,
        "State": "available",
        "SubnetId": "subnet-0e99b93155EXAMPLE",
        "VpcId": "vpc-081ec835f3EXAMPLE",
        "OwnerId": "123456789012",
        "AssignIpv6AddressOnCreation": false,
        "Ipv6CidrBlockAssociationSet": [],
        "Tags": [
            {
                "Key": "Name",
                "Value": "my-ipv4-only-subnet"
            }
        ],
        "SubnetArn": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0e99b93155EXAMPLE"
```

```
    }
}
```

範例 2：建立具有 IPv4 和 IPv6 CIDR 區塊的子網路

以下 `create-subnet` 範例會在指定 VPC 內建立具有指定 IPv4 和 IPv6 CIDR 區塊的子網路。

```
aws ec2 create-subnet \
    --vpc-id vpc-081ec835f3EXAMPLE \
    --cidr-block 10.0.0.0/24 \
    --ipv6-cidr-block 2600:1f16:cfe:3660::/64 \
    --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv4-ipv6-
subnet}]
```

輸出：

```
{
    "Subnet": {
        "AvailabilityZone": "us-west-2a",
        "AvailabilityZoneId": "usw2-az2",
        "AvailableIpAddressCount": 251,
        "CidrBlock": "10.0.0.0/24",
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false,
        "State": "available",
        "SubnetId": "subnet-0736441d38EXAMPLE",
        "VpcId": "vpc-081ec835f3EXAMPLE",
        "OwnerId": "123456789012",
        "AssignIpv6AddressOnCreation": false,
        "Ipv6CidrBlockAssociationSet": [
            {
                "AssociationId": "subnet-cidr-assoc-06c5f904499fcc623",
                "Ipv6CidrBlock": "2600:1f13:cfe:3660::/64",
                "Ipv6CidrBlockState": {
                    "State": "associating"
                }
            }
        ],
        "Tags": [
            {
                "Key": "Name",
                "Value": "my-ipv4-ipv6-subnet"
            }
```

```
        ],
        "SubnetArn": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0736441d38EXAMPLE"
    }
}
```

範例 3：建立僅具有 IPv6 CIDR 區塊的子網路

以下 `create-subnet` 範例會在指定 VPC 內建立具有指定 IPv6 CIDR 區塊的子網路。

```
aws ec2 create-subnet \
    --vpc-id vpc-081ec835f3EXAMPLE \
    --ipv6-native \
    --ipv6-cidr-block 2600:1f16:115:200::/64 \
    --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv6-only-
subnet}]
```

輸出：

```
{
    "Subnet": {
        "AvailabilityZone": "us-west-2a",
        "AvailabilityZoneId": "usw2-az2",
        "AvailableIpAddressCount": 0,
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false,
        "State": "available",
        "SubnetId": "subnet-03f720e7deEXAMPLE",
        "VpcId": "vpc-081ec835f3EXAMPLE",
        "OwnerId": "123456789012",
        "AssignIpv6AddressOnCreation": true,
        "Ipv6CidrBlockAssociationSet": [
            {
                "AssociationId": "subnet-cidr-assoc-01ef639edde556709",
                "Ipv6CidrBlock": "2600:1f13:cfe:3660::/64",
                "Ipv6CidrBlockState": {
                    "State": "associating"
                }
            }
        ],
        "Tags": [
            {
                "Key": "Name",
```

```
                "Value": "my-ipv6-only-subnet"
            }
        ],
        "SubnetArn": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-03f720e7deEXAMPLE"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 VPC 和子網路。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateSubnet。

create-tags

以下程式碼範例顯示如何使用 create-tags。

AWS CLI

範例 1：將標籤新增至資源

以下 create-tags 範例會將標籤 Stack=production 新增至指定的映像，或覆寫 AMI 的現有標籤，其中標籤金鑰為 Stack。

```
aws ec2 create-tags \
    --resources ami-1234567890abcdef0 \
    --tags Key=Stack,Value=production
```

此命令不會產生輸出

範例 2：將標籤新增至多個資源

下列 create-tags 範例會為 AMI 和執行個體新增 (或覆寫) 兩個標籤。其中一個標籤具有索引鍵 (webserver)，但沒有值 (值會設定為空白字串)。另一個標籤則有一個索引鍵 (stack) 和一個值 (Production)。

```
aws ec2 create-tags \
    --resources ami-1a2b3c4d i-1234567890abcdef0 \
    --tags Key=webserver,Value=   Key=stack,Value=Production
```

此命令不會產生輸出

範例 3：新增包含特殊字元的標籤

下列create-tags範例會新增執行個體[Group]=test的 標籤。中括號 ([和]) 是特殊字元，必須將其逸出。下列範例也會使用每個環境適用的行接續字元。

如果您使用的是 Windows，請包圍具有雙引號 (") 之特殊字元的元素，然後在每個雙引號字元前面加上反斜線 (\)，如下所示。

```
aws ec2 create-tags ^
    --resources i-1234567890abcdef0 ^
    --tags Key=\"[Group]\",Value=test
```

如果您使用的是 Windows PowerShell，請包圍具有雙引號 (") 之特殊字元的 元素，在每個雙引號字元前面加上反斜線 (\)，然後使用單引號 (') 包圍整個索引鍵和值結構，如下所示。

```
aws ec2 create-tags `
    --resources i-1234567890abcdef0 `
    --tags 'Key=\"[Group]\",Value=test'
```

如果您使用的是 Linux 或 OS X，請包圍具有雙引號 (") 之特殊字元的 元素，然後使用單引號 (') 包圍整個索引鍵和值結構，如下所示。

```
aws ec2 create-tags \
    --resources i-1234567890abcdef0 \
    --tags 'Key="[Group]",Value=test'
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的標記您的 Amazon EC2 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateTags。

create-traffic-mirror-filter-rule

以下程式碼範例顯示如何使用 create-traffic-mirror-filter-rule。

AWS CLI

為傳入 TCP 流量建立篩選規則

下列create-traffic-mirror-filter-rule範例會建立可用來鏡像所有傳入 TCP 流量的規則。執行此命令之前，請使用 create-traffic-mirror-filter建立流量鏡像篩選條件。

```
aws ec2 create-traffic-mirror-filter-rule \
    --description 'TCP Rule' \
```

```
    --destination-cidr-block 0.0.0.0/0  \
    --protocol 6 \
    --rule-action accept \
    --rule-number 1 \
    --source-cidr-block 0.0.0.0/0 \
    --traffic-direction ingress \
    --traffic-mirror-filter-id tmf-04812ff784b25ae67
```

輸出：

```
{
    "TrafficMirrorFilterRule": {
        "DestinationCidrBlock": "0.0.0.0/0",
        "TrafficMirrorFilterId": "tmf-04812ff784b25ae67",
        "TrafficMirrorFilterRuleId": "tmfr-02d20d996673f3732",
        "SourceCidrBlock": "0.0.0.0/0",
        "TrafficDirection": "ingress",
        "Description": "TCP Rule",
        "RuleNumber": 1,
        "RuleAction": "accept",
        "Protocol": 6
    },
    "ClientToken": "4752b573-40a6-4eac-a8a4-a72058761219"
}
```

如需詳細資訊，請參閱《流量鏡像指南》中的建立流量鏡像篩選條件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTrafficMirrorFilterRule。

create-traffic-mirror-filter

以下程式碼範例顯示如何使用 create-traffic-mirror-filter。

AWS CLI

建立流量鏡像篩選條件

下列create-traffic-mirror-filter範例會建立流量鏡像篩選條件。建立篩選條件之後，請使用 create-traffic-mirror-filter-rule 來新增規則。

```
aws ec2 create-traffic-mirror-filter \
    --description 'TCP Filter'
```

輸出：

```
{
    "ClientToken": "28908518-100b-4987-8233-8c744EXAMPLE",
    "TrafficMirrorFilter": {
        "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
        "Description": "TCP Filter",
        "EgressFilterRules": [],
        "IngressFilterRules": [],
        "Tags": [],
        "NetworkServices": []
    }
}
```

如需詳細資訊，請參閱《流量鏡像指南》中的建立流量鏡像篩選條件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTrafficMirrorFilter。

create-traffic-mirror-session

以下程式碼範例顯示如何使用 create-traffic-mirror-session。

AWS CLI

建立流量鏡像工作階段

下列create-traffic-mirror-session命令會為 25 個位元組封包的指定來源和目標建立流量鏡像工作階段。

```
aws ec2 create-traffic-mirror-session \
    --description 'example session' \
    --traffic-mirror-target-id tmt-07f75d8feeEXAMPLE \
    --network-interface-id eni-070203f901EXAMPLE \
    --session-number 1  \
    --packet-length 25 \
    --traffic-mirror-filter-id tmf-04812ff784EXAMPLE
```

輸出：

```
{
    "TrafficMirrorSession": {
        "TrafficMirrorSessionId": "tms-08a33b1214EXAMPLE",
```

```
        "TrafficMirrorTargetId": "tmt-07f75d8feeEXAMPLE",
        "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
        "NetworkInterfaceId": "eni-070203f901EXAMPLE",
        "OwnerId": "111122223333",
        "PacketLength": 25,
        "SessionNumber": 1,
        "VirtualNetworkId": 7159709,
        "Description": "example session",
        "Tags": []
    },
    "ClientToken": "5236cffc-ee13-4a32-bb5b-388d9da09d96"
}
```

如需詳細資訊，請參閱《流量鏡射指南》中的建立流量鏡射工作階段。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTrafficMirrorSession。

create-traffic-mirror-target

以下程式碼範例顯示如何使用 create-traffic-mirror-target。

AWS CLI

建立 Network Load Balancer 流量鏡像目標

下列create-traffic-mirror-target範例會建立 Network Load Balancer 流量鏡像目標。

```
aws ec2 create-traffic-mirror-target \
    --description 'Example Network Load Balancer Target' \
    --network-load-balancer-arn arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/net/NLB/7cdec873EXAMPLE
```

輸出：

```
{
    "TrafficMirrorTarget": {
        "Type": "network-load-balancer",
        "Tags": [],
        "Description": "Example Network Load Balancer Target",
        "OwnerId": "111122223333",
        "NetworkLoadBalancerArn": "arn:aws:elasticloadbalancing:us-
east-1:724145273726:loadbalancer/net/NLB/7cdec873EXAMPLE",
```

```
        "TrafficMirrorTargetId": "tmt-0dabe9b0a6EXAMPLE"
    },
    "ClientToken": "d5c090f5-8a0f-49c7-8281-72c796a21f72"
}
```

建立網路流量鏡像目標

下列create-traffic-mirror-target範例會建立網路界面流量鏡射目標。

```
aws ec2 create-traffic-mirror-target \
    --description 'Network interface target' \
    --network-interface-id eni-eni-01f6f631eEXAMPLE
```

輸出：

```
{
    "ClientToken": "5289a345-0358-4e62-93d5-47ef3061d65e",
    "TrafficMirrorTarget": {
        "Description": "Network interface target",
        "NetworkInterfaceId": "eni-01f6f631eEXAMPLE",
        "TrafficMirrorTargetId": "tmt-02dcdbe2abEXAMPLE",
        "OwnerId": "111122223333",
        "Type": "network-interface",
        "Tags": []
    }
}
```

如需詳細資訊，請參閱《流量鏡射指南》中的建立流量鏡射目標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTrafficMirrorTarget。

create-transit-gateway-connect-peer

以下程式碼範例顯示如何使用 create-transit-gateway-connect-peer。

AWS CLI

建立 Transit Gateway Connect 對等

下列create-transit-gateway-connect-peer範例會建立 Connect 對等。

```
aws ec2 create-transit-gateway-connect-peer \
```

```
    --transit-gateway-attachment-id tgw-attach-0f0927767cEXAMPLE \
    --peer-address 172.31.1.11 \
    --inside-cidr-blocks 169.254.6.0/29
```

輸出：

```
{
    "TransitGatewayConnectPeer": {
        "TransitGatewayAttachmentId": "tgw-attach-0f0927767cEXAMPLE",
        "TransitGatewayConnectPeerId": "tgw-connect-peer-0666adbac4EXAMPLE",
        "State": "pending",
        "CreationTime": "2021-10-13T03:35:17.000Z",
        "ConnectPeerConfiguration": {
            "TransitGatewayAddress": "10.0.0.234",
            "PeerAddress": "172.31.1.11",
            "InsideCidrBlocks": [
                "169.254.6.0/29"
            ],
            "Protocol": "gre",
            "BgpConfigurations": [
                {
                    "TransitGatewayAsn": 64512,
                    "PeerAsn": 64512,
                    "TransitGatewayAddress": "169.254.6.2",
                    "PeerAddress": "169.254.6.1",
                    "BgpStatus": "down"
                },
                {
                    "TransitGatewayAsn": 64512,
                    "PeerAsn": 64512,
                    "TransitGatewayAddress": "169.254.6.3",
                    "PeerAddress": "169.254.6.1",
                    "BgpStatus": "down"
                }
            ]
        }
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的 Transit Gateway Connect 連接和 Transit Gateway Connect 對等。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTransitGatewayConnectPeer。

create-transit-gateway-connect

以下程式碼範例顯示如何使用 create-transit-gateway-connect。

AWS CLI

建立傳輸閘道 Connect 連接

下列create-transit-gateway-connect範例會為指定的附件建立具有「gre」通訊協定的 Connect 連接。

```
aws ec2 create-transit-gateway-connect \
    --transport-transit-gateway-attachment-id tgw-attach-0a89069f57EXAMPLE \
    --options "Protocol=gre"
```

輸出：

```
{
    "TransitGatewayConnect": {
        "TransitGatewayAttachmentId": "tgw-attach-037012e5dcEXAMPLE",
        "TransportTransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
        "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
        "State": "pending",
        "CreationTime": "2021-03-09T19:59:17+00:00",
        "Options": {
            "Protocol": "gre"
        }
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的 Transit Gateway Connect 連接和 Transit Gateway Connect 對等。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTransitGatewayConnect。

create-transit-gateway-multicast-domain

以下程式碼範例顯示如何使用 create-transit-gateway-multicast-domain。

AWS CLI

範例 1：建立 IGMP 多點傳送網域

下列create-transit-gateway-multicast-domain範例會為指定的傳輸閘道建立多點傳送網域。在停用靜態來源的情況下，與多點傳送網域相關聯的子網路中的任何執行個體都可以傳送多點傳送流量。如果至少有一個成員使用 IGMP 通訊協定，您必須啟用 IGMPv2 支援。

```
aws ec2 create-transit-gateway-multicast-domain \
    --transit-gateway-id tgw-0bf0bffefaEXAMPLE \
    --options StaticSourcesSupport=disable,Igmpv2Support=enable
```

輸出：

```
{
    "TransitGatewayMulticastDomain": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c9e29e2a7EXAMPLE",
        "TransitGatewayId": "tgw-0bf0bffefaEXAMPLE",
        "TransitGatewayMulticastDomainArn": "arn:aws:ec2:us-
west-2:123456789012:transit-gateway-multicast-domain/tgw-mcast-
domain-0c9e29e2a7EXAMPLE",
        "OwnerId": "123456789012",
        "Options": {
            "Igmpv2Support": "enable",
            "StaticSourcesSupport": "disable",
            "AutoAcceptSharedAssociations": "disable"
        },
        "State": "pending",
        "CreationTime": "2021-09-29T22:17:13.000Z"
    }
}
```

範例 2：建立靜態多點傳送網域

下列create-transit-gateway-multicast-domain範例會為指定的傳輸閘道建立多點傳送網域。啟用靜態來源後，您必須靜態新增來源。

```
aws ec2 create-transit-gateway-multicast-domain \
    --transit-gateway-id tgw-0bf0bffefaEXAMPLE \
    --options StaticSourcesSupport=enable,Igmpv2Support=disable
```

輸出：

```
{
    "TransitGatewayMulticastDomain": {
```

```
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-000fb24d04EXAMPLE",
        "TransitGatewayId": "tgw-0bf0bffefaEXAMPLE",
        "TransitGatewayMulticastDomainArn": "arn:aws:ec2:us-
west-2:123456789012:transit-gateway-multicast-domain/tgw-mcast-
domain-000fb24d04EXAMPLE",
        "OwnerId": "123456789012",
        "Options": {
            "Igmpv2Support": "disable",
            "StaticSourcesSupport": "enable",
            "AutoAcceptSharedAssociations": "disable"
        },
        "State": "pending",
        "CreationTime": "2021-09-29T22:20:19.000Z"
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的管理多點傳送網域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTransitGatewayMulticastDomain。

create-transit-gateway-peering-attachment

以下程式碼範例顯示如何使用 create-transit-gateway-peering-attachment。

AWS CLI

建立傳輸閘道對等互連附件

下列create-transit-gateway-peering-attachment範例會在兩個指定的傳輸閘道之間建立對等連接請求。

```
aws ec2 create-transit-gateway-peering-attachment \
    --transit-gateway-id tgw-123abc05e04123abc \
    --peer-transit-gateway-id tgw-11223344aabbcc112 \
    --peer-account-id 123456789012 \
    --peer-region us-east-2
```

輸出：

```
{
    "TransitGatewayPeeringAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
```

```
        "RequesterTgwInfo": {
            "TransitGatewayId": "tgw-123abc05e04123abc",
            "OwnerId": "123456789012",
            "Region": "us-west-2"
        },
        "AccepterTgwInfo": {
            "TransitGatewayId": "tgw-11223344aabbcc112",
            "OwnerId": "123456789012",
            "Region": "us-east-2"
        },
        "State": "initiatingRequest",
        "CreationTime": "2019-12-09T11:38:05.000Z"
    }
}
```

如需詳細資訊，請參閱《傳輸閘道指南》中的傳輸閘道對等連接。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTransitGatewayPeeringAttachment。

create-transit-gateway-policy-table

以下程式碼範例顯示如何使用 create-transit-gateway-policy-table。

AWS CLI

建立傳輸閘道政策表

下列create-transit-gateway-policy-table範例會為指定的傳輸閘道建立傳輸閘道政策表。

```
aws ec2 create-transit-gateway-policy-table \
    --transit-gateway-id tgw-067f8505c18f0bd6e
```

輸出：

```
{
    "TransitGatewayPolicyTable": {
        "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
        "TransitGatewayId": "tgw-067f8505c18f0bd6e",
        "State": "pending",
        "CreationTime": "2023-11-28T16:36:43+00:00"
    }
```

```
}
```

如需詳細資訊，請參閱 Transit Gateway 使用者指南中的 Transit Gateway 政策表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTransitGatewayPolicyTable。

create-transit-gateway-prefix-list-reference

以下程式碼範例顯示如何使用 create-transit-gateway-prefix-list-reference。

AWS CLI

建立字首清單的參考

下列create-transit-gateway-prefix-list-reference範例會建立指定傳輸閘道路由表中指定字首清單的參考。

```
aws ec2 create-transit-gateway-prefix-list-reference \
    --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
    --prefix-list-id pl-11111122222222333 \
    --transit-gateway-attachment-id tgw-attach-aaaaaabbbbbb11111
```

輸出：

```
{
    "TransitGatewayPrefixListReference": {
        "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
        "PrefixListId": "pl-11111122222222333",
        "PrefixListOwnerId": "123456789012",
        "State": "pending",
        "Blackhole": false,
        "TransitGatewayAttachment": {
            "TransitGatewayAttachmentId": "tgw-attach-aaaaaabbbbbb11111",
            "ResourceType": "vpc",
            "ResourceId": "vpc-112233445566aabbc"
        }
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的建立字首清單參考。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTransitGatewayPrefixListReference。

create-transit-gateway-route-table

以下程式碼範例顯示如何使用 create-transit-gateway-route-table。

AWS CLI

建立傳輸閘道路由表

下列create-transit-gateway-route-table範例會為指定的傳輸閘道建立路由表。

```
aws ec2 create-transit-gateway-route-table \
    --transit-gateway-id tgw-0262a0e521EXAMPLE
```

輸出：

```
{
    "TransitGatewayRouteTable": {
        "TransitGatewayRouteTableId": "tgw-rtb-0960981be7EXAMPLE",
        "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
        "State": "pending",
        "DefaultAssociationRouteTable": false,
        "DefaultPropagationRouteTable": false,
        "CreationTime": "2019-07-10T19:01:46.000Z"
    }
}
```

如需詳細資訊，請參閱《傳輸閘道指南》中的建立傳輸閘道路由表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTransitGatewayRouteTable。

create-transit-gateway-route

以下程式碼範例顯示如何使用 create-transit-gateway-route。

AWS CLI

建立傳輸閘道路由

下列create-transit-gateway-route範例會為指定的路由表建立具有指定目的地的路由。

```
aws ec2 create-transit-gateway-route \
    --destination-cidr-block 10.0.2.0/24 \
    --transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE \
```

```
    --transit-gateway-attachment-id tgw-attach-0b5968d3b6EXAMPLE
```

輸出：

```
{
    "Route": {
        "DestinationCidrBlock": "10.0.2.0/24",
        "TransitGatewayAttachments": [
            {
                "ResourceId": "vpc-0065acced4EXAMPLE",
                "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
                "ResourceType": "vpc"
            }
        ],
        "Type": "static",
        "State": "active"
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的 Transit Gateway 路由表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTransitGatewayRoute。

create-transit-gateway-vpc-attachment

以下程式碼範例顯示如何使用 create-transit-gateway-vpc-attachment。

AWS CLI

範例 1：將傳輸閘道與 VPC 建立關聯

下列create-transit-gateway-vpc-attachment範例會建立傳輸閘道連接至指定的 VPC。

```
aws ec2 create-transit-gateway-vpc-attachment \
    --transit-gateway-id tgw-0262a0e521EXAMPLE \
    --vpc-id vpc-07e8ffd50f49335df \
    --subnet-id subnet-0752213d59EXAMPLE
```

輸出：

```
{
    "TransitGatewayVpcAttachment": {
```

```
        "TransitGatewayAttachmentId": "tgw-attach-0a34fe6b4fEXAMPLE",
        "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
        "VpcId": "vpc-07e8ffd50fEXAMPLE",
        "VpcOwnerId": "111122223333",
        "State": "pending",
        "SubnetIds": [
            "subnet-0752213d59EXAMPLE"
        ],
        "CreationTime": "2019-07-10T17:33:46.000Z",
        "Options": {
            "DnsSupport": "enable",
            "Ipv6Support": "disable"
        }
    }
}
```

如需詳細資訊，請參閱《傳輸閘道指南》中的建立 VPC 的傳輸閘道連接。

範例 2：將傳輸閘道與 VPC 中的多個子網路建立關聯

下列create-transit-gateway-vpc-attachment範例會建立傳輸閘道連接至指定的 VPC 和子網路。

```
aws ec2 create-transit-gateway-vpc-attachment \
    --transit-gateway-id tgw-02f776b1a7EXAMPLE  \
    --vpc-id vpc-3EXAMPLE \
    --subnet-ids "subnet-dEXAMPLE" "subnet-6EXAMPLE"
```

輸出：

```
{
    "TransitGatewayVpcAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-0e141e0bebEXAMPLE",
        "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
        "VpcId": "vpc-3EXAMPLE",
        "VpcOwnerId": "111122223333",
        "State": "pending",
        "SubnetIds": [
            "subnet-6EXAMPLE",
            "subnet-dEXAMPLE"
        ],
        "CreationTime": "2019-12-17T20:07:52.000Z",
        "Options": {
```

```
            "DnsSupport": "enable",
            "Ipv6Support": "disable"
        }
    }
}
```

如需詳細資訊，請參閱《傳輸閘道指南》中的建立 VPC 的傳輸閘道連接。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTransitGatewayVpcAttachment。

create-transit-gateway

以下程式碼範例顯示如何使用 create-transit-gateway。

AWS CLI

建立傳輸閘道

下列create-transit-gateway範例會建立傳輸閘道。

```
aws ec2 create-transit-gateway \
    --description MyTGW \
    --
options AmazonSideAsn=64516,AutoAcceptSharedAttachments=enable,DefaultRouteTableAssociation=
```

輸出：

```
{
    "TransitGateway": {
        "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
        "TransitGatewayArn": "arn:aws:ec2:us-east-2:111122223333:transit-gateway/
tgw-0262a0e521EXAMPLE",
        "State": "pending",
        "OwnerId": "111122223333",
        "Description": "MyTGW",
        "CreationTime": "2019-07-10T14:02:12.000Z",
        "Options": {
            "AmazonSideAsn": 64516,
            "AutoAcceptSharedAttachments": "enable",
            "DefaultRouteTableAssociation": "enable",
            "AssociationDefaultRouteTableId": "tgw-rtb-018774adf3EXAMPLE",
            "DefaultRouteTablePropagation": "enable",
```

```
            "PropagationDefaultRouteTableId": "tgw-rtb-018774adf3EXAMPLE",
            "VpnEcmpSupport": "enable",
            "DnsSupport": "enable"
        }
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的建立傳輸閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTransitGateway。

create-verified-access-endpoint

以下程式碼範例顯示如何使用 create-verified-access-endpoint。

AWS CLI

建立 Verified Access 端點

下列create-verified-access-endpoint範例會為指定的 Verified Access 群組建立 Verified Access 端點。指定的網路介面和安全群組必須屬於相同的 VPC。

```
aws ec2 create-verified-access-endpoint \
    --verified-access-group-id vagr-0dbe967baf14b7235 \
    --endpoint-type network-interface \
    --attachment-type vpc \
    --domain-certificate-arn arn:aws:acm:us-east-2:123456789012:certificate/
eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE \
    --application-domain example.com \
    --endpoint-domain-prefix my-ava-app \
    --security-group-ids sg-004915970c4c8f13a \
    --network-interface-
options NetworkInterfaceId=eni-0aec70418c8d87a0f,Protocol=https,Port=443 \
    --tag-specifications ResourceType=verified-access-
endpoint,Tags=[{Key=Name,Value=my-va-endpoint}]
```

輸出：

```
{
    "VerifiedAccessEndpoint": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
```

```
        "VerifiedAccessEndpointId": "vae-066fac616d4d546f2",
        "ApplicationDomain": "example.com",
        "EndpointType": "network-interface",
        "AttachmentType": "vpc",
        "DomainCertificateArn": "arn:aws:acm:us-east-2:123456789012:certificate/
eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE",
        "EndpointDomain": "my-ava-
app.edge-00c3372d53b1540bb.vai-0ce000c0b7643abea.prod.verified-access.us-
east-2.amazonaws.com",
        "SecurityGroupIds": [
            "sg-004915970c4c8f13a"
        ],
        "NetworkInterfaceOptions": {
            "NetworkInterfaceId": "eni-0aec70418c8d87a0f",
            "Protocol": "https",
            "Port": 443
        },
        "Status": {
            "Code": "pending"
        },
        "Description": "",
        "CreationTime": "2023-08-25T20:54:43",
        "LastUpdatedTime": "2023-08-25T20:54:43",
        "Tags": [
            {
                "Key": "Name",
                "Value": "my-va-endpoint"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Verified Access 使用者指南》中的 Verified Access 端點。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVerifiedAccessEndpoint。

create-verified-access-group

以下程式碼範例顯示如何使用 create-verified-access-group。

AWS CLI

建立 Verified Access 群組

下列create-verified-access-group範例會為指定的 Verified Access 執行個體建立 Verified Access 群組。

```
aws ec2 create-verified-access-group \
    --verified-access-instance-id vai-0ce000c0b7643abea \
    --tag-specifications ResourceType=verified-access-
group,Tags=[{Key=Name,Value=my-va-group}]
```

輸出：

```
{
    "VerifiedAccessGroup": {
        "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "Description": "",
        "Owner": "123456789012",
        "VerifiedAccessGroupArn": "arn:aws:ec2:us-east-2:123456789012:verified-
access-group/vagr-0dbe967baf14b7235",
        "CreationTime": "2023-08-25T19:55:19",
        "LastUpdatedTime": "2023-08-25T19:55:19",
        "Tags": [
            {
                "Key": "Name",
                "Value": "my-va-group"
            }
        ]
    }
}
```

如需詳細資訊，請參閱 Verified Access 使用者指南中的 Verified Access 群組。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVerifiedAccessGroup。

create-verified-access-instance

以下程式碼範例顯示如何使用 create-verified-access-instance。

AWS CLI

建立 Verified Access 執行個體

下列create-verified-access-instance範例會使用名稱標籤建立 Verified Access 執行個體。

```
aws ec2 create-verified-access-instance \
    --tag-specifications ResourceType=verified-access-
instance,Tags=[{Key=Name,Value=my-va-instance}]
```

輸出：

```
{
    "VerifiedAccessInstance": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "Description": "",
        "VerifiedAccessTrustProviders": [],
        "CreationTime": "2023-08-25T18:27:56",
        "LastUpdatedTime": "2023-08-25T18:27:56",
        "Tags": [
            {
                "Key": "Name",
                "Value": "my-va-instance"
            }
        ]
    }
}
```

如需詳細資訊，請參閱 Verified Access 使用者指南中的 Verified Access 執行個體。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVerifiedAccessInstance。

create-verified-access-trust-provider

以下程式碼範例顯示如何使用 create-verified-access-trust-provider。

AWS CLI

建立 Verified Access 信任提供者

下列create-verified-access-trust-provider範例使用 AWS Identity Center 設定 Verified Access 信任提供者。

```
aws ec2 create-verified-access-trust-provider \
    --trust-provider-type user \
    --user-trust-provider-type iam-identity-center \
    --policy-reference-name idc \
```

```
    --tag-specifications ResourceType=verified-access-trust-
provider,Tags=[{Key=Name,Value=my-va-trust-provider}]
```

輸出：

```
{
    "VerifiedAccessTrustProvider": {
        "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
        "Description": "",
        "TrustProviderType": "user",
        "UserTrustProviderType": "iam-identity-center",
        "PolicyReferenceName": "idc",
        "CreationTime": "2023-08-25T18:40:36",
        "LastUpdatedTime": "2023-08-25T18:40:36",
        "Tags": [
            {
                "Key": "Name",
                "Value": "my-va-trust-provider"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Verified Access 使用者指南》中的 Verified Access 的信任提供者。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVerifiedAccessTrustProvider。

create-volume

以下程式碼範例顯示如何使用 create-volume。

AWS CLI

建立空的一般用途 SSD (gp2) 磁碟區

下列create-volume範例會在指定的可用區域中建立 80 GiB 一般用途 SSD (gp2) 磁碟區。請注意，目前區域必須是 us-east-1，或者您可以新增 --region 參數來指定命令的區域。

```
aws ec2 create-volume \
    --volume-type gp2 \
    --size 80 \
    --availability-zone us-east-1a
```

輸出：

```
{
    "AvailabilityZone": "us-east-1a",
    "Tags": [],
    "Encrypted": false,
    "VolumeType": "gp2",
    "VolumeId": "vol-1234567890abcdef0",
    "State": "creating",
    "Iops": 240,
    "SnapshotId": "",
    "CreateTime": "YYYY-MM-DDTHH:MM:SS.000Z",
    "Size": 80
}
```

如果您未指定磁碟區類型，預設磁碟區類型為 gp2。

```
aws ec2 create-volume \
    --size 80 \
    --availability-zone us-east-1a
```

範例 2：從快照建立佈建 IOPS SSD (io1) 磁碟區

下列create-volume範例會使用指定的快照，在指定的可用區域中建立具有 1000 個佈建 IOPS 的佈建 IOPS SSD (io1) 磁碟區。

```
aws ec2 create-volume \
    --volume-type io1 \
    --iops 1000 \
    --snapshot-id snap-066877671789bd71b \
    --availability-zone us-east-1a
```

輸出：

```
{
    "AvailabilityZone": "us-east-1a",
    "Tags": [],
    "Encrypted": false,
    "VolumeType": "io1",
    "VolumeId": "vol-1234567890abcdef0",
    "State": "creating",
    "Iops": 1000,
```

```
    "SnapshotId": "snap-066877671789bd71b",
    "CreateTime": "YYYY-MM-DDTHH:MM:SS.000Z",
    "Size": 500
}
```

範例 3：建立加密磁碟區

下列create-volume範例使用 EBS 加密的預設 CMK 建立加密磁碟區。如果預設停用加密，您必須指定 --encrypted 參數，如下所示。

```
aws ec2 create-volume \
    --size 80 \
    --encrypted \
    --availability-zone us-east-1a
```

輸出：

```
{
    "AvailabilityZone": "us-east-1a",
    "Tags": [],
    "Encrypted": true,
    "VolumeType": "gp2",
    "VolumeId": "vol-1234567890abcdef0",
    "State": "creating",
    "Iops": 240,
    "SnapshotId": "",
    "CreateTime": "YYYY-MM-DDTHH:MM:SS.000Z",
    "Size": 80
}
```

如果預設啟用加密，即使沒有 --encrypted 參數，下列範例命令也會建立加密的磁碟區。

```
aws ec2 create-volume \
    --size 80 \
    --availability-zone us-east-1a
```

如果您使用 --kms-key-id 參數來指定客戶受管 CMK，即使預設啟用加密，仍必須指定 --encrypted 參數。

```
aws ec2 create-volume \
    --volume-type gp2 \
```

```
    --size 80 \
    --encrypted \
    --kms-key-id 0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE \
    --availability-zone us-east-1a
```

範例 4：使用標籤建立磁碟區

下列create-volume範例會建立磁碟區並新增兩個標籤。

```
aws ec2 create-volume \
    --availability-zone us-east-1a \
    --volume-type gp2 \
    --size 80 \
    --tag-specifications 'ResourceType=volume,Tags=[{Key=purpose,Value=production},
{Key=cost-center,Value=cc123}]'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVolume。

create-vpc-endpoint-connection-notification

以下程式碼範例顯示如何使用 create-vpc-endpoint-connection-notification。

AWS CLI

建立端點連線通知

此範例會針對特定端點服務建立通知，在介面端點連線至您的服務，以及服務已接受端點時提醒您。

命令：

```
aws ec2 create-vpc-endpoint-connection-notification --connection-notification-
arn arn:aws:sns:us-east-2:123456789012:VpceNotification --connection-
events Connect Accept --service-id vpce-svc-1237881c0d25a3abc
```

輸出：

```
{
   "ConnectionNotification": {
       "ConnectionNotificationState": "Enabled",
       "ConnectionNotificationType": "Topic",
```

```
        "ServiceId": "vpce-svc-1237881c0d25a3abc",
        "ConnectionEvents": [
            "Accept",
            "Connect"
        ],
        "ConnectionNotificationId": "vpce-nfn-008776de7e03f5abc",
        "ConnectionNotificationArn": "arn:aws:sns:us-
east-2:123456789012:VpceNotification"
    }
 }
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVpcEndpointConnectionNotification。

create-vpc-endpoint-service-configuration

以下程式碼範例顯示如何使用 create-vpc-endpoint-service-configuration。

AWS CLI

範例 1：為介面端點建立端點服務組態

下列create-vpc-endpoint-service-configuration範例會使用 Network Load Balancer 建立 VPC 端點服務組態nlb-vpce。此範例也會指定必須接受透過介面端點連線至服務的請求。

```
aws ec2 create-vpc-endpoint-service-configuration \
    --network-load-balancer-arns arn:aws:elasticloadbalancing:us-
east-1:123456789012:loadbalancer/net/nlb-vpce/e94221227f1ba532 \
    --acceptance-required
```

輸出：

```
{
    "ServiceConfiguration": {
        "ServiceType": [
            {
                "ServiceType": "Interface"
            }
        ],
        "NetworkLoadBalancerArns": [
            "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/net/
nlb-vpce/e94221227f1ba532"
```

```
        ],
        "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-03d5ebb7d9579a2b3",
        "ServiceState": "Available",
        "ServiceId": "vpce-svc-03d5ebb7d9579a2b3",
        "AcceptanceRequired": true,
        "AvailabilityZones": [
            "us-east-1d"
        ],
        "BaseEndpointDnsNames": [
            "vpce-svc-03d5ebb7d9579a2b3.us-east-1.vpce.amazonaws.com"
        ]
    }
}
```

如需詳細資訊，請參閱《 AWS PrivateLink 使用者指南》中的建立端點服務。

範例 2：為 Gateway Load Balancer 端點建立端點服務組態

下列create-vpc-endpoint-service-configuration範例使用 Gateway Load Balancer 建立 VPC 端點服務組態GWLBService。透過 Gateway Load Balancer 端點連線到服務的請求會自動接受。

```
aws ec2 create-vpc-endpoint-service-configuration \
    --gateway-load-balancer-arns arn:aws:elasticloadbalancing:us-
east-1:123456789012:loadbalancer/gwy/GWLBService/123123123123abcc \
    --no-acceptance-required
```

輸出：

```
{
    "ServiceConfiguration": {
        "ServiceType": [
            {
                "ServiceType": "GatewayLoadBalancer"
            }
        ],
        "ServiceId": "vpce-svc-123123a1c43abc123",
        "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-123123a1c43abc123",
        "ServiceState": "Available",
        "AvailabilityZones": [
            "us-east-1d"
        ],
```

```
        "AcceptanceRequired": false,
        "ManagesVpcEndpoints": false,
        "GatewayLoadBalancerArns": [
            "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/gwy/
GWLBService/123123123123abcc"
        ]
    }
}
```

如需詳細資訊，請參閱 AWS PrivateLink 使用者指南中的建立閘道Load Balancer端點服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVpcEndpointServiceConfiguration。

`create-vpc-endpoint`

以下程式碼範例顯示如何使用 `create-vpc-endpoint`。

AWS CLI

範例 1：建立閘道端點

下列create-vpc-endpoint範例會在 us-east-1區域中的 VPC vpc-1a2b3c4d和 Amazon S3 之間建立閘道 VPC 端點，並將路由表rtb-11aa22bb與端點建立關聯。

```
aws ec2 create-vpc-endpoint \
    --vpc-id vpc-1a2b3c4d \
    --service-name com.amazonaws.us-east-1.s3 \
    --route-table-ids rtb-11aa22bb
```

輸出：

```
{
    "VpcEndpoint": {
        "PolicyDocument": "{\"Version\":\"2008-10-17\",\"Statement\":[{\"Sid\":\"\",
\"Effect\":\"Allow\",\"Principal\":\"*\",\"Action\":\"*\",\"Resource\":\"*\"}]}",
        "VpcId": "vpc-1a2b3c4d",
        "State": "available",
        "ServiceName": "com.amazonaws.us-east-1.s3",
        "RouteTableIds": [
            "rtb-11aa22bb"
        ],
```

```
        "VpcEndpointId": "vpc-1a2b3c4d",
        "CreationTimestamp": "2015-05-15T09:40:50Z"
    }
}
```

如需詳細資訊，請參閱 AWS PrivateLink 使用者指南中的建立閘道端點。

範例 2：建立介面端點

下列create-vpc-endpoint範例會在 us-east-1區域中的 VPC vpc-1a2b3c4d和 Amazon S3 之間建立介面 VPC 端點。命令會在子網路 中建立端點subnet-1a2b3c4d，將其與安全群組 建立關聯sg-1a2b3c4d，並新增索引鍵為 "Service" 且值為 "S3" 的標籤。

```
aws ec2 create-vpc-endpoint \
    --vpc-id vpc-1a2b3c4d \
    --vpc-endpoint-type Interface \
    --service-name com.amazonaws.us-east-1.s3 \
    --subnet-ids subnet-7b16de0c \
    --security-group-id sg-1a2b3c4d \
    --tag-specifications ResourceType=vpc-endpoint,Tags=[{Key=service,Value=S3}]
```

輸出：

```
{
    "VpcEndpoint": {
        "VpcEndpointId": "vpce-1a2b3c4d5e6f1a2b3",
        "VpcEndpointType": "Interface",
        "VpcId": "vpc-1a2b3c4d",
        "ServiceName": "com.amazonaws.us-east-1.s3",
        "State": "pending",
        "RouteTableIds": [],
        "SubnetIds": [
            "subnet-1a2b3c4d"
        ],
        "Groups": [
            {
                "GroupId": "sg-1a2b3c4d",
                "GroupName": "default"
            }
        ],
        "PrivateDnsEnabled": false,
        "RequesterManaged": false,
```

```
        "NetworkInterfaceIds": [
            "eni-0b16f0581c8ac6877"
        ],
        "DnsEntries": [
            {
                "DnsName": "*.vpce-1a2b3c4d5e6f1a2b3-9hnenorg.s3.us-
east-1.vpce.amazonaws.com",
                "HostedZoneId": "Z7HUB22UULQXV"
            },
            {
                "DnsName": "*.vpce-1a2b3c4d5e6f1a2b3-9hnenorg-us-east-1c.s3.us-
east-1.vpce.amazonaws.com",
                "HostedZoneId": "Z7HUB22UULQXV"
            }
        ],
        "CreationTimestamp": "2021-03-05T14:46:16.030000+00:00",
        "Tags": [
            {
                "Key": "service",
                "Value": "S3"
            }
        ],
        "OwnerId": "123456789012"
    }
}
```

如需詳細資訊，請參閱《 AWS PrivateLink 使用者指南》中的建立介面 VPC 端點。

範例 3：建立 Gateway Load Balancer 端點

下列create-vpc-endpoint範例會在 VPC vpc-111122223333aabbc和 之間建立 Gateway Load Balancer 端點，以及使用 Gateway Load Balancer 設定的服務。

```
aws ec2 create-vpc-endpoint \
    --service-name com.amazonaws.vpce.us-east-1.vpce-svc-123123a1c43abc123 \
    --vpc-endpoint-type GatewayLoadBalancer \
    --vpc-id vpc-111122223333aabbc \
    --subnet-ids subnet-0011aabbcc2233445
```

輸出：

```
{
    "VpcEndpoint": {
```

```
        "VpcEndpointId": "vpce-aabbaabbaabbaabba",
        "VpcEndpointType": "GatewayLoadBalancer",
        "VpcId": "vpc-111122223333aabbc",
        "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-123123a1c43abc123",
        "State": "pending",
        "SubnetIds": [
            "subnet-0011aabbcc2233445"
        ],
        "RequesterManaged": false,
        "NetworkInterfaceIds": [
            "eni-01010120203030405"
        ],
        "CreationTimestamp": "2020-11-11T08:06:03.522Z",
        "OwnerId": "123456789012"
    }
}
```

如需詳細資訊，請參閱 AWS PrivateLink 使用者指南中的 Gateway Load Balancer 端點。

範例 4：建立資源端點

下列create-vpc-endpoint範例會建立資源端點。

```
aws ec2 create-vpc-endpoint \
    --vpc-endpoint-type Resource \
    --vpc-id vpc-111122223333aabbc \
    --subnet-ids subnet-0011aabbcc2233445 \
    --resource-configuration-arn arn:aws:vpc-lattice-us-
east-1:123456789012:resourceconfiguration/rcfg-0123abcde98765432
```

輸出：

```
{
    "VpcEndpoint": {
        "VpcEndpointId": "vpce-00939a7ed9EXAMPLE",
        "VpcEndpointType": "Resource",
        "VpcId": "vpc-111122223333aabbc",
        "State": "Pending",
        "SubnetIds": [
            "subnet-0011aabbcc2233445"
        ],
        "Groups": [
            {
```

```
                "GroupId": "sg-03e2f15fbfc09b000",
                "GroupName": "default"
            }
        ],
        "IpAddressType": "IPV4",
        "PrivateDnsEnabled": false,
        "CreationTimestamp": "2025-02-06T23:38:49.525000+00:00",
        "Tags": [],
        "OwnerId": "123456789012",
        "ResourceConfigurationArn": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceconfiguration/rcfg-0123abcde98765432"
    }
}
```

如需詳細資訊，請參閱 AWS PrivateLink 使用者指南中的資源端點。

範例 5：建立服務網路端點

下列create-vpc-endpoint範例會建立服務網路端點。

```
aws ec2 create-vpc-endpoint \
    --vpc-endpoint-type ServiceNetwork \
    --vpc-id vpc-111122223333aabbc \
    --subnet-ids subnet-0011aabbcc2233445 \
    --service-network-arn arn:aws:vpc-lattice:us-east-1:123456789012:servicenetwork/
sn-0101abcd5432abcd0 \
    --security-group-ids sg-0123456789012abcd
```

輸出：

```
{
    "VpcEndpoint": {
        "VpcEndpointId": "vpce-0f00567fa8EXAMPLE",
        "VpcEndpointType": "ServiceNetwork",
        "VpcId": "vpc-111122223333aabbc",
        "State": "Pending",
        "SubnetIds": [
            "subnet-0011aabbcc2233445"
        ],
        "Groups": [
            {
                "GroupId": "sg-0123456789012abcd",
                "GroupName": "my-security-group"
```

```
            }
        ],
        "IpAddressType": "IPV4",
        "PrivateDnsEnabled": false,
        "CreationTimestamp": "2025-02-06T23:44:20.449000+00:00",
        "Tags": [],
        "OwnerId": "123456789012",
        "ServiceNetworkArn": "arn:aws:vpc-lattice:us-
east-1:123456789012:servicenetwork/sn-0101abcd5432abcd0"
    }
}
```

如需詳細資訊，請參閱 AWS PrivateLink 使用者指南中的服務網路端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVpcEndpoint。

`create-vpc-peering-connection`

以下程式碼範例顯示如何使用 `create-vpc-peering-connection`。

AWS CLI

在 VPC 之間建立 VPCs 對等互連

此範例會請求 VPCs-1a2b3c4d 和 vpc-11122233 之間的對等連線。

命令：

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-
id vpc-11122233
```

輸出：

```
{
    "VpcPeeringConnection": {
        "Status": {
            "Message": "Initiating Request to 444455556666",
            "Code": "initiating-request"
        },
        "Tags": [],
        "RequesterVpcInfo": {
            "OwnerId": "444455556666",
```

```
            "VpcId": "vpc-1a2b3c4d",
            "CidrBlock": "10.0.0.0/28"
        },
        "VpcPeeringConnectionId": "pcx-111aaa111",
        "ExpirationTime": "2014-04-02T16:13:36.000Z",
        "AccepterVpcInfo": {
            "OwnerId": "444455556666",
            "VpcId": "vpc-11122233"
        }
    }
}
```

使用另一個帳戶中的 VPC 建立 VPC 對等互連

此範例會請求 VPC (vpc-1a2b3c4d) 與屬於 AWS 帳戶 123456789012 的 VPC (vpc-11122233) 之間的對等連線。

命令：

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-
id vpc-11122233 --peer-owner-id 123456789012
```

使用不同區域中的 VPC 建立 VPC 對等互連

此範例會請求目前區域中的 VPC (vpc-1a2b3c4d) 與區域中您帳戶中的 VPC (vpc-11122233) 之間的對等連線us-west-2。

命令：

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-
id vpc-11122233 --peer-region us-west-2
```

此範例會請求目前區域中的 VPC (vpc-1a2b3c4d) 與屬於該us-west-2區域中 AWS 帳戶 123456789012 的 VPC (vpc-11122233) 之間的對等連線。

命令：

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-
id vpc-11122233 --peer-owner-id 123456789012 --peer-region us-west-2
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVpcPeeringConnection。

create-vpc

以下程式碼範例顯示如何使用 create-vpc。

AWS CLI

範例 1：建立 VPC

以下 create-vpc 範例會建立具有指定 IPv4 CIDR 區塊和名稱標籤的 VPC。

```
aws ec2 create-vpc \
    --cidr-block 10.0.0.0/16 \
    --tag-specifications ResourceType=vpc,Tags=[{Key=Name,Value=MyVpc}]
```

輸出：

```
{
    "Vpc": {
        "CidrBlock": "10.0.0.0/16",
        "DhcpOptionsId": "dopt-5EXAMPLE",
        "State": "pending",
        "VpcId": "vpc-0a60eb65b4EXAMPLE",
        "OwnerId": "123456789012",
        "InstanceTenancy": "default",
        "Ipv6CidrBlockAssociationSet": [],
        "CidrBlockAssociationSet": [
            {
                "AssociationId": "vpc-cidr-assoc-07501b79ecEXAMPLE",
                "CidrBlock": "10.0.0.0/16",
                "CidrBlockState": {
                    "State": "associated"
                }
            }
        ],
        "IsDefault": false,
        "Tags": [
            {
                "Key": "Name",
                "Value": MyVpc"
            }
        ]
    }
}
```

範例 2：建立具有專用租用的 VPC

以下 `create-vpc` 範例會建立具有指定 IPv4 CIDR 區塊和專用租用的 VPC。

```
aws ec2 create-vpc \
    --cidr-block 10.0.0.0/16 \
    --instance-tenancy dedicated
```

輸出：

```
{
    "Vpc": {
        "CidrBlock": "10.0.0.0/16",
        "DhcpOptionsId": "dopt-19edf471",
        "State": "pending",
        "VpcId": "vpc-0a53287fa4EXAMPLE",
        "OwnerId": "111122223333",
        "InstanceTenancy": "dedicated",
        "Ipv6CidrBlockAssociationSet": [],
        "CidrBlockAssociationSet": [
            {
                "AssociationId": "vpc-cidr-assoc-00b24cc1c2EXAMPLE",
                "CidrBlock": "10.0.0.0/16",
                "CidrBlockState": {
                    "State": "associated"
                }
            }
        ],
        "IsDefault": false
    }
}
```

範例 3：建立具有 IPv6 CIDR 區塊的 VPC

以下 `create-vpc` 範例會建立具有 Amazon 提供之 IPv6 CIDR 區塊的 VPC。

```
aws ec2 create-vpc \
    --cidr-block 10.0.0.0/16 \
    --amazon-provided-ipv6-cidr-block
```

輸出：

```
{
    "Vpc": {
        "CidrBlock": "10.0.0.0/16",
        "DhcpOptionsId": "dopt-dEXAMPLE",
        "State": "pending",
        "VpcId": "vpc-0fc5e3406bEXAMPLE",
        "OwnerId": "123456789012",
        "InstanceTenancy": "default",
        "Ipv6CidrBlockAssociationSet": [
            {
                "AssociationId": "vpc-cidr-assoc-068432c60bEXAMPLE",
                "Ipv6CidrBlock": "",
                "Ipv6CidrBlockState": {
                    "State": "associating"
                },
                "Ipv6Pool": "Amazon",
                "NetworkBorderGroup": "us-west-2"
            }
        ],
        "CidrBlockAssociationSet": [
            {
                "AssociationId": "vpc-cidr-assoc-0669f8f9f5EXAMPLE",
                "CidrBlock": "10.0.0.0/16",
                "CidrBlockState": {
                    "State": "associated"
                }
            }
        ],
        "IsDefault": false
    }
}
```

範例 4：使用來自 IPAM 集區的 CIDR 建立 VPC

以下 `create-vpc` 範例會使用來自 Amazon VPC IP 位址管理器 (IPAM) 集區的 CIDR 建立 VPC。

Linux 和 macOS：

```
aws ec2 create-vpc \
    --ipv4-ipam-pool-id ipam-pool-0533048da7d823723 \
    --tag-specifications ResourceType=vpc,Tags='[{Key=Environment,Value="Preprod"},
{Key=Owner,Value="Build Team"}]'
```

Windows：

```
aws ec2 create-vpc ^
    --ipv4-ipam-pool-id ipam-pool-0533048da7d823723 ^
    --tag-specifications ResourceType=vpc,Tags=[{Key=Environment,Value="Preprod"},
{Key=Owner,Value="Build Team"}]
```

輸出：

```
{
    "Vpc": {
        "CidrBlock": "10.0.1.0/24",
        "DhcpOptionsId": "dopt-2afccf50",
        "State": "pending",
        "VpcId": "vpc-010e1791024eb0af9",
        "OwnerId": "123456789012",
        "InstanceTenancy": "default",
        "Ipv6CidrBlockAssociationSet": [],
        "CidrBlockAssociationSet": [
            {
                "AssociationId": "vpc-cidr-assoc-0a77de1d803226d4b",
                "CidrBlock": "10.0.1.0/24",
                "CidrBlockState": {
                    "State": "associated"
                }
            }
        ],
        "IsDefault": false,
        "Tags": [
            {
                "Key": "Environment",
                "Value": "Preprod"
            },
            {
                "Key": "Owner",
                "Value": "Build Team"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的建立使用 IPAM 集區 CIDR的 VPC。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateVpc。

create-vpn-connection-route

以下程式碼範例顯示如何使用 create-vpn-connection-route。

AWS CLI

建立 VPN 連接的靜態路由

此範例會為指定的 VPN 連線建立靜態路由。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 create-vpn-connection-route --vpn-connection-id vpn-40f41529 --destination-cidr-block 11.12.0.0/16
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVpnConnectionRoute。

create-vpn-connection

以下程式碼範例顯示如何使用 create-vpn-connection。

AWS CLI

範例 1：使用動態路由建立 VPN 連接

下列create-vpn-connection範例會在指定的虛擬私有閘道與指定的客戶閘道之間建立 VPN 連線，並將標籤套用至 VPN 連線。輸出包含客戶閘道裝置的組態資訊，格式為 XML。

```
aws ec2 create-vpn-connection \
    --type ipsec.1 \
    --customer-gateway-id cgw-001122334455aabbc \
    --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \
    --tag-specification 'ResourceType=vpn-connection,Tags=[{Key=Name,Value=BGP-VPN}]'
```

輸出：

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": "...configuration information...",
```

```
        "CustomerGatewayId": "cgw-001122334455aabbc",
        "Category": "VPN",
        "State": "pending",
        "VpnConnectionId": "vpn-123123123123abcab",
        "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b2",
        "Options": {
            "EnableAcceleration": false,
            "StaticRoutesOnly": false,
            "LocalIpv4NetworkCidr": "0.0.0.0/0",
            "RemoteIpv4NetworkCidr": "0.0.0.0/0",
            "TunnelInsideIpVersion": "ipv4",
            "TunnelOptions": [
                {},
                {}
            ]
        },
        "Routes": [],
        "Tags": [
             {
                "Key": "Name",
                "Value": "BGP-VPN"
            }
        ]
    }
}
```

如需詳細資訊，請參閱AWS 《 Site-to-Site VPN 使用者指南》中的 HowSite-to-Site VPN 運作。AWS Site-to-Site

範例 2：使用靜態路由建立 VPN 連接

下列create-vpn-connection範例會在指定的虛擬私有閘道與指定的客戶閘道之間建立 VPN 連線。選項會指定靜態路由。輸出包含客戶閘道裝置的組態資訊，格式為 XML。

```
aws ec2 create-vpn-connection \
    --type ipsec.1 \
    --customer-gateway-id cgw-001122334455aabbc \
    --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \
    --options "{\"StaticRoutesOnly\":true}"
```

輸出：

```
{
```

```
    "VpnConnection": {
        "CustomerGatewayConfiguration": "..configuration information...",
        "CustomerGatewayId": "cgw-001122334455aabbc",
        "Category": "VPN",
        "State": "pending",
        "VpnConnectionId": "vpn-123123123123abcab",
        "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b2",
        "Options": {
            "EnableAcceleration": false,
            "StaticRoutesOnly": true,
            "LocalIpv4NetworkCidr": "0.0.0.0/0",
            "RemoteIpv4NetworkCidr": "0.0.0.0/0",
            "TunnelInsideIpVersion": "ipv4",
            "TunnelOptions": [
                {},
                {}
            ]
        },
        "Routes": [],
        "Tags": []
    }
}
```

如需詳細資訊，請參閱AWS 《 Site-to-Site VPN 使用者指南》中的 HowSite-to-Site VPN 運作。AWS Site-to-Site

範例 3：建立 VPN 連線並指定您自己的 CIDR 和預先共用金鑰

下列create-vpn-connection範例會建立 VPN 連線，並指定每個通道的內部 IP 地址 CIDR 區塊和自訂預先共用金鑰。指定的值會在CustomerGatewayConfiguration資訊中傳回。

```
aws ec2 create-vpn-connection \
    --type ipsec.1 \
    --customer-gateway-id cgw-001122334455aabbc \
    --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \
    --options
 TunnelOptions='[{TunnelInsideCidr=169.254.12.0/30,PreSharedKey=ExamplePreSharedKey1},
{TunnelInsideCidr=169.254.13.0/30,PreSharedKey=ExamplePreSharedKey2}]'
```

輸出：

```
{
```

```
    "VpnConnection": {
        "CustomerGatewayConfiguration": "..configuration information...",
        "CustomerGatewayId": "cgw-001122334455aabbc",
        "Category": "VPN",
        "State": "pending",
        "VpnConnectionId": "vpn-123123123123abcab",
        "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b2",
        "Options": {
            "EnableAcceleration": false,
            "StaticRoutesOnly": false,
            "LocalIpv4NetworkCidr": "0.0.0.0/0",
            "RemoteIpv4NetworkCidr": "0.0.0.0/0",
            "TunnelInsideIpVersion": "ipv4",
            "TunnelOptions": [
                {
                    "OutsideIpAddress": "203.0.113.3",
                    "TunnelInsideCidr": "169.254.12.0/30",
                    "PreSharedKey": "ExamplePreSharedKey1"
                },
                {
                    "OutsideIpAddress": "203.0.113.5",
                    "TunnelInsideCidr": "169.254.13.0/30",
                    "PreSharedKey": "ExamplePreSharedKey2"
                }
            ]
        },
        "Routes": [],
        "Tags": []
    }
}
```

如需詳細資訊，請參閱AWS 《 Site-to-Site VPN 使用者指南》中的 HowSite-to-Site VPN 運作。AWS Site-to-Site

範例 4：建立支援 IPv6 流量的 VPN 連線

下列create-vpn-connection範例會建立 VPN 連線，以支援指定傳輸閘道與指定客戶閘道之間的 IPv6 流量。兩個通道的通道選項會指定 AWS 必須啟動 IKE 交涉。

```
aws ec2 create-vpn-connection \
    --type ipsec.1 \
    --transit-gateway-id tgw-12312312312312312 \
    --customer-gateway-id cgw-001122334455aabbc \
```

```
    --options TunnelInsideIpVersion=ipv6,TunnelOptions=[{StartupAction=start},
{StartupAction=start}]
```

輸出：

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": "..configuration information...",
        "CustomerGatewayId": "cgw-001122334455aabbc",
        "Category": "VPN",
        "State": "pending",
        "VpnConnectionId": "vpn-11111111122222222",
        "TransitGatewayId": "tgw-12312312312312312",
        "Options": {
            "EnableAcceleration": false,
            "StaticRoutesOnly": false,
            "LocalIpv6NetworkCidr": "::/0",
            "RemoteIpv6NetworkCidr": "::/0",
            "TunnelInsideIpVersion": "ipv6",
            "TunnelOptions": [
                {
                    "OutsideIpAddress": "203.0.113.3",
                    "StartupAction": "start"
                },
                {
                    "OutsideIpAddress": "203.0.113.5",
                    "StartupAction": "start"
                }
            ]
        },
        "Routes": [],
        "Tags": []
    }
}
```

如需詳細資訊，請參閱AWS 《 Site-to-Site VPN 使用者指南》中的 HowSite-to-Site VPN 運作。AWS Site-to-Site

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVpnConnection。

create-vpn-gateway

以下程式碼範例顯示如何使用 create-vpn-gateway。

AWS CLI

建立虛擬私有閘道

此範例會建立虛擬私有閘道。

命令：

```
aws ec2 create-vpn-gateway --type ipsec.1
```

輸出：

```
{
    "VpnGateway": {
        "AmazonSideAsn": 64512,
        "State": "available",
        "Type": "ipsec.1",
        "VpnGatewayId": "vgw-9a4cacf3",
        "VpcAttachments": []
    }
}
```

使用特定 Amazon 端 ASN 建立虛擬私有閘道

此範例會建立虛擬私有閘道，並指定 BGP 工作階段之 Amazon 端的自治系統編號 (ASN)。

命令：

```
aws ec2 create-vpn-gateway --type ipsec.1 --amazon-side-asn 65001
```

輸出：

```
{
    "VpnGateway": {
        "AmazonSideAsn": 65001,
        "State": "available",
        "Type": "ipsec.1",
        "VpnGatewayId": "vgw-9a4cacf3",
        "VpcAttachments": []
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVpnGateway。

delete-carrier-gateway

以下程式碼範例顯示如何使用 delete-carrier-gateway。

AWS CLI

刪除您的電信業者閘道

下列delete-carrier-gateway範例會刪除指定的電信業者閘道。

```
aws ec2 delete-carrier-gateway \
    --carrier-gateway-id cagw-0465cdEXAMPLE1111
```

輸出：

```
{
    "CarrierGateway": {
        "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
        "VpcId": "vpc-0c529aEXAMPLE1111",
        "State": "deleting",
        "OwnerId": "123456789012"
    }
}
```

如需詳細資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的電信業者閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCarrierGateway。

delete-client-vpn-endpoint

以下程式碼範例顯示如何使用 delete-client-vpn-endpoint。

AWS CLI

刪除 Client VPN 端點

下列delete-client-vpn-endpoint範例會刪除指定的 Client VPN 端點。

```
aws ec2 delete-client-vpn-endpoint \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

輸出：

```
{
    "Status": {
        "Code": "deleting"
    }
}
```

如需詳細資訊，請參閱 Client VPN 管理員指南中的 Client VPN 端點。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteClientVpnEndpoint。

delete-client-vpn-route

以下程式碼範例顯示如何使用 delete-client-vpn-route。

AWS CLI

刪除 Client VPN 端點的路由

下列delete-client-vpn-route範例會刪除 Client VPN 端點指定子網路的0.0.0.0/0路由。

```
aws ec2 delete-client-vpn-route \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --destination-cidr-block 0.0.0.0/0 \
    --target-vpc-subnet-id subnet-0123456789abcabca
```

輸出：

```
{
    "Status": {
        "Code": "deleting"
    }
}
```

如需詳細資訊，請參閱 AWS Client VPN 管理員指南中的路由。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteClientVpnRoute。

delete-coip-cidr

以下程式碼範例顯示如何使用 delete-coip-cidr。

AWS CLI

刪除客戶擁有的 IP (CoIP) 地址範圍

下列delete-coip-cidr範例會刪除指定 CoIP 集區中指定的 CoIP 地址範圍。

```
aws ec2 delete-coip-cidr \
    --cidr 14.0.0.0/24 \
    --coip-pool-id ipv4pool-coip-1234567890abcdefg
```

輸出：

```
{
    "CoipCidr": {
        "Cidr": "14.0.0.0/24",
        "CoipPoolId": "ipv4pool-coip-1234567890abcdefg",
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890"
    }
}
```

如需詳細資訊，請參閱 AWS Outposts 使用者指南中的客戶擁有的 IP 地址。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCoipCidr。

delete-coip-pool

以下程式碼範例顯示如何使用 delete-coip-pool。

AWS CLI

刪除客戶擁有的 IP (CoIP) 地址集區

下列delete-coip-pool範例會刪除 CoIP 地址的 CoIP 集區。

```
aws ec2 delete-coip-pool \
    --coip-pool-id ipv4pool-coip-1234567890abcdefg
```

輸出：

```
{
    "CoipPool": {
```

```
        "PoolId": "ipv4pool-coip-1234567890abcdefg",
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
        "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-
coip-1234567890abcdefg"
    }
}
```

如需詳細資訊，請參閱 AWS Outposts 使用者指南中的客戶擁有的 IP 地址。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCoipPool。

`delete-customer-gateway`

以下程式碼範例顯示如何使用 `delete-customer-gateway`。

AWS CLI

刪除客戶閘道

此範例會刪除指定的客戶閘道。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 delete-customer-gateway --customer-gateway-id cgw-0e11f167
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCustomerGateway。

`delete-dhcp-options`

以下程式碼範例顯示如何使用 `delete-dhcp-options`。

AWS CLI

刪除 DHCP 選項集

此範例會刪除指定的 DHCP 選項集。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 delete-dhcp-options --dhcp-options-id dopt-d9070ebb
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDhcpOptions。

delete-egress-only-internet-gateway

以下程式碼範例顯示如何使用 delete-egress-only-internet-gateway。

AWS CLI

刪除輸出限定網際網路閘道

此範例會刪除指定的輸出限定網際網路閘道。

命令：

```
aws ec2 delete-egress-only-internet-gateway --egress-only-internet-gateway-
id eigw-01eadbd45ecd7943f
```

輸出：

```
{
  "ReturnCode": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteEgressOnlyInternetGateway。

delete-fleets

以下程式碼範例顯示如何使用 delete-fleets。

AWS CLI

範例 1：刪除 EC2 機群並終止相關聯的執行個體

下列delete-fleets範例會刪除指定的 EC2 機群，並終止相關聯的隨需執行個體和 Spot 執行個體。

```
aws ec2 delete-fleets \
    --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
    --terminate-instances
```

輸出：

```
{
    "SuccessfulFleetDeletions": [
```

```
        {
            "CurrentFleetState": "deleted_terminating",
            "PreviousFleetState": "active",
            "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
        }
    ],
    "UnsuccessfulFleetDeletions": []
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud Linux 執行個體使用者指南》中的刪除 EC2 機群。

範例 2：刪除 EC2 機群而不終止相關聯的執行個體

下列delete-fleets範例會刪除指定的 EC2 機群，而不終止相關聯的隨需執行個體和 Spot 執行個體。

```
aws ec2 delete-fleets \
    --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
    --no-terminate-instances
```

輸出：

```
{
    "SuccessfulFleetDeletions": [
        {
            "CurrentFleetState": "deleted_running",
            "PreviousFleetState": "active",
            "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
        }
    ],
    "UnsuccessfulFleetDeletions": []
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud Linux 執行個體使用者指南》中的刪除 EC2 機群。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFleets。

delete-flow-logs

以下程式碼範例顯示如何使用 delete-flow-logs。

AWS CLI

刪除流程日誌

下列delete-flow-logs範例會刪除指定的流程日誌。

```
aws ec2 delete-flow-logs --flow-log-id fl-11223344556677889
```

輸出：

```
{
    "Unsuccessful": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFlowLogs。

delete-fpga-image

以下程式碼範例顯示如何使用 delete-fpga-image。

AWS CLI

刪除 Amazon FPGA 映像

此範例會刪除指定的 AFI。

命令：

```
aws ec2 delete-fpga-image --fpga-image-id afi-06b12350a123fbabc
```

輸出：

```
{
  "Return": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFpgaImage。

delete-instance-connect-endpoint

以下程式碼範例顯示如何使用 delete-instance-connect-endpoint。

AWS CLI

刪除 EC2 Instance Connect Endpoint

下列delete-instance-connect-endpoint範例會刪除指定的 EC2 Instance Connect Endpoint。

```
aws ec2 delete-instance-connect-endpoint \
    --instance-connect-endpoint-id eice-03f5e49b83924bbc7
```

輸出：

```
{
    "InstanceConnectEndpoint": {
        "OwnerId": "111111111111",
        "InstanceConnectEndpointId": "eice-0123456789example",
        "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-
connect-endpoint/eice-0123456789example",
        "State": "delete-in-progress",
        "StateMessage": "",
        "NetworkInterfaceIds": [],
        "VpcId": "vpc-0123abcd",
        "AvailabilityZone": "us-east-1d",
        "CreatedAt": "2023-02-07T12:05:37+00:00",
        "SubnetId": "subnet-0123abcd"
    }
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的移除 EC2 Instance Connect Endpoint。Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteInstanceConnectEndpoint。

delete-instance-event-window

以下程式碼範例顯示如何使用 delete-instance-event-window。

AWS CLI

範例 1：刪除事件時段

下列delete-instance-event-window範例會刪除事件時段。

```
aws ec2 delete-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890
```

輸出：

```
{
    "InstanceEventWindowState": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "State": "deleting"
    }
}
```

如需事件時段限制，請參閱《Amazon EC2 使用者指南》的排程事件一節中的考量事項。

範例 2：強制刪除事件時段

如果事件時段目前與目標相關聯，下列delete-instance-event-window範例強制力會刪除事件時段。

```
aws ec2 delete-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --force-delete
```

輸出：

```
{
    "InstanceEventWindowState": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "State": "deleting"
    }
}
```

如需事件時段限制，請參閱《Amazon EC2 使用者指南》的排程事件一節中的考量事項。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteInstanceEventWindow。

delete-internet-gateway

以下程式碼範例顯示如何使用 delete-internet-gateway。

AWS CLI

刪除網際網路閘道

下列delete-internet-gateway範例會刪除指定的網際網路閘道。

```
aws ec2 delete-internet-gateway \
    --internet-gateway-id igw-0d0fb496b3EXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的網際網路閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteInternetGateway。

delete-ipam-pool

以下程式碼範例顯示如何使用 delete-ipam-pool。

AWS CLI

刪除 IPAM 集區

在此範例中，您是 IPAM 委派管理員，想要刪除不再需要的 IPAM 集區，但集區已佈建 CIDR。除非您使用 --cascade選項，否則無法刪除已佈建 CIDRs的集區，因此您將使用 --cascade。

若要完成此請求：

您需要使用 describe-ipam-pools 取得的 IPAM 集區 ID。 --region 必須是 IPAM 主區域。

下列delete-ipam-pool範例會刪除您 AWS 帳戶中的 IPAM 集區。

```
aws ec2 delete-ipam-pool \
    --ipam-pool-id ipam-pool-050c886a3ca41cd5b \
    --cascade \
    --region us-east-1
```

輸出：

```
{
    "IpamPool": {
```

```
        "OwnerId": "320805250157",
        "IpamPoolId": "ipam-pool-050c886a3ca41cd5b",
        "IpamPoolArn": "arn:aws:ec2::320805250157:ipam-pool/ipam-
pool-050c886a3ca41cd5b",
        "IpamScopeArn": "arn:aws:ec2::320805250157:ipam-scope/ipam-
scope-0a158dde35c51107b",
        "IpamScopeType": "private",
        "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
        "IpamRegion": "us-east-1",
        "Locale": "None",
        "PoolDepth": 1,
        "State": "delete-in-progress",
        "Description": "example",
        "AutoImport": false,
        "AddressFamily": "ipv4",
        "AllocationMinNetmaskLength": 0,
        "AllocationMaxNetmaskLength": 32
    }
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的刪除集區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteIpamPool。

delete-ipam-resource-discovery

以下程式碼範例顯示如何使用 delete-ipam-resource-discovery。

AWS CLI

刪除資源探索

在此範例中，您是 IPAM 委派管理員，想要在整合 IPAM 與組織外部帳戶的過程中，刪除您建立以與其他 IPAM 管理員共用的非預設資源探索。

若要完成此請求：

--region 必須是您建立資源探索的區域。如果是 ，則無法刪除預設資源探索"IsDefault": true。預設資源探索是在建立 IPAM 的帳戶中自動建立的資源探索。若要刪除預設資源探索，您必須刪除 IPAM。

下列delete-ipam-resource-discovery範例會刪除資源探索。

```
aws ec2 delete-ipam-resource-discovery \
    --ipam-resource-discovery-id ipam-res-disco-0e39761475298ee0f \
    --region us-east-1
```

輸出：

```
{
    "IpamResourceDiscovery": {
        "OwnerId": "149977607591",
        "IpamResourceDiscoveryId": "ipam-res-disco-0e39761475298ee0f",
        "IpamResourceDiscoveryArn": "arn:aws:ec2::149977607591:ipam-resource-
discovery/ipam-res-disco-0e39761475298ee0f",
        "IpamResourceDiscoveryRegion": "us-east-1",
        "OperatingRegions": [
            {
                "RegionName": "us-east-1"
            }
        ],
        "IsDefault": false,
        "State": "delete-in-progress"
    }
}
```

如需資源探索的詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的使用資源探索。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteIpamResourceDiscovery。

delete-ipam-scope

以下程式碼範例顯示如何使用 delete-ipam-scope。

AWS CLI

刪除 IPAM 範圍

下列delete-ipam-scope範例會刪除 IPAM。

```
aws ec2 delete-ipam-scope \
    --ipam-scope-id ipam-scope-01c1ebab2b63bd7e4
```

輸出：

```
{
    "IpamScope": {
        "OwnerId": "123456789012",
        "IpamScopeId": "ipam-scope-01c1ebab2b63bd7e4",
        "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-01c1ebab2b63bd7e4",
        "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
        "IpamRegion": "us-east-1",
        "IpamScopeType": "private",
        "IsDefault": false,
        "Description": "Example description",
        "PoolCount": 0,
        "State": "delete-in-progress"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的刪除範圍。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteIpamScope。

delete-ipam

以下程式碼範例顯示如何使用 delete-ipam。

AWS CLI

刪除 IPAM

下列delete-ipam範例會刪除 IPAM。

```
aws ec2 delete-ipam \
    --ipam-id ipam-036486dfa6af58ee0
```

輸出：

```
{
    "Ipam": {
        "OwnerId": "123456789012",
        "IpamId": "ipam-036486dfa6af58ee0",
        "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-036486dfa6af58ee0",
        "IpamRegion": "us-east-1",
        "PublicDefaultScopeId": "ipam-scope-071b8042b0195c183",
```

```
        "PrivateDefaultScopeId": "ipam-scope-0807405dece705a30",
        "ScopeCount": 2,
        "OperatingRegions": [
            {
                "RegionName": "us-east-1"
            },
            {
                "RegionName": "us-east-2"
            },
            {
                "RegionName": "us-west-1"
            }
        ],
        "State": "delete-in-progress"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的刪除 IPAM。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteIpam。

delete-key-pair

以下程式碼範例顯示如何使用 delete-key-pair。

AWS CLI

刪除金鑰對

下列delete-key-pair範例會刪除指定的金鑰對。

```
aws ec2 delete-key-pair \
    --key-name my-key-pair
```

輸出：

```
{
    "Return": true,
    "KeyPairId": "key-03c8d3aceb53b507"
}
```

如需詳細資訊，請參閱《 AWS 命令列界面使用者指南》中的建立和刪除金鑰對。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteKeyPair。

delete-launch-template-versions

以下程式碼範例顯示如何使用 delete-launch-template-versions。

AWS CLI

刪除啟動範本版本

此範例會刪除指定的啟動範本版本。

命令：

```
aws ec2 delete-launch-template-versions --launch-template-id lt-0abcd290751193123 --versions 1
```

輸出：

```
{
  "UnsuccessfullyDeletedLaunchTemplateVersions": [],
  "SuccessfullyDeletedLaunchTemplateVersions": [
      {
          "LaunchTemplateName": "TestVersion",
          "VersionNumber": 1,
          "LaunchTemplateId": "lt-0abcd290751193123"
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLaunchTemplateVersions。

delete-launch-template

以下程式碼範例顯示如何使用 delete-launch-template。

AWS CLI

刪除啟動範本

此範例會刪除指定的啟動範本。

命令：

```
aws ec2 delete-launch-template --launch-template-id lt-0abcd290751193123
```

輸出：

```
{
  "LaunchTemplate": {
      "LatestVersionNumber": 2,
      "LaunchTemplateId": "lt-0abcd290751193123",
      "LaunchTemplateName": "TestTemplate",
      "DefaultVersionNumber": 2,
      "CreatedBy": "arn:aws:iam::123456789012:root",
      "CreateTime": "2017-11-23T16:46:25.000Z"
  }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteLaunchTemplate。

delete-local-gateway-route-table-virtual-interface-group-association

以下程式碼範例顯示如何使用 delete-local-gateway-route-table-virtual-interface-group-association。

AWS CLI

取消本機閘道路由表與虛擬介面 (VIFs) 群組的關聯

下列delete-local-gateway-route-table-virtual-interface-group-association範例會刪除指定本機閘道路由表與 VIF 群組之間的關聯。

```
aws ec2 delete-local-gateway-route-table-virtual-interface-group-association \
    --local-gateway-route-table-virtual-interface-group-association-id lgw-vif-grp-assoc-exampleid12345678
```

輸出：

```
{
    "LocalGatewayRouteTableVirtualInterfaceGroupAssociation": {
        "LocalGatewayRouteTableVirtualInterfaceGroupAssociationId": "lgw-vif-grp-assoc-exampleid12345678",
```

```
        "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-exampleid0123abcd",
        "LocalGatewayId": "lgw-exampleid11223344",
        "LocalGatewayRouteTableId": "lgw-rtb-exampleidabcd1234",
        "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-
gateway-route-table/lgw-rtb-exampleidabcd1234",
        "OwnerId": "111122223333",
        "State": "disassociating",
        "Tags": []
    }
}
```

如需詳細資訊，請參閱 AWS Outposts 使用者指南中的 VIF 群組關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLocalGatewayRouteTableVirtualInterfaceGroupAssociation。

delete-local-gateway-route-table-vpc-association

以下程式碼範例顯示如何使用 delete-local-gateway-route-table-vpc-association。

AWS CLI

取消本機閘道路由表與 VPC 的關聯

下列delete-local-gateway-route-table-vpc-association範例會刪除指定本機閘道路由表與 VPC 之間的關聯。

```
aws ec2 delete-local-gateway-route-table-vpc-association \
    --local-gateway-route-table-vpc-association-id vpc-example0123456789
```

輸出：

```
{
    "LocalGatewayRouteTableVpcAssociation": {
        "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-abcd1234wxyz56789",
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
        "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:555555555555:local-
gateway-route-table/lgw-rtb-abcdefg1234567890",
        "LocalGatewayId": "lgw-exampleid01234567",
        "VpcId": "vpc-example0123456789",
        "OwnerId": "555555555555",
        "State": "disassociating"
```

```
    }
}
```

如需詳細資訊，請參閱 AWS Outposts 使用者指南中的 VPC 關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLocalGatewayRouteTableVpcAssociation。

`delete-local-gateway-route-table`

以下程式碼範例顯示如何使用 `delete-local-gateway-route-table`。

AWS CLI

刪除本機閘道路由表

下列`delete-local-gateway-route-table`範例會使用直接 VPC 路由模式建立本機閘道路由表。

```
aws ec2 delete-local-gateway-route-table \
    --local-gateway-route-table-id lgw-rtb-abcdefg1234567890
```

輸出：

```
{
    "LocalGatewayRouteTable": {
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
        "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-
gateway-route-table/lgw-rtb-abcdefg1234567890",
        "LocalGatewayId": "lgw-1a2b3c4d5e6f7g8h9",
        "OutpostArn": "arn:aws:outposts:us-west-2:111122223333:outpost/
op-021345abcdef67890",
        "OwnerId": "111122223333",
        "State": "deleting",
        "Tags": [],
        "Mode": "direct-vpc-routing"
    }
}
```

如需詳細資訊，請參閱《AWS Outposts 使用者指南》中的本機閘道路油表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLocalGatewayRouteTable。

delete-local-gateway-route

以下程式碼範例顯示如何使用 delete-local-gateway-route。

AWS CLI

從本機閘道路由表刪除路由

下列delete-local-gateway-route範例會從指定的本機閘道路由表刪除指定的路由。

```
aws ec2 delete-local-gateway-route \
    --destination-cidr-block 0.0.0.0/0 \
    --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE
```

輸出：

```
{
    "Route": {
        "DestinationCidrBlock": "0.0.0.0/0",
        "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
        "Type": "static",
        "State": "deleted",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7EXAMPLE"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLocalGatewayRoute。

delete-managed-prefix-list

以下程式碼範例顯示如何使用 delete-managed-prefix-list。

AWS CLI

刪除字首清單

下列delete-managed-prefix-list範例會刪除指定的字首清單。

```
aws ec2 delete-managed-prefix-list \
    --prefix-list-id pl-0123456abcabcabc1
```

輸出：

```
{
    "PrefixList": {
        "PrefixListId": "pl-0123456abcabcabc1",
        "AddressFamily": "IPv4",
        "State": "delete-in-progress",
        "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-0123456abcabcabc1",
        "PrefixListName": "test",
        "MaxEntries": 10,
        "Version": 1,
        "OwnerId": "123456789012"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的受管字首清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteManagedPrefixList。

delete-nat-gateway

以下程式碼範例顯示如何使用 delete-nat-gateway。

AWS CLI

刪除 NAT 閘道

此範例會刪除 NAT 閘道 nat-04ae55e711cec5680。

命令：

```
aws ec2 delete-nat-gateway --nat-gateway-id nat-04ae55e711cec5680
```

輸出：

```
{
   "NatGatewayId": "nat-04ae55e711cec5680"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNatGateway。

delete-network-acl-entry

以下程式碼範例顯示如何使用 delete-network-acl-entry。

AWS CLI

刪除網路 ACL 項目

此範例會從指定的網路 ACL 刪除輸入規則編號 100。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 delete-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 100
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNetworkAclEntry。

delete-network-acl

以下程式碼範例顯示如何使用 delete-network-acl。

AWS CLI

刪除網路 ACL

此範例會刪除指定的網路 ACL。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 delete-network-acl --network-acl-id acl-5fb85d36
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNetworkAcl。

delete-network-insights-access-scope-analysis

以下程式碼範例顯示如何使用 delete-network-insights-access-scope-analysis。

AWS CLI

刪除網路存取範圍分析

下列delete-network-insights-access-scope-analysis範例會刪除指定的網路存取範圍分析。

```
aws ec2 delete-network-insights-access-scope-analysis \
    --network-insights-access-scope-analysis-id nisa-01234567891abcdef
```

輸出：

```
{
    "NetworkInsightsAccessScopeAnalysisId": "nisa-01234567891abcdef
}
```

如需詳細資訊，請參閱《Network Access Analyzer 指南》中的使用 AWS CLI 開始使用 Network Access Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNetworkInsightsAccessScopeAnalysis。

delete-network-insights-access-scope

以下程式碼範例顯示如何使用 delete-network-insights-access-scope。

AWS CLI

刪除網路存取範圍

下列delete-network-insights-access-scope範例會刪除指定的網路存取範圍。

```
aws ec2 delete-network-insights-access-scope \
    --network-insights-access-scope-id nis-123456789abc01234
```

輸出：

```
{
    "NetworkInsightsAccessScopeId": "nis-123456789abc01234"
}
```

如需詳細資訊，請參閱《Network Access Analyzer 指南》中的使用 AWS CLI 開始使用 Network Access Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNetworkInsightsAccessScope。

delete-network-insights-analysis

以下程式碼範例顯示如何使用 delete-network-insights-analysis。

AWS CLI

刪除路徑分析

下列delete-network-insights-analysis範例會刪除指定的分析。

```
aws ec2 delete-network-insights-analysis \
    --network-insights-analysis-id nia-02207aa13eb480c7a
```

輸出：

```
{
    "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a"
}
```

如需詳細資訊，請參閱 Reachability Analyzer 指南中的開始使用 AWS CLI。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNetworkInsightsAnalysis。

delete-network-insights-path

以下程式碼範例顯示如何使用 delete-network-insights-path。

AWS CLI

刪除路徑

下列delete-network-insights-path範例會刪除指定的路徑。您必須先使用 delete-network-insights-analysis命令刪除其所有分析，才能刪除路徑。

```
aws ec2 delete-network-insights-path \
    --network-insights-path-id nip-0b26f224f1d131fa8
```

輸出：

```
{
    "NetworkInsightsPathId": "nip-0b26f224f1d131fa8"
}
```

如需詳細資訊，請參閱 Reachability Analyzer 指南中的開始使用 AWS CLI。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNetworkInsightsPath。

delete-network-interface-permission

以下程式碼範例顯示如何使用 delete-network-interface-permission。

AWS CLI

刪除網路界面許可

此範例會刪除指定的網路介面許可。

命令：

```
aws ec2 delete-network-interface-permission --network-interface-permission-id eni-perm-06fd19020ede149ea
```

輸出：

```
{
  "Return": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNetworkInterfacePermission。

delete-network-interface

以下程式碼範例顯示如何使用 delete-network-interface。

AWS CLI

刪除網路界面

此範例會刪除指定的網路介面。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 delete-network-interface --network-interface-id eni-e5aa89a3
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNetworkInterface。

delete-placement-group

以下程式碼範例顯示如何使用 delete-placement-group。

AWS CLI

刪除置放群組

此範例命令會刪除指定的置放群組。

命令：

```
aws ec2 delete-placement-group --group-name my-cluster
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePlacementGroup。

delete-queued-reserved-instances

以下程式碼範例顯示如何使用 delete-queued-reserved-instances。

AWS CLI

刪除已排入佇列的購買

下列delete-queued-reserved-instances範例會刪除已排入佇列以供購買的指定預留執行個體。

```
aws ec2 delete-queued-reserved-instances \
    --reserved-instances-ids af9f760e-6f91-4559-85f7-4980eexample
```

輸出：

```
{
    "SuccessfulQueuedPurchaseDeletions": [
        {
            "ReservedInstancesId": "af9f760e-6f91-4559-85f7-4980eexample"
        }
    ],
    "FailedQueuedPurchaseDeletions": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteQueuedReservedInstances。

delete-route-table

以下程式碼範例顯示如何使用 delete-route-table。

AWS CLI

刪除路由表

此範例會刪除指定的路由表。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 delete-route-table --route-table-id rtb-22574640
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRouteTable。

delete-route

以下程式碼範例顯示如何使用 delete-route。

AWS CLI

刪除路由

此範例會從指定的路由表中刪除指定的路由。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 delete-route --route-table-id rtb-22574640 --destination-cidr-
block 0.0.0.0/0
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRoute。

delete-security-group

以下程式碼範例顯示如何使用 delete-security-group。

AWS CLI

[EC2-Classic] 刪除安全群組

此範例會刪除名為 MySecurityGroup 的安全群組。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 delete-security-group --group-name MySecurityGroup
```

[EC2-VPC] 刪除安全群組

此範例會刪除 ID 為 sg-903004f8 的安全群組。請注意，EC2-VPC 的安全群組不能按名稱引用。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 delete-security-group --group-id sg-903004f8
```

如需詳細資訊，請參閱《AWS 命令行介面使用者指南》中的「使用安全群組」。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteSecurityGroup。

delete-snapshot

以下程式碼範例顯示如何使用 delete-snapshot。

AWS CLI

刪除快照

此範例命令會刪除快照 ID 為 snap-1234567890abcdef0 的快照。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 delete-snapshot --snapshot-id snap-1234567890abcdef0
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteSnapshot。

delete-spot-datafeed-subscription

以下程式碼範例顯示如何使用 delete-spot-datafeed-subscription。

AWS CLI

取消 Spot 執行個體資料饋送訂閱

此範例命令會刪除帳戶的 Spot 資料饋送訂閱。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 delete-spot-datafeed-subscription
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSpotDatafeedSubscription。

delete-subnet-cidr-reservation

以下程式碼範例顯示如何使用 delete-subnet-cidr-reservation。

AWS CLI

刪除子網路 CIDR 保留

下列delete-subnet-cidr-reservation範例會刪除指定的子網路 CIDR 保留。

```
aws ec2 delete-subnet-cidr-reservation \
    --subnet-cidr-reservation-id scr-044f977c4eEXAMPLE
```

輸出：

```
{
    "DeletedSubnetCidrReservation": {
        "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
        "SubnetId": "subnet-03c51e2e6cEXAMPLE",
        "Cidr": "10.1.0.16/28",
        "ReservationType": "prefix",
        "OwnerId": "123456789012"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 子網 CIDR 保留。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSubnetCidrReservation。

delete-subnet

以下程式碼範例顯示如何使用 delete-subnet。

AWS CLI

刪除子網路

此範例會刪除指定的子網路。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 delete-subnet --subnet-id subnet-9d4a7b6c
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSubnet。

delete-tags

以下程式碼範例顯示如何使用 delete-tags。

AWS CLI

範例 1：從資源刪除標籤

下列delete-tags範例Stack=Test會從指定的映像中刪除標籤。當您同時指定值和金鑰名稱時，只有在標籤的值符合指定的值時，才會刪除標籤。

```
aws ec2 delete-tags \
    --resources ami-1234567890abcdef0 \
    --tags Key=Stack,Value=Test
```

您可以選擇指定標籤的值。下列delete-tags範例purpose會從指定的執行個體刪除具有金鑰名稱的標籤，無論標籤的標籤值為何。

```
aws ec2 delete-tags \
    --resources i-1234567890abcdef0 \
    --tags Key=purpose
```

如果您將空字串指定為標籤值，則只有在標籤的值為空字串時，才會刪除標籤。下列delete-tags範例指定空字串做為要刪除的標籤的標籤值。

```
aws ec2 delete-tags \
    --resources i-1234567890abcdef0 \
    --tags Key=Name,Value=
```

範例 2：從多個資源刪除標籤

下列delete-tags範例會從執行個體和 AMI 中刪除 tag``Purpose=Test``。如先前範例所示，您可以從 命令省略標籤值。

```
aws ec2 delete-tags \
    --resources i-1234567890abcdef0 ami-1234567890abcdef0 \
    --tags Key=Purpose
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTags。

delete-traffic-mirror-filter-rule

以下程式碼範例顯示如何使用 delete-traffic-mirror-filter-rule。

AWS CLI

刪除流量鏡像篩選條件規則

下列delete-traffic-mirror-filter-rule範例會刪除指定的流量鏡像篩選條件規則。

```
aws ec2 delete-traffic-mirror-filter-rule \
    --traffic-mirror-filter-rule-id tmfr-081f71283bEXAMPLE
```

輸出：

```
{
    "TrafficMirrorFilterRuleId": "tmfr-081f71283bEXAMPLE"
}
```

如需詳細資訊，請參閱《流量鏡射指南》中的修改流量鏡射篩選條件規則。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTrafficMirrorFilterRule。

delete-traffic-mirror-filter

以下程式碼範例顯示如何使用 delete-traffic-mirror-filter。

AWS CLI

刪除流量鏡像篩選條件

下列delete-traffic-mirror-filter範例會刪除指定的流量鏡像篩選條件。

```
aws ec2 delete-traffic-mirror-filter \
    --traffic-mirror-filter-id tmf-0be0b25fcdEXAMPLE
```

輸出：

```
{
    "TrafficMirrorFilterId": "tmf-0be0b25fcdEXAMPLE"
}
```

如需詳細資訊，請參閱《流量鏡射指南》中的刪除流量鏡射篩選條件。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTrafficMirrorFilter。

delete-traffic-mirror-session

以下程式碼範例顯示如何使用 delete-traffic-mirror-session。

AWS CLI

刪除流量鏡像工作階段

下列delete-traffic-mirror-session範例會刪除指定的流量鏡像工作階段。

```
aws ec2 delete-traffic-mirror-session \
    --traffic-mirror-session-id tms-0af3141ce5EXAMPLE
```

輸出：

```
{
    "TrafficMirrorSessionId": "tms-0af3141ce5EXAMPLE"
}
```

如需詳細資訊，請參閱《流量鏡射指南》中的刪除流量鏡射工作階段。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTrafficMirrorSession。

delete-traffic-mirror-target

以下程式碼範例顯示如何使用 delete-traffic-mirror-target。

AWS CLI

刪除流量鏡像目標

下列delete-traffic-mirror-target範例會刪除指定的流量鏡像目標。

```
aws ec2 delete-traffic-mirror-target \
    --traffic-mirror-target-id tmt-060f48ce9EXAMPLE
```

輸出：

```
{
    "TrafficMirrorTargetId": "tmt-060f48ce9EXAMPLE"
}
```

如需詳細資訊，請參閱《流量鏡射指南》中的刪除流量鏡射目標。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTrafficMirrorTarget。

delete-transit-gateway-connect-peer

以下程式碼範例顯示如何使用 delete-transit-gateway-connect-peer。

AWS CLI

刪除 Transit Gateway Connect 對等

下列delete-transit-gateway-connect-peer範例會刪除指定的 Connect 對等。

```
aws ec2 delete-transit-gateway-connect-peer \
    --transit-gateway-connect-peer-id tgw-connect-peer-0666adbac4EXAMPLE
```

輸出：

```
{
    "TransitGatewayConnectPeer": {
        "TransitGatewayAttachmentId": "tgw-attach-0f0927767cEXAMPLE",
        "TransitGatewayConnectPeerId": "tgw-connect-peer-0666adbac4EXAMPLE",
        "State": "deleting",
        "CreationTime": "2021-10-13T03:35:17.000Z",
```

```
        "ConnectPeerConfiguration": {
            "TransitGatewayAddress": "10.0.0.234",
            "PeerAddress": "172.31.1.11",
            "InsideCidrBlocks": [
                "169.254.6.0/29"
            ],
            "Protocol": "gre",
            "BgpConfigurations": [
                {
                    "TransitGatewayAsn": 64512,
                    "PeerAsn": 64512,
                    "TransitGatewayAddress": "169.254.6.2",
                    "PeerAddress": "169.254.6.1",
                    "BgpStatus": "down"
                },
                {
                    "TransitGatewayAsn": 64512,
                    "PeerAsn": 64512,
                    "TransitGatewayAddress": "169.254.6.3",
                    "PeerAddress": "169.254.6.1",
                    "BgpStatus": "down"
                }
            ]
        }
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的 Transit Gateway Connect 連接和 Transit Gateway Connect 對等。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTransitGatewayConnectPeer。

delete-transit-gateway-connect

以下程式碼範例顯示如何使用 delete-transit-gateway-connect。

AWS CLI

刪除傳輸閘道 Connect 連接

下列delete-transit-gateway-connect範例會刪除指定的 Connect 連接。

```
aws ec2 delete-transit-gateway-connect \
```

```
    --transit-gateway-attachment-id tgw-attach-037012e5dcEXAMPLE
```

輸出：

```
{
    "TransitGatewayConnect": {
        "TransitGatewayAttachmentId": "tgw-attach-037012e5dcEXAMPLE",
        "TransportTransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
        "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
        "State": "deleting",
        "CreationTime": "2021-03-09T19:59:17+00:00",
        "Options": {
            "Protocol": "gre"
        }
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的 Transit Gateway Connect 連接和 Transit Gateway Connect 對等。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTransitGatewayConnect。

delete-transit-gateway-multicast-domain

以下程式碼範例顯示如何使用 delete-transit-gateway-multicast-domain。

AWS CLI

刪除傳輸閘道多點傳送網域

下列delete-transit-gateway-multicast-domain範例會刪除指定的多點傳送網域。

```
aws ec2 delete-transit-gateway-multicast-domain \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

輸出：

```
{
    "TransitGatewayMulticastDomain": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-02bb79002bEXAMPLE",
        "TransitGatewayId": "tgw-0d88d2d0d5EXAMPLE",
```

```
        "State": "deleting",
        "CreationTime": "2019-11-20T22:02:03.000Z"
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的管理多點傳送網域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTransitGatewayMulticastDomain。

delete-transit-gateway-peering-attachment

以下程式碼範例顯示如何使用 delete-transit-gateway-peering-attachment。

AWS CLI

刪除傳輸閘道對等互連附件

下列delete-transit-gateway-peering-attachment範例會刪除指定的傳輸閘道對等互連附件。

```
aws ec2 delete-transit-gateway-peering-attachment \
    --transit-gateway-attachment-id tgw-attach-4455667788aabbccd
```

輸出：

```
{
    "TransitGatewayPeeringAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
        "RequesterTgwInfo": {
            "TransitGatewayId": "tgw-123abc05e04123abc",
            "OwnerId": "123456789012",
            "Region": "us-west-2"
        },
        "AccepterTgwInfo": {
            "TransitGatewayId": "tgw-11223344aabbcc112",
            "OwnerId": "123456789012",
            "Region": "us-east-2"
        },
        "State": "deleting",
        "CreationTime": "2019-12-09T11:38:31.000Z"
    }
```

```
}
```

如需詳細資訊，請參閱《傳輸閘道指南》中的傳輸閘道對等連接。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTransitGatewayPeeringAttachment。

delete-transit-gateway-policy-table

以下程式碼範例顯示如何使用 delete-transit-gateway-policy-table。

AWS CLI

刪除傳輸閘道政策表

下列delete-transit-gateway-policy-table範例會刪除指定的傳輸閘道政策表。

```
aws ec2 delete-transit-gateway-policy-table \
    --transit-gateway-policy-table-id tgw-ptb-0a16f134b78668a81
```

輸出：

```
{
    "TransitGatewayPolicyTables": [
        {
            "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
            "TransitGatewayId": "tgw-067f8505c18f0bd6e",
            "State": "deleting",
            "CreationTime": "2023-11-28T16:36:43+00:00",
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱《傳輸閘道使用者指南》中的傳輸閘道政策表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTransitGatewayPolicyTable。

delete-transit-gateway-prefix-list-reference

以下程式碼範例顯示如何使用 delete-transit-gateway-prefix-list-reference。

AWS CLI

刪除字首清單參考

下列delete-transit-gateway-prefix-list-reference範例會刪除指定的字首清單參考。

```
aws ec2 delete-transit-gateway-prefix-list-reference \
    --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
    --prefix-list-id pl-11111122222222333
```

輸出：

```
{
    "TransitGatewayPrefixListReference": {
        "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
        "PrefixListId": "pl-11111122222222333",
        "PrefixListOwnerId": "123456789012",
        "State": "deleting",
        "Blackhole": false,
        "TransitGatewayAttachment": {
            "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",
            "ResourceType": "vpc",
            "ResourceId": "vpc-112233445566aabbc"
        }
    }
}
```

如需詳細資訊，請參閱《傳輸閘道指南》中的字首清單參考。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTransitGatewayPrefixListReference。

delete-transit-gateway-route-table

以下程式碼範例顯示如何使用 delete-transit-gateway-route-table。

AWS CLI

刪除傳輸閘道路由表

下列delete-transit-gateway-route-table範例會刪除指定的傳輸閘道路由表。

```
aws ec2  delete-transit-gateway-route-table \
```

```
    --transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE
```

輸出：

```
{
    "TransitGatewayRouteTable": {
        "TransitGatewayRouteTableId": "tgw-rtb-0b6f6aaa01EXAMPLE",
        "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
        "State": "deleting",
        "DefaultAssociationRouteTable": false,
        "DefaultPropagationRouteTable": false,
        "CreationTime": "2019-07-17T20:27:26.000Z"
    }
}
```

如需詳細資訊，請參閱《傳輸閘道指南》中的刪除傳輸閘道路由表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTransitGatewayRouteTable。

delete-transit-gateway-route

以下程式碼範例顯示如何使用 delete-transit-gateway-route。

AWS CLI

從路由表刪除 CIDR 區塊

下列delete-transit-gateway-route範例會從指定的傳輸閘道路由表刪除 CIDR 區塊。

```
aws ec2 delete-transit-gateway-route \
    --transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE \
    --destination-cidr-block 10.0.2.0/24
```

輸出：

```
{
    "Route": {
        "DestinationCidrBlock": "10.0.2.0/24",
        "TransitGatewayAttachments": [
            {
                "ResourceId": "vpc-0065acced4EXAMPLE",
                "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
```

```
                "ResourceType": "vpc"
            }
        ],
        "Type": "static",
        "State": "deleted"
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的刪除靜態路由。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTransitGatewayRoute。

`delete-transit-gateway-vpc-attachment`

以下程式碼範例顯示如何使用 `delete-transit-gateway-vpc-attachment`。

AWS CLI

刪除傳輸閘道 VPC 連接

下列`delete-transit-gateway-vpc-attachment`範例會刪除指定的 VPC 連接。

```
aws ec2 delete-transit-gateway-vpc-attachment \
    --transit-gateway-attachment-id tgw-attach-0d2c54bdbEXAMPLE
```

輸出：

```
{
    "TransitGatewayVpcAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-0d2c54bdb3EXAMPLE",
        "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
        "VpcId": "vpc-0065acced4f61c651",
        "VpcOwnerId": "111122223333",
        "State": "deleting",
        "CreationTime": "2019-07-17T16:04:27.000Z"
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的刪除 VPC 連接。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTransitGatewayVpcAttachment。

delete-transit-gateway

以下程式碼範例顯示如何使用 delete-transit-gateway。

AWS CLI

刪除傳輸閘道

下列delete-transit-gateway範例會刪除指定的傳輸閘道。

```
aws ec2  delete-transit-gateway \
    --transit-gateway-id tgw-01f04542b2EXAMPLE
```

輸出：

```
{
    "TransitGateway": {
        "TransitGatewayId": "tgw-01f04542b2EXAMPLE",
        "State": "deleting",
        "OwnerId": "123456789012",
        "Description": "Example Transit Gateway",
        "CreationTime": "2019-08-27T15:04:35.000Z",
        "Options": {
            "AmazonSideAsn": 64515,
            "AutoAcceptSharedAttachments": "disable",
            "DefaultRouteTableAssociation": "enable",
            "AssociationDefaultRouteTableId": "tgw-rtb-0ce7a6948fEXAMPLE",
            "DefaultRouteTablePropagation": "enable",
            "PropagationDefaultRouteTableId": "tgw-rtb-0ce7a6948fEXAMPLE",
            "VpnEcmpSupport": "enable",
            "DnsSupport": "enable"
        }
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的刪除傳輸閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTransitGateway。

delete-verified-access-endpoint

以下程式碼範例顯示如何使用 delete-verified-access-endpoint。

AWS CLI

刪除已驗證存取端點

下列delete-verified-access-endpoint範例會刪除指定的 Verified Access 端點。

```
aws ec2 delete-verified-access-endpoint \
    --verified-access-endpoint-id vae-066fac616d4d546f2
```

輸出：

```
{
    "VerifiedAccessEndpoint": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
        "VerifiedAccessEndpointId": "vae-066fac616d4d546f2",
        "ApplicationDomain": "example.com",
        "EndpointType": "network-interface",
        "AttachmentType": "vpc",
        "DomainCertificateArn": "arn:aws:acm:us-east-2:123456789012:certificate/
eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE",
        "EndpointDomain": "my-ava-
app.edge-00c3372d53b1540bb.vai-0ce000c0b7643abea.prod.verified-access.us-
east-2.amazonaws.com",
        "SecurityGroupIds": [
            "sg-004915970c4c8f13a"
        ],
        "NetworkInterfaceOptions": {
            "NetworkInterfaceId": "eni-0aec70418c8d87a0f",
            "Protocol": "https",
            "Port": 443
        },
        "Status": {
            "Code": "deleting"
        },
        "Description": "Testing Verified Access",
        "CreationTime": "2023-08-25T20:54:43",
        "LastUpdatedTime": "2023-08-25T22:46:32"
    }
}
```

如需詳細資訊，請參閱《Verified Access 使用者指南》中的 Verified Access 端點。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVerifiedAccessEndpoint。

delete-verified-access-group

以下程式碼範例顯示如何使用 delete-verified-access-group。

AWS CLI

刪除 Verified Access 群組

下列delete-verified-access-group範例會刪除指定的 Verified Access 群組。

```
aws ec2 delete-verified-access-group \
    --verified-access-group-id vagr-0dbe967baf14b7235
```

輸出：

```
{
    "VerifiedAccessGroup": {
        "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "Description": "Testing Verified Access",
        "Owner": "123456789012",
        "VerifiedAccessGroupArn": "arn:aws:ec2:us-east-2:123456789012:verified-
access-group/vagr-0dbe967baf14b7235",
        "CreationTime": "2023-08-25T19:55:19",
        "LastUpdatedTime": "2023-08-25T22:49:03",
        "DeletionTime": "2023-08-26T00:58:31"
    }
}
```

如需詳細資訊，請參閱 Verified Access 使用者指南中的 Verified Access 群組。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVerifiedAccessGroup。

delete-verified-access-instance

以下程式碼範例顯示如何使用 delete-verified-access-instance。

AWS CLI

刪除 Verified Access 執行個體

下列delete-verified-access-instance範例會刪除指定的 Verified Access 執行個體。

```
aws ec2 delete-verified-access-instance \
    --verified-access-instance-id vai-0ce000c0b7643abea
```

輸出：

```
{
    "VerifiedAccessInstance": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "Description": "Testing Verified Access",
        "VerifiedAccessTrustProviders": [],
        "CreationTime": "2023-08-25T18:27:56",
        "LastUpdatedTime": "2023-08-26T01:00:18"
    }
}
```

如需詳細資訊，請參閱 Verified Access 使用者指南中的 Verified Access 執行個體。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVerifiedAccessInstance。

delete-verified-access-trust-provider

以下程式碼範例顯示如何使用 delete-verified-access-trust-provider。

AWS CLI

刪除 Verified Access 信任提供者

下列delete-verified-access-trust-provider範例會刪除指定的 Verified Access 信任提供者。

```
aws ec2 delete-verified-access-trust-provider \
    --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

輸出：

```
{
    "VerifiedAccessTrustProvider": {
        "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
        "Description": "Testing Verified Access",
        "TrustProviderType": "user",
```

```
        "UserTrustProviderType": "iam-identity-center",
        "PolicyReferenceName": "idc",
        "CreationTime": "2023-08-25T18:40:36",
        "LastUpdatedTime": "2023-08-25T18:40:36"
    }
}
```

如需詳細資訊，請參閱《Verified Access 使用者指南》中的 Verified Access 的信任提供者。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVerifiedAccessTrustProvider。

delete-volume

以下程式碼範例顯示如何使用 delete-volume。

AWS CLI

刪除磁碟區

此範例命令會刪除磁碟區 ID 為 的可用磁碟區vol-049df61146c4d7901。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 delete-volume --volume-id vol-049df61146c4d7901
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVolume。

delete-vpc-endpoint-connection-notifications

以下程式碼範例顯示如何使用 delete-vpc-endpoint-connection-notifications。

AWS CLI

刪除端點連線通知

此範例會刪除指定的端點連線通知。

命令：

```
aws ec2 delete-vpc-endpoint-connection-notifications --connection-notification-
ids vpce-nfn-008776de7e03f5abc
```

輸出：

```
{
  "Unsuccessful": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVpcEndpointConnectionNotifications。

delete-vpc-endpoint-service-configurations

以下程式碼範例顯示如何使用 delete-vpc-endpoint-service-configurations。

AWS CLI

刪除端點服務組態

此範例會刪除指定的端點服務組態。

命令：

```
aws ec2 delete-vpc-endpoint-service-configurations --service-ids vpce-
svc-03d5ebb7d9579a2b3
```

輸出：

```
{
   "Unsuccessful": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVpcEndpointServiceConfigurations。

delete-vpc-endpoints

以下程式碼範例顯示如何使用 delete-vpc-endpoints。

AWS CLI

刪除端點

此範例會刪除端點 vpce-aa22bb33 和 vpce-1a2b3c4d。如果命令部分成功或失敗，則會傳回失敗項目的清單。如果命令成功，傳回的清單會是空的。

命令：

```
aws ec2 delete-vpc-endpoints --vpc-endpoint-ids vpce-aa22bb33 vpce-1a2b3c4d
```

輸出：

```
{
  "Unsuccessful": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVpcEndpoints。

`delete-vpc-peering-connection`

以下程式碼範例顯示如何使用 `delete-vpc-peering-connection`。

AWS CLI

刪除 VPC 對等互連

此範例會刪除指定的 VPC 互連連線。

命令：

```
aws ec2 delete-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

輸出：

```
{
    "Return": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVpcPeeringConnection。

`delete-vpc`

以下程式碼範例顯示如何使用 `delete-vpc`。

AWS CLI

刪除 VPC

此範例會刪除指定的 VPC。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 delete-vpc --vpc-id vpc-a01106c2
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVpc。

delete-vpn-connection-route

以下程式碼範例顯示如何使用 delete-vpn-connection-route。

AWS CLI

從 VPN 連接刪除靜態路由

此範例會從指定的 VPN 連接刪除指定的靜態路由。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 delete-vpn-connection-route --vpn-connection-id vpn-40f41529 --destination-
cidr-block 11.12.0.0/16
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVpnConnectionRoute。

delete-vpn-connection

以下程式碼範例顯示如何使用 delete-vpn-connection。

AWS CLI

刪除 VPN 連線

此範例會刪除指定的 VPN 連線。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 delete-vpn-connection --vpn-connection-id vpn-40f41529
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVpnConnection。

delete-vpn-gateway

以下程式碼範例顯示如何使用 delete-vpn-gateway。

AWS CLI

刪除虛擬私有閘道

此範例會刪除指定的虛擬私有閘道。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 delete-vpn-gateway --vpn-gateway-id vgw-9a4cacf3
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVpnGateway。

deprovision-byoip-cidr

以下程式碼範例顯示如何使用 deprovision-byoip-cidr。

AWS CLI

移除 IP 地址範圍而不使用

下列範例會將指定的地址範圍從 中移除 AWS。

```
aws ec2 deprovision-byoip-cidr \
    --cidr 203.0.113.25/24
```

輸出：

```
{
    "ByoipCidr": {
        "Cidr": "203.0.113.25/24",
        "State": "pending-deprovision"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeprovisionByoipCidr。

deprovision-ipam-pool-cidr

以下程式碼範例顯示如何使用 deprovision-ipam-pool-cidr。

AWS CLI

取消佈建 IPAM 集區 CIDR

下列deprovision-ipam-pool-cidr範例會取消佈建佈建至 IPAM 集區的 CIDR。

(Linux)：

```
aws ec2 deprovision-ipam-pool-cidr \
    --ipam-pool-id ipam-pool-02ec043a19bbe5d08 \
    --cidr 11.0.0.0/16
```

(Windows)：

```
aws ec2 deprovision-ipam-pool-cidr ^
    --ipam-pool-id ipam-pool-02ec043a19bbe5d08 ^
    --cidr 11.0.0.0/16
```

輸出：

```
{
    "IpamPoolCidr": {
        "Cidr": "11.0.0.0/16",
        "State": "pending-deprovision"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的取消佈建集區 CIDRs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeprovisionIpamPoolCidr。

deregister-image

以下程式碼範例顯示如何使用 deregister-image。

AWS CLI

取消註冊 AMI

此範例會取消註冊指定的 AMI。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 deregister-image --image-id ami-4fa54026
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterImage。

deregister-instance-event-notification-attributes

以下程式碼範例顯示如何使用 deregister-instance-event-notification-attributes。

AWS CLI

範例 1：從事件通知中移除所有標籤

下列deregister-instance-event-notification-attributes範例會移除 IncludeAllTagsOfInstance=true，其具有IncludeAllTagsOfInstance將 設定為 的效果false。

```
aws ec2 deregister-instance-event-notification-attributes \
    --instance-tag-attribute IncludeAllTagsOfInstance=true
```

輸出：

```
{
    "InstanceTagAttribute": {
        "InstanceTagKeys": [],
        "IncludeAllTagsOfInstance": true
    }
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud Linux 執行個體使用者指南》中的執行個體的排程事件。

範例 2：從事件通知中移除特定標籤

下列deregister-instance-event-notification-attributes範例會從事件通知中包含的標籤移除指定的標籤。若要描述事件通知中包含的其餘標籤，請使用 describe-instance-event-notification-attributes。

```
aws ec2 deregister-instance-event-notification-attributes \
    --instance-tag-attribute InstanceTagKeys="tag-key2"
```

輸出：

```
{
    "InstanceTagAttribute": {
        "InstanceTagKeys": [
            "tag-key2"
        ],
        "IncludeAllTagsOfInstance": false
    }
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud Linux 執行個體使用者指南》中的執行個體的排程事件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterInstanceEventNotificationAttributes。

deregister-transit-gateway-multicast-group-members

以下程式碼範例顯示如何使用 deregister-transit-gateway-multicast-group-members。

AWS CLI

從多點傳送群組取消註冊群組成員

此範例會從傳輸閘道多點傳送群組取消註冊指定的網路介面群組成員。

```
aws ec2 deregister-transit-gateway-multicast-group-members \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE \
    --group-ip-address 224.0.1.0 \
    --network-interface-ids eni-0e246d3269EXAMPLE
```

輸出：

```
{
    "DeregisteredMulticastGroupMembers": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef7EXAMPLE",
        "RegisteredNetworkInterfaceIds": [
```

```
            "eni-0e246d3269EXAMPLE"
        ],
        "GroupIpAddress": "224.0.1.0"
    }
}
```

如需詳細資訊，請參閱 AWS Transit Gateways 使用者指南中的從多點傳送群組取消註冊成員。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterTransitGatewayMulticastGroupMembers。

deregister-transit-gateway-multicast-group-source

以下程式碼範例顯示如何使用 deregister-transit-gateway-multicast-group-source。

AWS CLI

從傳輸閘道多點傳送群組取消註冊來源

此範例會從多點傳送群組取消註冊指定的網路介面群組來源。

```
aws ec2 register-transit-gateway-multicast-group-sources \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \
    --group-ip-address 224.0.1.0 \
    --network-interface-ids eni-07f290fc3c090cbae
```

輸出：

```
{
    "DeregisteredMulticastGroupSources": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
        "DeregisteredNetworkInterfaceIds": [
            "eni-07f290fc3c090cbae"
        ],
        "GroupIpAddress": "224.0.1.0"
    }
}
```

如需詳細資訊，請參閱 AWS Transit Gateways 使用者指南中的從多點傳送群組取消註冊來源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterTransitGatewayMulticastGroupSource。

describe-account-attributes

以下程式碼範例顯示如何使用 describe-account-attributes。

AWS CLI

描述您 AWS 帳戶的所有屬性

此範例說明您 AWS 帳戶的屬性。

命令：

```
aws ec2 describe-account-attributes
```

輸出：

```
{
    "AccountAttributes": [
        {
            "AttributeName": "vpc-max-security-groups-per-interface",
            "AttributeValues": [
                {
                    "AttributeValue": "5"
                }
            ]
        },
        {
            "AttributeName": "max-instances",
            "AttributeValues": [
                {
                    "AttributeValue": "20"
                }
            ]
        },
        {
            "AttributeName": "supported-platforms",
            "AttributeValues": [
                {
                    "AttributeValue": "EC2"
                },
                {
                    "AttributeValue": "VPC"
                }
```

```
            ]
        },
        {
            "AttributeName": "default-vpc",
            "AttributeValues": [
                {
                    "AttributeValue": "none"
                }
            ]
        },
        {
            "AttributeName": "max-elastic-ips",
            "AttributeValues": [
                {
                    "AttributeValue": "5"
                }
            ]
        },
        {
            "AttributeName": "vpc-max-elastic-ips",
            "AttributeValues": [
                {
                    "AttributeValue": "5"
                }
            ]
        }
    ]
}
```

描述您 AWS 帳戶的單一屬性

此範例說明您 AWS 帳戶的 supported-platforms 屬性。

命令：

```
aws ec2 describe-account-attributes --attribute-names supported-platforms
```

輸出：

```
{
    "AccountAttributes": [
        {
            "AttributeName": "supported-platforms",
```

```
            "AttributeValues": [
                {
                    "AttributeValue": "EC2"
                },
                {
                    "AttributeValue": "VPC"
                }
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAccountAttributes。

`describe-address-transfers`

以下程式碼範例顯示如何使用 `describe-address-transfers`。

AWS CLI

描述彈性 IP 地址傳輸

下列`describe-address-transfers`範例說明指定彈性 IP 地址的彈性 IP 地址轉移。

```
aws ec2 describe-address-transfers \
    --allocation-ids eipalloc-09ad461b0d03f6aaf
```

輸出：

```
{
    "AddressTransfers": [
        {
            "PublicIp": "100.21.184.216",
            "AllocationId": "eipalloc-09ad461b0d03f6aaf",
            "TransferAccountId": "123456789012",
            "TransferOfferExpirationTimestamp": "2023-02-22T22:51:01.000Z",
            "AddressTransferStatus": "pending"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的轉移彈性 IP 地址。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAddressTransfers。

`describe-addresses-attribute`

以下程式碼範例顯示如何使用 `describe-addresses-attribute`。

AWS CLI

檢視與彈性 IP 地址相關聯的網域名稱屬性

下列`describe-addresses-attribute`範例會傳回與彈性 IP 地址相關聯的網域名稱屬性。

Linux：

```
aws ec2 describe-addresses-attribute \
    --allocation-ids eipalloc-abcdef01234567890 \
    --attribute domain-name
```

Windows：

```
aws ec2 describe-addresses-attribute ^
    --allocation-ids eipalloc-abcdef01234567890 ^
    --attribute domain-name
```

輸出：

```
{
    "Addresses": [
        {
            "PublicIp": "192.0.2.0",
            "AllocationId": "eipalloc-abcdef01234567890",
            "PtrRecord": "example.com."
        }
    ]
}
```

若要檢視彈性 IP 地址的屬性，您必須先將網域名稱與彈性 IP 地址建立關聯。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的為電子郵件應用程式使用反向 DNS，或《AWS CLI 命令參考》中的 modify-address-attribute。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAddressesAttribute。

describe-addresses

以下程式碼範例顯示如何使用 describe-addresses。

AWS CLI

範例 1：擷取有關您所有彈性 IP 地址的詳細資訊

以下 describe addresses 範例顯示有關您彈性 IP 地址的詳細資訊。

```
aws ec2 describe-addresses
```

輸出：

```
{
    "Addresses": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "PublicIp": "198.51.100.0",
            "PublicIpv4Pool": "amazon",
            "Domain": "standard"
        },
        {
            "Domain": "vpc",
            "PublicIpv4Pool": "amazon",
            "InstanceId": "i-1234567890abcdef0",
            "NetworkInterfaceId": "eni-12345678",
            "AssociationId": "eipassoc-12345678",
            "NetworkInterfaceOwnerId": "123456789012",
            "PublicIp": "203.0.113.0",
            "AllocationId": "eipalloc-12345678",
            "PrivateIpAddress": "10.0.1.241"
        }
    ]
}
```

範例 2：擷取有關 EC2-VPC 適用之彈性 IP 地址的詳細資訊

下列 describe-addresses 範例會顯示彈性 IP 地址的詳細資訊，以便搭配 VPC 中的執行個體使用。

```
aws ec2 describe-addresses \
```

```
    --filters "Name=domain,Values=vpc"
```

輸出：

```
{
    "Addresses": [
        {
            "Domain": "vpc",
            "PublicIpv4Pool": "amazon",
            "InstanceId": "i-1234567890abcdef0",
            "NetworkInterfaceId": "eni-12345678",
            "AssociationId": "eipassoc-12345678",
            "NetworkInterfaceOwnerId": "123456789012",
            "PublicIp": "203.0.113.0",
            "AllocationId": "eipalloc-12345678",
            "PrivateIpAddress": "10.0.1.241"
        }
    ]
}
```

範例 3：擷取有關透過配置 ID 所指定的彈性 IP 地址的詳細資訊

下列 `describe-addresses` 範例顯示具有指定配置 ID (已與 EC2-VPC 中的執行個體建立關聯) 的彈性 IP 地址的詳細資訊。

```
aws ec2 describe-addresses \
    --allocation-ids eipalloc-282d9641
```

輸出：

```
{
    "Addresses": [
        {
            "Domain": "vpc",
            "PublicIpv4Pool": "amazon",
            "InstanceId": "i-1234567890abcdef0",
            "NetworkInterfaceId": "eni-1a2b3c4d",
            "AssociationId": "eipassoc-123abc12",
            "NetworkInterfaceOwnerId": "1234567891012",
            "PublicIp": "203.0.113.25",
            "AllocationId": "eipalloc-282d9641",
            "PrivateIpAddress": "10.251.50.12"
```

```
        }
    ]
}
```

範例 4：擷取有關透過其 VPC 私有 IP 地址所指定的彈性 IP 地址的詳細資訊

下列 `describe-addresses` 範例針對已與 EC2-VPC 中特定私有 IP 位址建立關聯的彈性 IP 地址，顯示詳細資訊。

```
aws ec2 describe-addresses \
    --filters "Name=private-ip-address,Values=10.251.50.12"
```

範例 5：擷取有關 EC2-Classic 中彈性 IP 地址的詳細資訊

下列 `describe-addresses` 範例會顯示彈性 IP 地址的詳細資訊，以便用於 EC2-Classic。

```
aws ec2 describe-addresses \
    --filters "Name=domain,Values=standard"
```

輸出：

```
{
    "Addresses": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "PublicIp": "203.0.110.25",
            "PublicIpv4Pool": "amazon",
            "Domain": "standard"
        }
    ]
}
```

範例 6：擷取有關透過其公有 IP 地址所指定的彈性 IP 地址的詳細資訊

下列 `describe-addresses` 範例顯示具有值 `203.0.110.25` (已與 EC2-Classic 中的執行個體建立關聯) 的彈性 IP 地址的詳細資訊。

```
aws ec2 describe-addresses \
    --public-ips 203.0.110.25
```

輸出：

```
{
    "Addresses": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "PublicIp": "203.0.110.25",
            "PublicIpv4Pool": "amazon",
            "Domain": "standard"
        }
    ]
}
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeAddresses。

describe-aggregate-id-format

以下程式碼範例顯示如何使用 describe-aggregate-id-format。

AWS CLI

描述區域中所有資源類型的較長 ID 格式設定

下列describe-aggregate-id-format範例說明目前區域的整體長 ID 格式狀態。Deadline 值表示這些資源從短 ID 格式永久切換為長 ID 格式的截止日期已過期。UseLongIdsAggregated 值表示所有 IAM 使用者和 IAM 角色都設定為對所有資源類型使用長 ID 格式。

```
aws ec2 describe-aggregate-id-format
```

輸出：

```
{
    "UseLongIdsAggregated": true,
    "Statuses": [
        {
            "Deadline": "2018-08-13T02:00:00.000Z",
            "Resource": "network-interface-attachment",
            "UseLongIds": true
        },
        {
            "Deadline": "2016-12-13T02:00:00.000Z",
```

```
            "Resource": "instance",
            "UseLongIds": true
        },
        {
            "Deadline": "2018-08-13T02:00:00.000Z",
            "Resource": "elastic-ip-association",
            "UseLongIds": true
        },
        ...
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAggregateIdFormat。

describe-availability-zones

以下程式碼範例顯示如何使用 describe-availability-zones。

AWS CLI

描述您的可用區域

下列範例 describe-availability-zones 針對可供您使用的可用區域顯示詳細資訊。回應包含僅適用於目前區域的可用區域。在這個範例中，它預設在 us-west-2 (奧勒岡) 區域使用設定檔。

```
aws ec2 describe-availability-zones
```

輸出：

```
{
    "AvailabilityZones": [
        {
            "State": "available",
            "OptInStatus": "opt-in-not-required",
            "Messages": [],
            "RegionName": "us-west-2",
            "ZoneName": "us-west-2a",
            "ZoneId": "usw2-az1",
            "GroupName": "us-west-2",
            "NetworkBorderGroup": "us-west-2"
        },
```

```
        {
            "State": "available",
            "OptInStatus": "opt-in-not-required",
            "Messages": [],
            "RegionName": "us-west-2",
            "ZoneName": "us-west-2b",
            "ZoneId": "usw2-az2",
            "GroupName": "us-west-2",
            "NetworkBorderGroup": "us-west-2"
        },
        {
            "State": "available",
            "OptInStatus": "opt-in-not-required",
            "Messages": [],
            "RegionName": "us-west-2",
            "ZoneName": "us-west-2c",
            "ZoneId": "usw2-az3",
            "GroupName": "us-west-2",
            "NetworkBorderGroup": "us-west-2"
        },
        {
            "State": "available",
            "OptInStatus": "opt-in-not-required",
            "Messages": [],
            "RegionName": "us-west-2",
            "ZoneName": "us-west-2d",
            "ZoneId": "usw2-az4",
            "GroupName": "us-west-2",
            "NetworkBorderGroup": "us-west-2"
        },
        {
            "State": "available",
            "OptInStatus": "opted-in",
            "Messages": [],
            "RegionName": "us-west-2",
            "ZoneName": "us-west-2-lax-1a",
            "ZoneId": "usw2-lax1-az1",
            "GroupName": "us-west-2-lax-1",
            "NetworkBorderGroup": "us-west-2-lax-1"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeAvailabilityZones。

describe-aws-network-performance-metric-subscription

以下程式碼範例顯示如何使用 describe-aws-network-performance-metric-subscription。

AWS CLI

描述您的指標訂閱

下列describe-aws-network-performance-metric-subscriptions範例說明您的指標訂閱。

```
aws ec2 describe-aws-network-performance-metric-subscriptions
```

輸出：

```
{
    "Subscriptions": [
        {
            "Source": "us-east-1",
            "Destination": "eu-west-1",
            "Metric": "aggregate-latency",
            "Statistic": "p50",
            "Period": "five-minutes"
        }
    ]
}
```

如需詳細資訊，請參閱《 基礎設施效能使用者指南》中的管理訂閱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAwsNetworkPerformanceMetricSubscription。

describe-aws-network-performance-metric-subscriptions

以下程式碼範例顯示如何使用 describe-aws-network-performance-metric-subscriptions。

AWS CLI

描述您的指標訂閱

下列describe-aws-network-performance-metric-subscriptions範例說明您的指標訂閱。

```
aws ec2 describe-aws-network-performance-metric-subscriptions
```

輸出：

```
{
    "Subscriptions": [
        {
            "Source": "us-east-1",
            "Destination": "eu-west-1",
            "Metric": "aggregate-latency",
            "Statistic": "p50",
            "Period": "five-minutes"
        }
    ]
}
```

如需詳細資訊，請參閱《 基礎設施效能使用者指南》中的管理訂閱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAwsNetworkPerformanceMetricSubscriptions。

describe-bundle-tasks

以下程式碼範例顯示如何使用 describe-bundle-tasks。

AWS CLI

描述您的套件任務

此範例說明所有套件任務。

命令：

```
aws ec2 describe-bundle-tasks
```

輸出：

```
{
  "BundleTasks": [
```

```
    {
      "UpdateTime": "2015-09-15T13:26:54.000Z",
      "InstanceId": "i-1234567890abcdef0",
      "Storage": {
        "S3": {
            "Prefix": "winami",
            "Bucket": "bundletasks"
        }
      },
      "State": "bundling",
      "StartTime": "2015-09-15T13:24:35.000Z",
      "Progress": "3%",
      "BundleId": "bun-2a4e041c"
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeBundleTasks。

describe-byoip-cidrs

以下程式碼範例顯示如何使用 describe-byoip-cidrs。

AWS CLI

描述您的佈建地址範圍

下列describe-byoip-cidrs範例顯示您佈建供 使用的公有 IPv4 地址範圍詳細資訊 AWS。

```
aws ec2 describe-byoip-cidrs
```

輸出：

```
{
    "ByoipCidrs": [
        {
            "Cidr": "203.0.113.25/24",
            "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
            "State": "provisioned"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeByoipCidrs。

describe-capacity-reservation-fleets

以下程式碼範例顯示如何使用 describe-capacity-reservation-fleets。

AWS CLI

檢視容量保留機群

下列describe-capacity-reservation-fleets範例列出指定容量保留機群的組態和容量資訊。它也會列出機群中個別容量預留的詳細資訊。

```
aws ec2 describe-capacity-reservation-fleets \
    --capacity-reservation-fleet-ids crf-abcdef01234567890
```

輸出：

```
{
    "CapacityReservationFleets": [
        {
            "State": "active",
            "EndDate": "2022-12-31T23:59:59.000Z",
            "InstanceMatchCriteria": "open",
            "Tags": [],
            "CapacityReservationFleetId": "crf-abcdef01234567890",
            "Tenancy": "default",
            "InstanceTypeSpecifications": [
                {
                    "CapacityReservationId": "cr-1234567890abcdef0",
                    "AvailabilityZone": "us-east-1a",
                    "FulfilledCapacity": 5.0,
                    "Weight": 1.0,
                    "CreateDate": "2022-07-02T08:34:33.398Z",
                    "InstancePlatform": "Linux/UNIX",
                    "TotalInstanceCount": 5,
                    "Priority": 1,
                    "EbsOptimized": true,
                    "InstanceType": "m5.xlarge"
                }
            ],
            "TotalTargetCapacity": 5,
```

```
            "TotalFulfilledCapacity": 5.0,
            "CreateTime": "2022-07-02T08:34:33.397Z",
            "AllocationStrategy": "prioritized"
        }
    ]
}
```

如需容量保留機群的詳細資訊，請參閱《Amazon EC2 使用者指南》中的容量保留機群。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCapacityReservationFleets。

describe-capacity-reservations

以下程式碼範例顯示如何使用 describe-capacity-reservations。

AWS CLI

範例 1：描述一或多個容量保留

下列describe-capacity-reservations範例顯示目前 AWS 區域中所有容量保留的詳細資訊。

```
aws ec2 describe-capacity-reservations
```

輸出：

```
{
    "CapacityReservations": [
        {
            "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
            "OwnerId": "123456789111",
            "CapacityReservationArn": "arn:aws:ec2:us-east-1:123456789111:capacity-
reservation/cr-1234abcd56EXAMPLE",
            "AvailabilityZoneId": "use1-az2",
            "InstanceType": "c5.large",
            "InstancePlatform": "Linux/UNIX",
            "AvailabilityZone": "us-east-1a",
            "Tenancy": "default",
            "TotalInstanceCount": 1,
            "AvailableInstanceCount": 1,
            "EbsOptimized": true,
            "EphemeralStorage": false,
            "State": "active",
```

```
            "StartDate": "2024-10-23T15:00:24+00:00",
            "EndDateType": "unlimited",
            "InstanceMatchCriteria": "open",
            "CreateDate": "2024-10-23T15:00:24+00:00",
            "Tags": [],
            "CapacityAllocations": []
        },
        {
            "CapacityReservationId": "cr-abcdEXAMPLE9876ef ",
            "OwnerId": "123456789111",
            "CapacityReservationArn": "arn:aws:ec2:us-east-1:123456789111:capacity-
reservation/cr-abcdEXAMPLE9876ef",
            "AvailabilityZoneId": "use1-az2",
            "InstanceType": "c4.large",
            "InstancePlatform": "Linux/UNIX",
            "AvailabilityZone": "us-east-1a",
            "Tenancy": "default",
            "TotalInstanceCount": 1,
            "AvailableInstanceCount": 1,
            "EbsOptimized": true,
            "EphemeralStorage": false,
            "State": "cancelled",
            "StartDate": "2024-10-23T15:01:03+00:00",
            "EndDateType": "unlimited",
            "InstanceMatchCriteria": "open",
            "CreateDate": "2024-10-23T15:01:02+00:00",
            "Tags": [],
            "CapacityAllocations": []
        }
    ]
}
```

範例 2：描述一或多個容量保留

下列describe-capacity-reservations範例顯示指定容量保留的詳細資訊。

```
aws ec2 describe-capacity-reservations \
    --capacity-reservation-ids cr-1234abcd56EXAMPLE
```

輸出：

```
{
    "CapacityReservations": [
```

```
        {
            "CapacityReservationId": "cr-abcdEXAMPLE9876ef ",
            "OwnerId": "123456789111",
            "CapacityReservationArn": "arn:aws:ec2:us-east-1:123456789111:capacity-
reservation/cr-abcdEXAMPLE9876ef",
            "AvailabilityZoneId": "use1-az2",
            "InstanceType": "c4.large",
            "InstancePlatform": "Linux/UNIX",
            "AvailabilityZone": "us-east-1a",
            "Tenancy": "default",
            "TotalInstanceCount": 1,
            "AvailableInstanceCount": 1,
            "EbsOptimized": true,
            "EphemeralStorage": false,
            "State": "active",
            "StartDate": "2024-10-23T15:01:03+00:00",
            "EndDateType": "unlimited",
            "InstanceMatchCriteria": "open",
            "CreateDate": "2024-10-23T15:01:02+00:00",
            "Tags": [],
            "CapacityAllocations": []
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud Linux 執行個體使用者指南》中的檢視容量保留。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCapacityReservations。

describe-carrier-gateways

以下程式碼範例顯示如何使用 `describe-carrier-gateways`。

AWS CLI

描述所有電信業者閘道

下列`describe-carrier-gateways`範例列出所有電信業者閘道。

```
aws ec2 describe-carrier-gateways
```

輸出：

```
{
    "CarrierGateways": [
        {
            "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
            "VpcId": "vpc-0c529aEXAMPLE",
            "State": "available",
            "OwnerId": "123456789012",
            "Tags": [
                {

                    "Key": "example",
                    "Value": "tag"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的電信業者閘道<https://docs.aws.amazon.com/vpc/latest/userguide/Carrier_Gateway.html>。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCarrierGateways。

describe-classic-link-instances

以下程式碼範例顯示如何使用 `describe-classic-link-instances`。

AWS CLI

描述連結的 EC2-Classic 執行個體

此範例列出所有連結的 EC2-Classic 執行個體。

命令：

```
aws ec2 describe-classic-link-instances
```

輸出：

```
{
        "Instances": [
```

```
                {
                        "InstanceId": "i-1234567890abcdef0",
                        "VpcId": "vpc-88888888",
                        "Groups": [
                                {
                                        "GroupId": "sg-11122233"
                                }
                        ],
                        "Tags": [
                                {
                                        "Value": "ClassicInstance",
                                        "Key": "Name"
                                }
                        ]
                },
                {
                        "InstanceId": "i-0598c7d356eba48d7",
                        "VpcId": "vpc-12312312",
                        "Groups": [
                                {
                                        "GroupId": "sg-aabbccdd"
                                }
                        ],
                        "Tags": [
                                {
                                        "Value": "ClassicInstance2",
                                        "Key": "Name"
                                }
                        ]
                }
        ]
}
```

此範例列出所有連結的 EC2-Classic 執行個體，並篩選回應以僅包含連結至 VPC vpc-88888888 的執行個體。

命令：

```
aws ec2 describe-classic-link-instances --filter "Name=vpc-id,Values=vpc-88888888"
```

輸出：

```
{
```

```
        "Instances": [
                {
                        "InstanceId": "i-1234567890abcdef0",
                        "VpcId": "vpc-88888888",
                        "Groups": [
                                {
                                        "GroupId": "sg-11122233"
                                }
                        ],
                        "Tags": [
                                {
                                        "Value": "ClassicInstance",
                                        "Key": "Name"
                                }
                        ]
                }
        ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClassicLinkInstances。

describe-client-vpn-authorization-rules

以下程式碼範例顯示如何使用 describe-client-vpn-authorization-rules。

AWS CLI

描述 Client VPN 端點的授權規則

下列describe-client-vpn-authorization-rules範例顯示指定 Client VPN 端點的授權規則詳細資訊。

```
aws ec2 describe-client-vpn-authorization-rules \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

輸出：

```
{
    "AuthorizationRules": [
        {
            "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
```

```
            "GroupId": "",
            "AccessAll": true,
            "DestinationCidr": "0.0.0.0/0",
            "Status": {
                "Code": "active"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Client VPN 管理員指南中的授權規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClientVpnAuthorizationRules。

describe-client-vpn-connections

以下程式碼範例顯示如何使用 describe-client-vpn-connections。

AWS CLI

描述 Client VPN 端點的連線

下列describe-client-vpn-connections範例顯示用戶端連線至指定 Client VPN 端點的詳細資訊。

```
aws ec2 describe-client-vpn-connections \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

輸出：

```
{
    "Connections": [
        {
            "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
            "Timestamp": "2019-08-12 07:58:34",
            "ConnectionId": "cvpn-connection-0e03eb24267165acd",
            "ConnectionEstablishedTime": "2019-08-12 07:57:14",
            "IngressBytes": "32302",
            "EgressBytes": "5696",
            "IngressPackets": "332",
```

```
            "EgressPackets": "67",
            "ClientIp": "172.31.0.225",
            "CommonName": "client1.domain.tld",
            "Status": {
                "Code": "terminated"
            },
            "ConnectionEndTime": "2019-08-12 07:58:34"
        },
        {
            "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
            "Timestamp": "2019-08-12 08:02:54",
            "ConnectionId": "cvpn-connection-00668867a40f18253",
            "ConnectionEstablishedTime": "2019-08-12 08:02:53",
            "IngressBytes": "2951",
            "EgressBytes": "2611",
            "IngressPackets": "9",
            "EgressPackets": "6",
            "ClientIp": "172.31.0.226",
            "CommonName": "client1.domain.tld",
            "Status": {
                "Code": "active"
            },
            "ConnectionEndTime": "-"
        }
    ]
}
```

如需詳細資訊，請參閱 Client VPN 管理員指南中的用戶端連線。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClientVpnConnections。

describe-client-vpn-endpoints

以下程式碼範例顯示如何使用 `describe-client-vpn-endpoints`。

AWS CLI

描述您的 Client VPN 端點

下列`describe-client-vpn-endpoints`範例顯示所有 Client VPN 端點的詳細資訊。

```
aws ec2 describe-client-vpn-endpoints
```

輸出：

```
{
    "ClientVpnEndpoints": [
        {
            "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
            "Description": "Endpoint for Admin access",
            "Status": {
                "Code": "available"
            },
            "CreationTime": "2020-11-13T11:37:27",
            "DnsName": "*.cvpn-endpoint-123456789123abcde.prod.clientvpn.ap-
south-1.amazonaws.com",
            "ClientCidrBlock": "172.31.0.0/16",
            "DnsServers": [
                "8.8.8.8"
            ],
            "SplitTunnel": false,
            "VpnProtocol": "openvpn",
            "TransportProtocol": "udp",
            "VpnPort": 443,
            "ServerCertificateArn": "arn:aws:acm:ap-
south-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "AuthenticationOptions": [
                {
                    "Type": "certificate-authentication",
                    "MutualAuthentication": {
                        "ClientRootCertificateChain": "arn:aws:acm:ap-
south-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
                    }
                }
            ],
            "ConnectionLogOptions": {
                "Enabled": true,
                "CloudwatchLogGroup": "Client-vpn-connection-logs",
                "CloudwatchLogStream": "cvpn-endpoint-123456789123abcde-ap-
south-1-2020/11/13-FCD8HEMVaCcw"
            },
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Client VPN"
                }
            ],
```

```
            "SecurityGroupIds": [
                "sg-aabbcc11223344567"
            ],
            "VpcId": "vpc-a87f92c1",
            "SelfServicePortalUrl": "https://self-service.clientvpn.amazonaws.com/
endpoints/cvpn-endpoint-123456789123abcde",
            "ClientConnectOptions": {
                 "Enabled": false
            }
        }
    ]
}
```

如需詳細資訊，請參閱 Client VPN 管理員指南中的 Client VPN 端點。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClientVpnEndpoints。

`describe-client-vpn-routes`

以下程式碼範例顯示如何使用 `describe-client-vpn-routes`。

AWS CLI

描述 Client VPN 端點的路由

下列`describe-client-vpn-routes`範例顯示指定 Client VPN 端點路由的詳細資訊。

```
aws ec2 describe-client-vpn-routes \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

輸出：

```
{
    "Routes": [
        {
            "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
            "DestinationCidr": "10.0.0.0/16",
            "TargetSubnet": "subnet-0123456789abcabca",
            "Type": "Nat",
            "Origin": "associate",
            "Status": {
                "Code": "active"
```

```
            },
            "Description": "Default Route"
        },
        {
            "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
            "DestinationCidr": "0.0.0.0/0",
            "TargetSubnet": "subnet-0123456789abcabca",
            "Type": "Nat",
            "Origin": "add-route",
            "Status": {
                "Code": "active"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Client VPN 管理員指南中的路由。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClientVpnRoutes。

describe-client-vpn-target-networks

以下程式碼範例顯示如何使用 describe-client-vpn-target-networks。

AWS CLI

描述 Client VPN 端點的目標網路

下列describe-client-vpn-target-networks範例顯示指定 Client VPN 端點之目標網路的詳細資訊。

```
aws ec2 describe-client-vpn-target-networks \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

輸出：

```
{
    "ClientVpnTargetNetworks": [
        {
            "AssociationId": "cvpn-assoc-012e837060753dc3d",
            "VpcId": "vpc-11111222222333333",
            "TargetNetworkId": "subnet-0123456789abcabca",
```

```
            "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
            "Status": {
                "Code": "associating"
            },
            "SecurityGroups": [
                "sg-012345678910abcab"
            ]
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Client VPN 管理員指南中的目標網路。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClientVpnTargetNetworks。

describe-coip-pools

以下程式碼範例顯示如何使用 `describe-coip-pools`。

AWS CLI

描述客戶擁有的 IP 地址集區

下列`describe-coip-pools`範例說明您 AWS 帳戶中客戶擁有的 IP 地址集區。

```
aws ec2 describe-coip-pools
```

輸出：

```
{
    "CoipPools": [
        {
            "PoolId": "ipv4pool-coip-123a45678bEXAMPLE",
            "PoolCidrs": [
                "0.0.0.0/0"
            ],
            "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
            "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-
coip-123a45678bEXAMPLE"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Outposts 使用者指南中的客戶擁有的 IP 地址。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCoipPools。

describe-conversion-tasks

以下程式碼範例顯示如何使用 describe-conversion-tasks。

AWS CLI

檢視轉換任務的狀態

此範例會傳回 ID import-i-ffvko9js 轉換任務的狀態。

命令：

```
aws ec2 describe-conversion-tasks --conversion-task-ids import-i-ffvko9js
```

輸出：

```
{
    "ConversionTasks": [
        {
            "ConversionTaskId": "import-i-ffvko9js",
            "ImportInstance": {
                "InstanceId": "i-1234567890abcdef0",
                "Volumes": [
                    {
                        "Volume": {
                            "Id": "vol-049df61146c4d7901",
                            "Size": 16
                        },
                        "Status": "completed",
                        "Image": {
                            "Size": 1300687360,
                            "ImportManifestUrl": "https://s3.amazonaws.com/
myimportbucket/411443cd-d620-4f1c-9d66-13144EXAMPLE/RHEL5.vmdkmanifest.xml?
AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Expires=140EXAMPLE&Signature=XYNhznHNgCqsjDxL9wRL
%2FJvEXAMPLE",
                            "Format": "VMDK"
                        },
                        "BytesConverted": 1300682960,
```

```
                    "AvailabilityZone": "us-east-1d"
                }
            ]
        },
        "ExpirationTime": "2014-05-14T22:06:23Z",
        "State": "completed"
    }
]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeConversionTasks。

describe-customer-gateways

以下程式碼範例顯示如何使用 describe-customer-gateways。

AWS CLI

描述您的客戶閘道

此範例說明您的客戶閘道。

命令：

```
aws ec2 describe-customer-gateways
```

輸出：

```
{
    "CustomerGateways": [
        {
            "CustomerGatewayId": "cgw-b4dc3961",
            "IpAddress": "203.0.113.12",
            "State": "available",
            "Type": "ipsec.1",
            "BgpAsn": "65000"
        },
        {
            "CustomerGatewayId": "cgw-0e11f167",
            "IpAddress": "12.1.2.3",
            "State": "available",
            "Type": "ipsec.1",
```

```
            "BgpAsn": "65534"
        }
    ]
}
```

描述特定客戶閘道

此範例說明指定的客戶閘道。

命令：

```
aws ec2 describe-customer-gateways --customer-gateway-ids cgw-0e11f167
```

輸出：

```
{
    "CustomerGateways": [
        {
            "CustomerGatewayId": "cgw-0e11f167",
            "IpAddress": "12.1.2.3",
            "State": "available",
            "Type": "ipsec.1",
            "BgpAsn": "65534"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCustomerGateways。

describe-dhcp-options

以下程式碼範例顯示如何使用 describe-dhcp-options。

AWS CLI

範例 1：描述您的 DHCP 選項

下列describe-dhcp-options範例會擷取 DHCP 選項的詳細資訊。

```
aws ec2 describe-dhcp-options
```

輸出：

```
{
    "DhcpOptions": [
        {
            "DhcpConfigurations": [
                {
                    "Key": "domain-name",
                    "Values": [
                        {
                            "Value": "us-east-2.compute.internal"
                        }
                    ]
                },
                {
                    "Key": "domain-name-servers",
                    "Values": [
                        {
                            "Value": "AmazonProvidedDNS"
                        }
                    ]
                }
            ],
            "DhcpOptionsId": "dopt-19edf471",
            "OwnerId": "111122223333"
        },
        {
            "DhcpConfigurations": [
                {
                    "Key": "domain-name",
                    "Values": [
                        {
                            "Value": "us-east-2.compute.internal"
                        }
                    ]
                },
                {
                    "Key": "domain-name-servers",
                    "Values": [
                        {
                            "Value": "AmazonProvidedDNS"
                        }
                    ]
                }
```

```
            ],
            "DhcpOptionsId": "dopt-fEXAMPLE",
            "OwnerId": "111122223333"
        }
    ]
}
```

如需詳細資訊，請參閱AWS 《VPC 使用者指南》中的使用 DHCP 選項集。

範例 2：描述您的 DHCP 選項並篩選輸出

下列describe-dhcp-options範例說明您的 DHCP 選項，並使用篩選條件僅傳回example.com網域名稱伺服器具有 的 DHCP 選項。此範例使用 --query 參數，在輸出中僅顯示組態資訊和 ID。

```
aws ec2 describe-dhcp-options \
    --filters Name=key,Values=domain-name-servers Name=value,Values=example.com \
    --query "DhcpOptions[*].[DhcpConfigurations,DhcpOptionsId]"
```

輸出：

```
[
    [
        [
            {
                "Key": "domain-name",
                "Values": [
                    {
                        "Value": "example.com"
                    }
                ]
            },
            {
                "Key": "domain-name-servers",
                "Values": [
                    {
                        "Value": "172.16.16.16"
                    }
                ]
            }
        ],
        "dopt-001122334455667ab"
    ]
```

```
]
```

如需詳細資訊，請參閱AWS 《VPC 使用者指南》中的使用 DHCP 選項集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDhcpOptions。

describe-egress-only-internet-gateways

以下程式碼範例顯示如何使用 describe-egress-only-internet-gateways。

AWS CLI

描述您的輸出限定網際網路閘道

此範例說明您的輸出限定網際網路閘道。

命令：

```
aws ec2 describe-egress-only-internet-gateways
```

輸出：

```
{
  "EgressOnlyInternetGateways": [
      {
          "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
          "Attachments": [
              {
                  "State": "attached",
                  "VpcId": "vpc-0c62a468"
              }
          ]
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEgressOnlyInternetGateways。

describe-elastic-gpus

以下程式碼範例顯示如何使用 describe-elastic-gpus。

AWS CLI

描述彈性 GPU

命令：

```
aws ec2 describe-elastic-gpus --elastic-gpu-
ids egpu-12345678901234567890abcdefghijkl
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeElasticGpus。

describe-export-image-tasks

以下程式碼範例顯示如何使用 describe-export-image-tasks。

AWS CLI

監控匯出映像任務

下列describe-export-image-tasks範例會檢查指定匯出映像任務的狀態。Amazon S3 中產生的映像檔案為 my-export-bucket/exports/export-ami-1234567890abcdef0.vmdk。

```
aws ec2 describe-export-image-tasks \
    --export-image-task-ids export-ami-1234567890abcdef0
```

進行中匯出映像任務的輸出。

```
{
    "ExportImageTasks": [
        {
            "ExportImageTaskId": "export-ami-1234567890abcdef0"
            "Progress": "21",
            "S3ExportLocation": {
                "S3Bucket": "my-export-bucket",
                "S3Prefix": "exports/"
            },
            "Status": "active",
            "StatusMessage": "updating"
        }
    ]
}
```

匯出映像任務的輸出已完成。

```
{
    "ExportImageTasks": [
        {
            "ExportImageTaskId": "export-ami-1234567890abcdef0"
            "S3ExportLocation": {
                "S3Bucket": "my-export-bucket",
                "S3Prefix": "exports/"
            },
            "Status": "completed"
        }
    ]
}
```

如需詳細資訊，請參閱《VM Import/Export 使用者指南》中的從 AMI 匯出 VM。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeExportImageTasks。

`describe-export-tasks`

以下程式碼範例顯示如何使用 `describe-export-tasks`。

AWS CLI

列出執行個體匯出任務的詳細資訊

此範例說明 ID export-i-fh8sjjsq 的匯出任務。

命令：

```
aws ec2 describe-export-tasks --export-task-ids export-i-fh8sjjsq
```

輸出：

```
{
    "ExportTasks": [
        {
            "State": "active",
            "InstanceExportDetails": {
                "InstanceId": "i-1234567890abcdef0",
                "TargetEnvironment": "vmware"
            },
```

```
            "ExportToS3Task": {
                "S3Bucket": "myexportbucket",
                "S3Key": "RHEL5export-i-fh8sjjsq.ova",
                "DiskImageFormat": "vmdk",
                "ContainerFormat": "ova"
            },
            "Description": "RHEL5 instance",
            "ExportTaskId": "export-i-fh8sjjsq"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeExportTasks。

describe-fast-launch-images

以下程式碼範例顯示如何使用 `describe-fast-launch-images`。

AWS CLI

描述為更快速啟動而設定之 Windows AMIs 的詳細資訊

下列`describe-fast-launch-images`範例說明您帳戶中設定為更快速啟動的每個 AMIs 的詳細資訊，包括資源類型、快照組態、啟動範本詳細資訊、平行啟動次數上限、AMI 擁有者 ID、快速啟動組態的狀態、狀態變更的原因，以及狀態變更發生的時間。

```
aws ec2 describe-fast-launch-images
```

輸出：

```
{
    "FastLaunchImages": [
        {
            "ImageId": "ami-01234567890abcedf",
            "ResourceType": "snapshot",
            "SnapshotConfiguration": {},
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-01234567890abcedf",
                "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-
a8c6215d-94e6-441b-9272-dbd1f87b07e2",
                "Version": "1"
            },
```

```
            "MaxParallelLaunches": 6,
            "OwnerId": "0123456789123",
            "State": "enabled",
            "StateTransitionReason": "Client.UserInitiated",
            "StateTransitionTime": "2022-01-27T22:20:06.552000+00:00"
        }
    ]
}
```

如需設定 Windows AMI 以加快啟動速度的詳細資訊，請參閱《Amazon EC2 使用者指南》中的設定 AMI 以加快啟動速度。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFastLaunchImages。

describe-fast-snapshot-restores

以下程式碼範例顯示如何使用 describe-fast-snapshot-restores。

AWS CLI

描述快速快照還原

下列describe-fast-snapshot-restores範例顯示狀態為 之所有快速快照還原的詳細資訊disabled。

```
aws ec2 describe-fast-snapshot-restores \
    --filters Name=state,Values=disabled
```

輸出：

```
{
    "FastSnapshotRestores": [
        {
            "SnapshotId": "snap-1234567890abcdef0",
            "AvailabilityZone": "us-west-2c",
            "State": "disabled",
            "StateTransitionReason": "Client.UserInitiated - Lifecycle state
 transition",
            "OwnerId": "123456789012",
            "EnablingTime": "2020-01-25T23:57:49.596Z",
            "OptimizingTime": "2020-01-25T23:58:25.573Z",
            "EnabledTime": "2020-01-25T23:59:29.852Z",
```

```
            "DisablingTime": "2020-01-26T00:40:56.069Z",
            "DisabledTime": "2020-01-26T00:41:27.390Z"
        }
    ]
}
```

下列describe-fast-snapshot-restores範例說明所有快速快照還原。

```
aws ec2 describe-fast-snapshot-restores
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFastSnapshotRestores。

describe-fleet-history

以下程式碼範例顯示如何使用 describe-fleet-history。

AWS CLI

描述 EC2 機群歷史記錄

下列describe-fleet-history範例會傳回從指定時間開始的指定 EC2 機群歷史記錄。輸出適用於有兩個執行中執行個體的 EC2 機群。

```
aws ec2 describe-fleet-history \
    --fleet-id fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
    --start-time 2020-09-01T00:00:00Z
```

輸出：

```
{
    "HistoryRecords": [
        {
            "EventInformation": {
                "EventSubType": "submitted"
            },
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:05.000Z"
        },
        {
            "EventInformation": {
                "EventSubType": "active"
            },
```

```
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:15.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "t2.small, ami-07c8bc5c1ce9598c3, ...",
                "EventSubType": "progress"
            },
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "{\"instanceType\":\"t2.small\", ...}",
                "EventSubType": "launched",
                "InstanceId": "i-083a1c446e66085d2"
            },
            "EventType": "instanceChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "{\"instanceType\":\"t2.small\", ...}",
                "EventSubType": "launched",
                "InstanceId": "i-090db02406cc3c2d6"
            },
            "EventType": "instanceChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        }
    ],
    "LastEvaluatedTime": "2020-09-01T19:10:19.000Z",
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
    "StartTime": "2020-08-31T23:53:20.000Z"
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud Linux 執行個體使用者指南》中的管理 EC2 機群。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFleetHistory。

describe-fleet-instances

以下程式碼範例顯示如何使用 describe-fleet-instances。

AWS CLI

描述 EC2 機群的執行中執行個體

下列describe-fleet-instances範例說明指定 EC2 機群的執行中執行個體。

```
aws ec2 describe-fleet-instances \
    --fleet-id 12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE
```

輸出：

```
{
    "ActiveInstances": [
        {
            "InstanceId": "i-090db02406cc3c2d6",
            "InstanceType": "t2.small",
            "SpotInstanceRequestId": "sir-a43gtpfk",
            "InstanceHealth": "healthy"
        },
        {
            "InstanceId": "i-083a1c446e66085d2",
            "InstanceType": "t2.small",
            "SpotInstanceRequestId": "sir-iwcit2nj",
            "InstanceHealth": "healthy"
        }
    ],
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud Linux 執行個體使用者指南》中的管理 EC2 機群。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFleetInstances。

describe-fleets

以下程式碼範例顯示如何使用 describe-fleets。

AWS CLI

描述 EC2 機群

下列describe-fleets範例說明指定的 EC2 機群。

```
aws ec2 describe-fleets \
    --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE
```

輸出：

```
{
    "Fleets": [
        {
            "ActivityStatus": "pending_fulfillment",
            "CreateTime": "2020-09-01T18:26:05.000Z",
            "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
            "FleetState": "active",
            "ExcessCapacityTerminationPolicy": "termination",
            "FulfilledCapacity": 0.0,
            "FulfilledOnDemandCapacity": 0.0,
            "LaunchTemplateConfigs": [
                {
                    "LaunchTemplateSpecification": {
                        "LaunchTemplateId": "lt-0e632f2855a979cd5",
                        "Version": "1"
                    }
                }
            ],
            "TargetCapacitySpecification": {
                "TotalTargetCapacity": 2,
                "OnDemandTargetCapacity": 0,
                "SpotTargetCapacity": 2,
                "DefaultTargetCapacityType": "spot"
            },
            "TerminateInstancesWithExpiration": false,
            "Type": "maintain",
            "ReplaceUnhealthyInstances": false,
            "SpotOptions": {
                "AllocationStrategy": "lowestPrice",
                "InstanceInterruptionBehavior": "terminate",
                "InstancePoolsToUseCount": 1
            },
            "OnDemandOptions": {
                "AllocationStrategy": "lowestPrice"
            }
        }
```

```
    ]
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud Linux 執行個體使用者指南》中的管理 EC2 機群。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFleets。

describe-flow-logs

以下程式碼範例顯示如何使用 describe-flow-logs。

AWS CLI

範例 1：描述所有流程日誌

下列describe-flow-logs範例會顯示所有流程日誌的詳細資訊。

```
aws ec2 describe-flow-logs
```

輸出：

```
{
    "FlowLogs": [
        {
            "CreationTime": "2018-02-21T13:22:12.644Z",
            "DeliverLogsPermissionArn": "arn:aws:iam::123456789012:role/flow-logs-
role",
            "DeliverLogsStatus": "SUCCESS",
            "FlowLogId": "fl-aabbccdd112233445",
            "MaxAggregationInterval": 600,
            "FlowLogStatus": "ACTIVE",
            "LogGroupName": "FlowLogGroup",
            "ResourceId": "subnet-12345678901234567",
            "TrafficType": "ALL",
            "LogDestinationType": "cloud-watch-logs",
            "LogFormat": "${version} ${account-id} ${interface-id} ${srcaddr}
 ${dstaddr} ${srcport} ${dstport} ${protocol} ${packets} ${bytes} ${start} ${end}
 ${action} ${log-status}"
        },
        {
            "CreationTime": "2020-02-04T15:22:29.986Z",
```

```
            "DeliverLogsStatus": "SUCCESS",
            "FlowLogId": "fl-01234567890123456",
            "MaxAggregationInterval": 60,
            "FlowLogStatus": "ACTIVE",
            "ResourceId": "vpc-00112233445566778",
            "TrafficType": "ACCEPT",
            "LogDestinationType": "s3",
            "LogDestination": "arn:aws:s3:::my-flow-log-bucket/custom",
            "LogFormat": "${version} ${vpc-id} ${subnet-id} ${instance-id}
 ${interface-id} ${account-id} ${type} ${srcaddr} ${dstaddr} ${srcport} ${dstport}
 ${pkt-srcaddr} ${pkt-dstaddr} ${protocol} ${bytes} ${packets} ${start} ${end}
 ${action} ${tcp-flags} ${log-status}"
        }
    ]
}
```

範例 2：描述流程日誌的子集

下列`describe-flow-logs`範例使用篩選條件，僅顯示 Amazon CloudWatch Logs 中指定日誌群組中流程日誌的詳細資訊。

```
aws ec2 describe-flow-logs \
    --filter "Name=log-group-name,Values=MyFlowLogs"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFlowLogs。

`describe-fpga-image-attribute`

以下程式碼範例顯示如何使用 `describe-fpga-image-attribute`。

AWS CLI

描述 Amazon FPGA 映像的屬性

此範例說明指定 AFI 的載入許可。

命令：

```
aws ec2 describe-fpga-image-attribute --fpga-image-id afi-0d123e123bfc85abc --
attribute loadPermission
```

輸出：

```
{
  "FpgaImageAttribute": {
      "FpgaImageId": "afi-0d123e123bfc85abc",
      "LoadPermissions": [
          {
              "UserId": "123456789012"
          }
      ]
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFpgaImageAttribute。

describe-fpga-images

以下程式碼範例顯示如何使用 describe-fpga-images。

AWS CLI

描述 Amazon FPGA 映像

此範例說明帳戶 擁有AFIs123456789012。

命令：

```
aws ec2 describe-fpga-images --filters Name=owner-id,Values=123456789012
```

輸出：

```
{
  "FpgaImages": [
      {
          "UpdateTime": "2017-12-22T12:09:14.000Z",
          "Name": "my-afi",
          "PciId": {
              "SubsystemVendorId": "0xfedd",
              "VendorId": "0x1d0f",
              "DeviceId": "0xf000",
              "SubsystemId": "0x1d51"
          },
          "FpgaImageGlobalId": "agfi-123cb27b5e84a0abc",
```

```
            "Public": false,
            "State": {
                "Code": "available"
            },
            "ShellVersion": "0x071417d3",
            "OwnerId": "123456789012",
            "FpgaImageId": "afi-0d123e123bfc85abc",
            "CreateTime": "2017-12-22T11:43:33.000Z",
            "Description": "my-afi"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFpgaImages。

describe-host-reservation-offerings

以下程式碼範例顯示如何使用 describe-host-reservation-offerings。

AWS CLI

描述專用主機保留方案

此範例說明可供購買的 M4 執行個體系列的專用主機預留。

命令：

```
aws ec2 describe-host-reservation-offerings --filter Name=instance-family,Values=m4
```

輸出：

```
{
    "OfferingSet": [
        {
            "HourlyPrice": "1.499",
            "OfferingId": "hro-03f707bf363b6b324",
            "InstanceFamily": "m4",
            "PaymentOption": "NoUpfront",
            "UpfrontPrice": "0.000",
            "Duration": 31536000
        },
```

```
        {
            "HourlyPrice": "1.045",
            "OfferingId": "hro-0ef9181cabdef7a02",
            "InstanceFamily": "m4",
            "PaymentOption": "NoUpfront",
            "UpfrontPrice": "0.000",
            "Duration": 94608000
        },
        {
            "HourlyPrice": "0.714",
            "OfferingId": "hro-04567a15500b92a51",
            "InstanceFamily": "m4",
            "PaymentOption": "PartialUpfront",
            "UpfrontPrice": "6254.000",
            "Duration": 31536000
        },
        {
            "HourlyPrice": "0.484",
            "OfferingId": "hro-0d5d7a9d23ed7fbfe",
            "InstanceFamily": "m4",
            "PaymentOption": "PartialUpfront",
            "UpfrontPrice": "12720.000",
            "Duration": 94608000
        },
        {
            "HourlyPrice": "0.000",
            "OfferingId": "hro-05da4108ca998c2e5",
            "InstanceFamily": "m4",
            "PaymentOption": "AllUpfront",
            "UpfrontPrice": "23913.000",
            "Duration": 94608000
        },
        {
            "HourlyPrice": "0.000",
            "OfferingId": "hro-0a9f9be3b95a3dc8f",
            "InstanceFamily": "m4",
            "PaymentOption": "AllUpfront",
            "UpfrontPrice": "12257.000",
            "Duration": 31536000
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeHostReservationOfferings。

describe-host-reservations

以下程式碼範例顯示如何使用 describe-host-reservations。

AWS CLI

描述您帳戶中的專用主機預留

此範例說明您帳戶中的專用主機預留。

命令：

```
aws ec2 describe-host-reservations
```

輸出：

```
{
  "HostReservationSet": [
      {
          "Count": 1,
          "End": "2019-01-10T12:14:09Z",
          "HourlyPrice": "1.499",
          "InstanceFamily": "m4",
          "OfferingId": "hro-03f707bf363b6b324",
          "PaymentOption": "NoUpfront",
          "State": "active",
          "HostIdSet": [
              "h-013abcd2a00cbd123"
          ],
          "Start": "2018-01-10T12:14:09Z",
          "HostReservationId": "hr-0d418a3a4ffc669ae",
          "UpfrontPrice": "0.000",
          "Duration": 31536000
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeHostReservations。

describe-hosts

以下程式碼範例顯示如何使用 describe-hosts。

AWS CLI

檢視專用主機的詳細資訊

下列`describe-hosts`範例顯示您 AWS 帳戶中`available`專用主機的詳細資訊。

```
aws ec2 describe-hosts --filter "Name=state,Values=available"
```

輸出：

```
{
    "Hosts": [
        {
            "HostId": "h-07879acf49EXAMPLE",
            "Tags": [
                {
                    "Value": "production",
                    "Key": "purpose"
                }
            ],
            "HostProperties": {
                "Cores": 48,
                "TotalVCpus": 96,
                "InstanceType": "m5.large",
                "Sockets": 2
            },
            "Instances": [],
            "State": "available",
            "AvailabilityZone": "eu-west-1a",
            "AvailableCapacity": {
                "AvailableInstanceCapacity": [
                    {
                        "AvailableCapacity": 48,
                        "InstanceType": "m5.large",
                        "TotalCapacity": 48
                    }
                ],
                "AvailableVCpus": 96
            },
            "HostRecovery": "on",
            "AllocationTime": "2019-08-19T08:57:44.000Z",
            "AutoPlacement": "off"
        }
```

```
    ]
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud Linux 執行個體使用者指南》中的檢視專用主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeHosts。

`describe-iam-instance-profile-associations`

以下程式碼範例顯示如何使用 `describe-iam-instance-profile-associations`。

AWS CLI

描述 IAM 執行個體設定檔關聯

此範例描述所有 IAM 執行個體設定檔關聯。

命令：

```
aws ec2 describe-iam-instance-profile-associations
```

輸出：

```
{
  "IamInstanceProfileAssociations": [
      {
          "InstanceId": "i-09eb09efa73ec1dee",
          "State": "associated",
          "AssociationId": "iip-assoc-0db249b1f25fa24b8",
          "IamInstanceProfile": {
              "Id": "AIPAJVQN4F5WVLGCJDRGM",
              "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"
          }
      },
      {
          "InstanceId": "i-0402909a2f4dffd14",
          "State": "associating",
          "AssociationId": "iip-assoc-0d1ec06278d29f44a",
          "IamInstanceProfile": {
              "Id": "AGJAJVQN4F5WVLGCJABCM",
              "Arn": "arn:aws:iam::123456789012:instance-profile/user1-role"
          }
```

```
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeIamInstanceProfileAssociations。

describe-id-format

以下程式碼範例顯示如何使用 describe-id-format。

AWS CLI

範例 1：描述資源的 ID 格式

下列describe-id-format範例說明安全群組的 ID 格式。

```
aws ec2 describe-id-format \
    --resource security-group
```

在下列範例輸出中， Deadline值指出此資源類型的永久從短 ID 格式切換為長 ID 格式的截止日期，已於 2018 年 8 月 15 日 UTC 00：00 過期。

```
{
    "Statuses": [
        {
            "Deadline": "2018-08-15T00:00:00.000Z",
            "Resource": "security-group",
            "UseLongIds": true
        }
    ]
}
```

範例 2：描述所有資源的 ID 格式

下列describe-id-format範例說明所有資源類型的 ID 格式。支援短 ID 格式的所有資源類型都切換為使用長 ID 格式。

```
aws ec2 describe-id-format
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeIdFormat。

describe-identity-id-format

以下程式碼範例顯示如何使用 describe-identity-id-format。

AWS CLI

描述 IAM 角色的 ID 格式

下列describe-identity-id-format範例說明 IAM 角色EC2Role在您 AWS 帳戶中建立的執行個體所收到的 ID 格式。

```
aws ec2 describe-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:role/my-iam-role \
    --resource instance
```

下列輸出指出此角色建立的執行個體會收到長 IDs ID。

```
{
    "Statuses": [
        {
            "Deadline": "2016-12-15T00:00:00Z",
            "Resource": "instance",
            "UseLongIds": true
        }
    ]
}
```

描述 IAM 使用者的 ID 格式

下列describe-identity-id-format範例說明 AdminUser AWS 帳戶中 IAM 使用者所建立快照所收到的 ID 格式。

```
aws ec2 describe-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:user/AdminUser \
    --resource snapshot
```

輸出表示此使用者建立的快照會以長 IDs 格式接收 ID。

```
{
    "Statuses": [
        {
            "Deadline": "2016-12-15T00:00:00Z",
```

```
            "Resource": "snapshot",
            "UseLongIds": true
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeIdentityIdFormat。

`describe-image-attribute`

以下程式碼範例顯示如何使用 `describe-image-attribute`。

AWS CLI

描述 AMI 的啟動許可

此範例說明指定 AMI 的啟動許可。

命令：

```
aws ec2 describe-image-attribute --image-id ami-5731123e --
attribute launchPermission
```

輸出：

```
{
    "LaunchPermissions": [
        {
            "UserId": "123456789012"
        }
    ],
    "ImageId": "ami-5731123e",
}
```

描述 AMI 的產品代碼

此範例說明指定 AMI 的產品代碼。請注意，此 AMI 沒有產品代碼。

命令：

```
aws ec2 describe-image-attribute --image-id ami-5731123e --attribute productCodes
```

輸出：

```
{
    "ProductCodes": [],
    "ImageId": "ami-5731123e",
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeImageAttribute。

describe-images

以下程式碼範例顯示如何使用 describe-images。

AWS CLI

範例 1：描述 AMI

下列 describe-images 範例描述指定區域中的指定 AMI。

```
aws ec2 describe-images \
    --region us-east-1 \
    --image-ids ami-1234567890EXAMPLE
```

輸出：

```
{
    "Images": [
        {
            "VirtualizationType": "hvm",
            "Description": "Provided by Red Hat, Inc.",
            "PlatformDetails": "Red Hat Enterprise Linux",
            "EnaSupport": true,
            "Hypervisor": "xen",
            "State": "available",
            "SriovNetSupport": "simple",
            "ImageId": "ami-1234567890EXAMPLE",
            "UsageOperation": "RunInstances:0010",
            "BlockDeviceMappings": [
                {
                    "DeviceName": "/dev/sda1",
                    "Ebs": {
                        "SnapshotId": "snap-111222333444aaabb",
```

```
                        "DeleteOnTermination": true,
                        "VolumeType": "gp2",
                        "VolumeSize": 10,
                        "Encrypted": false
                    }
                }
            ],
            "Architecture": "x86_64",
            "ImageLocation": "123456789012/RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-
GP2",
            "RootDeviceType": "ebs",
            "OwnerId": "123456789012",
            "RootDeviceName": "/dev/sda1",
            "CreationDate": "2019-05-10T13:17:12.000Z",
            "Public": true,
            "ImageType": "machine",
            "Name": "RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-GP2"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Amazon Machine Images (AMI)。

範例 2：根據篩選條件描述 AMI

以下 `describe-images` 範例描述 Amazon 所提供，且受 Amazon EBS 支援的 Windows AMI。

```
aws ec2 describe-images \
    --owners amazon \
    --filters "Name=platform,Values=windows" "Name=root-device-type,Values=ebs"
```

如需 `describe-images` 的輸出範例，請參閱範例 1。

如需使用篩選條件的其他範例，請參閱《Amazon EC2 使用者指南》中的列出與篩選您的資源。

範例 3：根據標籤描述 AMI

下列 `describe-images` 範例描述具有標籤 `Type=Custom` 的所有 AMI。此範例使用 `--query` 參數，僅顯示 AMI ID。

```
aws ec2 describe-images \
    --filters "Name=tag:Type,Values=Custom" \
    --query 'Images[*].[ImageId]' \
```

```
    --output text
```

輸出：

```
ami-1234567890EXAMPLE
ami-0abcdef1234567890
```

如需使用標籤篩選條件的其他範例，請參閱《Amazon EC2 使用者指南》中的使用標籤。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeImages。

describe-import-image-tasks

以下程式碼範例顯示如何使用 describe-import-image-tasks。

AWS CLI

監控匯入映像任務

下列describe-import-image-tasks範例會檢查指定匯入映像任務的狀態。

```
aws ec2 describe-import-image-tasks \
    --import-task-ids import-ami-1234567890abcdef0
```

進行中匯入映像任務的輸出。

```
{
    "ImportImageTasks": [
        {
            "ImportTaskId": "import-ami-1234567890abcdef0",
            "Progress": "28",
            "SnapshotDetails": [
                {
                    "DiskImageSize": 705638400.0,
                    "Format": "ova",
                    "Status": "completed",
                    "UserBucket": {
                        "S3Bucket": "my-import-bucket",
                        "S3Key": "vms/my-server-vm.ova"
                    }
                }
            ],
```

```
            "Status": "active",
            "StatusMessage": "converting"
        }
    ]
}
```

已完成匯入映像任務的輸出。產生的 AMI ID 由 提供ImageId。

```
{
    "ImportImageTasks": [
        {
            "ImportTaskId": "import-ami-1234567890abcdef0",
            "ImageId": "ami-1234567890abcdef0",
            "SnapshotDetails": [
                {
                    "DiskImageSize": 705638400.0,
                    "Format": "ova",
                    "SnapshotId": "snap-1234567890abcdef0"
                    "Status": "completed",
                    "UserBucket": {
                        "S3Bucket": "my-import-bucket",
                        "S3Key": "vms/my-server-vm.ova"
                    }
                }
            ],
            "Status": "completed"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeImportImageTasks。

describe-import-snapshot-tasks

以下程式碼範例顯示如何使用 describe-import-snapshot-tasks。

AWS CLI

監控匯入快照任務

下列describe-import-snapshot-tasks範例會檢查指定匯入快照任務的狀態。

```
aws ec2 describe-import-snapshot-tasks \
```

```
    --import-task-ids import-snap-1234567890abcdef0
```

進行中匯入快照任務的輸出：

```
{
    "ImportSnapshotTasks": [
        {
            "Description": "My server VMDK",
            "ImportTaskId": "import-snap-1234567890abcdef0",
            "SnapshotTaskDetail": {
                "Description": "My server VMDK",
                "DiskImageSize": "705638400.0",
                "Format": "VMDK",
                "Progress": "42",
                "Status": "active",
                "StatusMessage": "downloading/converting",
                "UserBucket": {
                    "S3Bucket": "my-import-bucket",
                    "S3Key": "vms/my-server-vm.vmdk"
                }
            }
        }
    ]
}
```

已完成匯入快照任務的輸出。產生的快照 ID 由 提供SnapshotId。

```
{
    "ImportSnapshotTasks": [
        {
            "Description": "My server VMDK",
            "ImportTaskId": "import-snap-1234567890abcdef0",
            "SnapshotTaskDetail": {
                "Description": "My server VMDK",
                "DiskImageSize": "705638400.0",
                "Format": "VMDK",
                "SnapshotId": "snap-1234567890abcdef0"
                "Status": "completed",
                "UserBucket": {
                    "S3Bucket": "my-import-bucket",
                    "S3Key": "vms/my-server-vm.vmdk"
                }
            }
```

```
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeImportSnapshotTasks。

describe-instance-attribute

以下程式碼範例顯示如何使用 describe-instance-attribute。

AWS CLI

描述執行個體類型

此範例說明指定執行個體的執行個體類型。

命令：

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --
attribute instanceType
```

輸出：

```
{
    "InstanceId": "i-1234567890abcdef0"
    "InstanceType": {
        "Value": "t1.micro"
    }
}
```

描述 disableApiTermination 屬性

此範例說明指定執行個體的 disableApiTermination 屬性。

命令：

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --
attribute disableApiTermination
```

輸出：

```
{
```

```
"InstanceId": "i-1234567890abcdef0"
    "DisableApiTermination": {
        "Value": "false"
    }
}
```

描述執行個體的區塊型設備映射

此範例說明指定執行個體的 blockDeviceMapping 屬性。

命令：

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --
attribute blockDeviceMapping
```

輸出：

```
{
    "InstanceId": "i-1234567890abcdef0"
    "BlockDeviceMappings": [
        {
            "DeviceName": "/dev/sda1",
            "Ebs": {
                "Status": "attached",
                "DeleteOnTermination": true,
                "VolumeId": "vol-049df61146c4d7901",
                "AttachTime": "2013-05-17T22:42:34.000Z"
            }
        },
        {
            "DeviceName": "/dev/sdf",
            "Ebs": {
                "Status": "attached",
                "DeleteOnTermination": false,
                "VolumeId": "vol-049df61146c4d7901",
                "AttachTime": "2013-09-10T23:07:00.000Z"
            }
        }
    ],
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInstanceAttribute。

describe-instance-connect-endpoints

以下程式碼範例顯示如何使用 describe-instance-connect-endpoints。

AWS CLI

描述 EC2 Instance Connect Endpoint

下列describe-instance-connect-endpoints範例說明指定的 EC2 Instance Connect Endpoint。

```
aws ec2 describe-instance-connect-endpoints \
    --region us-east-1 \
    --instance-connect-endpoint-ids eice-0123456789example
```

輸出：

```
{
    "InstanceConnectEndpoints": [
        {
            "OwnerId": "111111111111",
            "InstanceConnectEndpointId": "eice-0123456789example",
            "InstanceConnectEndpointArn": "arn:aws:ec2:us-
east-1:111111111111:instance-connect-endpoint/eice-0123456789example",
            "State": "create-complete",
            "StateMessage": "",
            "DnsName": "eice-0123456789example.b67b86ba.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com",
            "NetworkInterfaceIds": [
                "eni-0123456789example"
            ],
            "VpcId": "vpc-0123abcd",
            "AvailabilityZone": "us-east-1d",
            "CreatedAt": "2023-02-07T12:05:37+00:00",
            "SubnetId": "subnet-0123abcd",
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的建立 EC2 Instance Connect Endpoint。Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInstanceConnectEndpoints。

describe-instance-credit-specifications

以下程式碼範例顯示如何使用 describe-instance-credit-specifications。

AWS CLI

描述一或多個執行個體 CPU 用量的點數選項

下列describe-instance-credit-specifications範例說明指定執行個體的 CPU 點數選項。

```
aws ec2 describe-instance-credit-specifications \
    --instance-ids i-1234567890abcdef0
```

輸出：

```
{
    "InstanceCreditSpecifications": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "CpuCredits": "unlimited"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的使用爆量效能執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInstanceCreditSpecifications。

describe-instance-event-notification-attributes

以下程式碼範例顯示如何使用 describe-instance-event-notification-attributes。

AWS CLI

描述排程事件通知的標籤

下列describe-instance-event-notification-attributes範例說明要在排程事件通知中顯示的標籤。

```
aws ec2 describe-instance-event-notification-attributes
```

輸出：

```
{
    "InstanceTagAttribute": {
        "InstanceTagKeys": [],
        "IncludeAllTagsOfInstance": true
    }
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud Linux 執行個體使用者指南》中的執行個體的排程事件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInstanceEventNotificationAttributes。

describe-instance-event-windows

以下程式碼範例顯示如何使用 describe-instance-event-windows。

AWS CLI

範例 1：描述所有事件時段

下列describe-instance-event-windows範例說明指定區域中的所有事件時段。

```
aws ec2 describe-instance-event-windows \
    --region us-east-1
```

輸出：

```
{
    "InstanceEventWindows": [
        {
            "InstanceEventWindowId": "iew-0abcdef1234567890",
            "Name": "myEventWindowName",
            "CronExpression": "* 21-23 * * 2,3",
            "AssociationTarget": {
                "InstanceIds": [
                    "i-1234567890abcdef0",
```

```
                    "i-0598c7d356eba48d7"
                ],
                "Tags": [],
                "DedicatedHostIds": []
            },
            "State": "active",
            "Tags": []
        }

        ...

    ],
    "NextToken": "9d624e0c-388b-4862-a31e-a85c64fc1d4a"
}
```

範例 2：描述特定事件時段

下列describe-instance-event-windows範例使用 instance-event-window 參數描述特定事件時段，藉此描述特定事件。

```
aws ec2 describe-instance-event-windows \
    --region us-east-1 \
    --instance-event-window-ids iew-0abcdef1234567890
```

輸出：

```
{
    "InstanceEventWindows": [
        {
            "InstanceEventWindowId": "iew-0abcdef1234567890",
            "Name": "myEventWindowName",
            "CronExpression": "* 21-23 * * 2,3",
            "AssociationTarget": {
                "InstanceIds": [
                    "i-1234567890abcdef0",
                    "i-0598c7d356eba48d7"
                ],
                "Tags": [],
                "DedicatedHostIds": []
            },
            "State": "active",
            "Tags": []
        }
```

```
}
```

範例 3：描述符合一或多個篩選條件的事件時段

下列describe-instance-event-windows範例說明使用 filter 參數符合一或多個篩選條件的事件時段。instance-id 篩選條件用於描述與指定執行個體相關聯的所有事件時段。使用篩選條件時，它會執行直接比對。不過，instance-id 篩選條件是不同的。如果沒有直接符合執行個體 ID，則會回復到與事件視窗的間接關聯，例如執行個體的標籤或專用主機 ID （如果執行個體是專用主機）。

```
aws ec2 describe-instance-event-windows \
    --region us-east-1 \
    --filters Name=instance-id,Values=i-1234567890abcdef0 \
    --max-results 100 \
    --next-token <next-token-value>
```

輸出：

```
{
    "InstanceEventWindows": [
        {
            "InstanceEventWindowId": "iew-0dbc0adb66f235982",
            "TimeRanges": [
                {
                    "StartWeekDay": "sunday",
                    "StartHour": 2,
                    "EndWeekDay": "sunday",
                    "EndHour": 8
                }
            ],
            "Name": "myEventWindowName",
            "AssociationTarget": {
                "InstanceIds": [],
                "Tags": [],
                "DedicatedHostIds": [
                    "h-0140d9a7ecbd102dd"
                ]
            },
            "State": "active",
            "Tags": []
        }
    ]
```

```
}
```

在範例輸出中，執行個體位於與事件視窗相關聯的專用主機上。

如需事件時段限制，請參閱《Amazon EC2 使用者指南》中的考量事項。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInstanceEventWindows。

describe-instance-image-metadata

以下程式碼範例顯示如何使用 describe-instance-image-metadata。

AWS CLI

範例 1：描述所有執行個體的 AMI 中繼資料

下列describe-instance-image-metadata範例說明您 AWS 帳戶中指定區域中所有執行個體的 AMI 中繼資料。

```
aws ec2 describe-instance-image-metadata \
    --region us-east-1
```

輸出：

```
{
    "InstanceImageMetadata": [
        {
            "InstanceId": "i-1234567890EXAMPLE",
            "InstanceType": "t2.micro",
            "LaunchTime": "2024-08-28T11:25:45+00:00",
            "AvailabilityZone": "us-east-1a",
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "OwnerId": "123412341234",
            "Tags": [
                {
                    "Key": "MyTagName",
                    "Value": "my-tag-value"
                }
            ],
            "ImageMetadata": {
```

```
                "ImageId": "ami-0b752bf1df193a6c4",
                "Name": "al2023-ami-2023.5.20240819.0-kernel-6.1-x86_64",
                "OwnerId": "137112412989",
                "State": "available",
                "ImageOwnerAlias": "amazon",
                "CreationDate": "2023-01-25T17:20:40Z",
                "DeprecationTime": "2025-01-25T17:20:40Z",
                "IsPublic": true
            }
        }
    ],
    "NextToken": "...EXAMPLEwIAABAA2JHaFxLnEXAMPLE..."
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Amazon EC2 中的 Amazon Machine Image。 Amazon EC2

範例 2：描述指定執行個體的 AMI 中繼資料

下列describe-instance-image-metadata範例說明指定執行個體的 AMI 中繼資料。

```
aws ec2 describe-instance-image-metadata \
    --region us-east-1 \
    --instance-ids i-1234567890EXAMPLE i-0987654321EXAMPLE
```

輸出：

```
{
    "InstanceImageMetadata": [
        {
            "InstanceId": "i-1234567890EXAMPLE",
            "InstanceType": "t2.micro",
            "LaunchTime": "2024-08-28T11:25:45+00:00",
            "AvailabilityZone": "us-east-1a",
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "OwnerId": "123412341234",
            "Tags": [
                {
                    "Key": "MyTagName",
                    "Value": "my-tag-value"
```

```
                }
            ],
            "ImageMetadata": {
                "ImageId": "ami-0b752bf1df193a6c4",
                "Name": "al2023-ami-2023.5.20240819.0-kernel-6.1-x86_64",
                "OwnerId": "137112412989",
                "State": "available",
                "ImageOwnerAlias": "amazon",
                "CreationDate": "2023-01-25T17:20:40Z",
                "DeprecationTime": "2025-01-25T17:20:40Z",
                "IsPublic": true
            }
        },
        {
            "InstanceId": "i-0987654321EXAMPLE",
            "InstanceType": "t2.micro",
            "LaunchTime": "2024-08-28T11:25:45+00:00",
            "AvailabilityZone": "us-east-1a",
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "OwnerId": "123412341234",
            "Tags": [
                {
                    "Key": "MyTagName",
                    "Value": "my-tag-value"
                }
            ],
            "ImageMetadata": {
                "ImageId": "ami-0b752bf1df193a6c4",
                "Name": "al2023-ami-2023.5.20240819.0-kernel-6.1-x86_64",
                "OwnerId": "137112412989",
                "State": "available",
                "ImageOwnerAlias": "amazon",
                "CreationDate": "2023-01-25T17:20:40Z",
                "DeprecationTime": "2025-01-25T17:20:40Z",
                "IsPublic": true
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Amazon EC2 中的 Amazon Machine Image。 Amazon EC2

範例 3：根據篩選條件描述執行個體的 AMI 中繼資料

下列describe-instance-image-metadata範例說明us-east-1a可用區域中 t2.nano和 t2.micro執行個體的 AMI 中繼資料。

```
aws ec2 describe-instance-image-metadata \
    --region us-east-1 \
    --filters Name=availability-zone,Values=us-east-1a Name=instance-
type,Values=t2.nano,t2.micro
```

輸出：

```
{
    "InstanceImageMetadata": [
        {
            "InstanceId": "i-1234567890EXAMPLE",
            "InstanceType": "t2.micro",
            "LaunchTime": "2024-08-28T11:25:45+00:00",
            "AvailabilityZone": "us-east-1a",
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "OwnerId": "123412341234",
            "Tags": [
                {
                    "Key": "MyTagName",
                    "Value": "my-tag-value"
                }
            ],
            "ImageMetadata": {
                "ImageId": "ami-0b752bf1df193a6c4",
                "Name": "al2023-ami-2023.5.20240819.0-kernel-6.1-x86_64",
                "OwnerId": "137112412989",
                "State": "available",
                "ImageOwnerAlias": "amazon",
                "CreationDate": "2023-01-25T17:20:40Z",
                "DeprecationTime": "2025-01-25T17:20:40Z",
                "IsPublic": true
            }
```

```
        },
        {
            "InstanceId": "i-0987654321EXAMPLE",
            "InstanceType": "t2.micro",
            "LaunchTime": "2024-08-28T11:25:45+00:00",
            "AvailabilityZone": "us-east-1a",
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "OwnerId": "123412341234",
            "Tags": [
                {
                    "Key": "MyTagName",
                    "Value": "my-tag-value"
                }
            ],
            "ImageMetadata": {
                "ImageId": "ami-0b752bf1df193a6c4",
                "Name": "al2023-ami-2023.5.20240819.0-kernel-6.1-x86_64",
                "OwnerId": "137112412989",
                "State": "available",
                "ImageOwnerAlias": "amazon",
                "CreationDate": "2023-01-25T17:20:40Z",
                "DeprecationTime": "2025-01-25T17:20:40Z",
                "IsPublic": true
            }
        }
    ],
    "NextToken": "...EXAMPLEV7ixRYHwIAABAA2JHaFxLnDAzpatfEXAMPLE..."
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Amazon EC2 中的 Amazon Machine Image。 Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInstanceImageMetadata。

`describe-instance-status`

以下程式碼範例顯示如何使用 `describe-instance-status`。

AWS CLI

描述執行個體的狀態

下列 describe-instance-status 範例會描述指定執行個體的目前狀態。

```
aws ec2 describe-instance-status \
    --instance-ids i-1234567890abcdef0
```

輸出：

```
{
    "InstanceStatuses": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "InstanceState": {
                "Code": 16,
                "Name": "running"
            },
            "AvailabilityZone": "us-east-1d",
            "SystemStatus": {
                "Status": "ok",
                "Details": [
                    {
                        "Status": "passed",
                        "Name": "reachability"
                    }
                ]
            },
            "InstanceStatus": {
                "Status": "ok",
                "Details": [
                    {
                        "Status": "passed",
                        "Name": "reachability"
                    }
                ]
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的監控您的執行個體狀態。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeInstanceStatus。

describe-instance-topology

以下程式碼範例顯示如何使用 describe-instance-topology。

AWS CLI

描述所有執行個體的執行個體拓撲

下列describe-instance-topology範例說明所有執行個體的拓撲，這些執行個體符合此命令支援的執行個體類型。

```
aws ec2 describe-instance-topology \
    --region us-west-2
```

輸出：

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "my-ml-cpg",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "p4d.24xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
```

```
        {
            "InstanceId": "i-3333333333example",
            "InstanceType": "trn1.32xlarge",
            "NetworkNodes": [
                "nn-1212121212example",
                "nn-1211122211example",
                "nn-1311133311example"
            ],
            "ZoneId": "usw2-az4",
            "AvailabilityZone": "us-west-2d"
        },
        {
            "InstanceId": "i-444444444example",
            "InstanceType": "trn1.2xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-5434334334example",
                "nn-1235301234example"
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

如需詳細資訊，包括更多範例，請參閱《Amazon EC2 使用者指南》中的 Amazon EC2 執行個體拓撲。 Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInstanceTopology。

describe-instance-type-offerings

以下程式碼範例顯示如何使用 describe-instance-type-offerings。

AWS CLI

範例 1：列出區域中提供的執行個體類型

下列describe-instance-type-offerings範例列出 區域中提供的執行個體類型，設定為 CLI AWS 的預設區域。

```
aws ec2 describe-instance-type-offerings
```

若要列出不同區域中提供的執行個體類型，請使用 --region 參數指定區域。

```
aws ec2 describe-instance-type-offerings \
    --region us-east-2
```

輸出：

```
{
  "InstanceTypeOfferings": [
      {
          "InstanceType": "m5.2xlarge",
          "LocationType": "region",
          "Location": "us-east-2"
      },
      {
          "InstanceType": "t3.micro",
          "LocationType": "region",
          "Location": "us-east-2"
      },
      ...
  ]
}
```

範例 2：列出可用區域中提供的執行個體類型

下列describe-instance-type-offerings範例列出指定可用區域中提供的執行個體類型。可用區域必須位於指定的區域。

```
aws ec2 describe-instance-type-offerings \
    --location-type availability-zone \
    --filters Name=location,Values=us-east-2a \
    --region us-east-2
```

範例 3：檢查是否支援執行個體類型

下列describe-instance-type-offerings命令指出指定區域中是否支援c5.xlarge執行個體類型。

```
aws ec2 describe-instance-type-offerings \
    --filters Name=instance-type,Values=c5.xlarge \
```

```
    --region us-east-2
```

下列describe-instance-type-offerings範例列出指定區域中支援的所有 C5 執行個體類型。

```
aws ec2 describe-instance-type-offerings \
    --filters Name=instance-type,Values=c5* \
    --query "InstanceTypeOfferings[].InstanceType" \
    --region us-east-2
```

輸出：

```
[
    "c5d.12xlarge",
    "c5d.9xlarge",
    "c5n.xlarge",
    "c5.xlarge",
    "c5d.metal",
    "c5n.metal",
    "c5.large",
    "c5d.2xlarge",
    "c5n.4xlarge",
    "c5.2xlarge",
    "c5n.large",
    "c5n.9xlarge",
    "c5d.large",
    "c5.18xlarge",
    "c5d.18xlarge",
    "c5.12xlarge",
    "c5n.18xlarge",
    "c5.metal",
    "c5d.4xlarge",
    "c5.24xlarge",
    "c5d.xlarge",
    "c5n.2xlarge",
    "c5d.24xlarge",
    "c5.9xlarge",
    "c5.4xlarge"
]
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInstanceTypeOfferings。

describe-instance-types

以下程式碼範例顯示如何使用 describe-instance-types。

AWS CLI

範例 1：描述執行個體類型

下列 describe-instance-types 範例顯示指定執行個體類型的詳細資訊。

```
aws ec2 describe-instance-types \
    --instance-types t2.micro
```

輸出：

```
{
    "InstanceTypes": [
        {
            "InstanceType": "t2.micro",
            "CurrentGeneration": true,
            "FreeTierEligible": true,
            "SupportedUsageClasses": [
                "on-demand",
                "spot"
            ],
            "SupportedRootDeviceTypes": [
                "ebs"
            ],
            "BareMetal": false,
            "Hypervisor": "xen",
            "ProcessorInfo": {
                "SupportedArchitectures": [
                    "i386",
                    "x86_64"
                ],
                "SustainedClockSpeedInGhz": 2.5
            },
            "VCpuInfo": {
                "DefaultVCpus": 1,
                "DefaultCores": 1,
                "DefaultThreadsPerCore": 1,
                "ValidCores": [
                    1
                ],
```

```
                "ValidThreadsPerCore": [
                    1
                ]
            },
            "MemoryInfo": {
                "SizeInMiB": 1024
            },
            "InstanceStorageSupported": false,
            "EbsInfo": {
                "EbsOptimizedSupport": "unsupported",
                "EncryptionSupport": "supported"
            },
            "NetworkInfo": {
                "NetworkPerformance": "Low to Moderate",
                "MaximumNetworkInterfaces": 2,
                "Ipv4AddressesPerInterface": 2,
                "Ipv6AddressesPerInterface": 2,
                "Ipv6Supported": true,
                "EnaSupport": "unsupported"
            },
            "PlacementGroupInfo": {
                "SupportedStrategies": [
                    "partition",
                    "spread"
                ]
            },
            "HibernationSupported": false,
            "BurstablePerformanceSupported": true,
            "DedicatedHostsSupported": false,
            "AutoRecoverySupported": true
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud Linux 執行個體使用者指南》中的執行個體類型。

範例 2：若要篩選可用的執行個體類型

您可以指定篩選條件，將結果範圍限制為具有特定特性的執行個體類型。下列 `describe-instance-types` 範例列出支援休眠的執行個體類型。

```
aws ec2 describe-instance-types \
```

```
    --filters Name=hibernation-supported,Values=true --query
 'InstanceTypes[*].InstanceType'
```

輸出：

```
[
    "m5.8xlarge",
    "r3.large",
    "c3.8xlarge",
    "r5.large",
    "m4.4xlarge",
    "c4.large",
    "m5.xlarge",
    "m4.xlarge",
    "c3.large",
    "c4.8xlarge",
    "c4.4xlarge",
    "c5.xlarge",
    "c5.12xlarge",
    "r5.4xlarge",
    "c5.4xlarge"
]
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud Linux 執行個體使用者指南》中的執行個體類型。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeInstanceTypes。

describe-instances

以下程式碼範例顯示如何使用 describe-instances。

AWS CLI

範例 1：描述執行個體

下列 describe-instances 範例會描述指定的執行個體。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0
```

輸出：

```
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-0abcdef1234567890",
                    "InstanceId": "i-1234567890abcdef0",
                    "InstanceType": "t3.nano",
                    "KeyName": "my-key-pair",
                    "LaunchTime": "2022-11-15T10:48:59+00:00",
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "Placement": {
                        "AvailabilityZone": "us-east-2a",
                        "GroupName": "",
                        "Tenancy": "default"
                    },
                    "PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
                    "PrivateIpAddress": "10-0-0-157",
                    "ProductCodes": [],
                    "PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
                    "PublicIpAddress": "34.253.223.13",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "StateTransitionReason": "",
                    "SubnetId": "subnet-04a636d18e83cfacb",
                    "VpcId": "vpc-1234567890abcdef0",
                    "Architecture": "x86_64",
                    "BlockDeviceMappings": [
                        {
                            "DeviceName": "/dev/xvda",
                            "Ebs": {
                                "AttachTime": "2022-11-15T10:49:00+00:00",
                                "DeleteOnTermination": true,
                                "Status": "attached",
                                "VolumeId": "vol-02e6ccdca7de29cf2"
                            }
```

```
                        }
                    ],
                    "ClientToken": "1234abcd-1234-abcd-1234-d46a8903e9bc",
                    "EbsOptimized": true,
                    "EnaSupport": true,
                    "Hypervisor": "xen",
                    "IamInstanceProfile": {
                        "Arn": "arn:aws:iam::111111111111:instance-profile/
AmazonSSMRoleForInstancesQuickSetup",
                        "Id": "11111111111111111111"
                    },
                    "NetworkInterfaces": [
                        {
                            "Association": {
                                "IpOwnerId": "amazon",
                                "PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
                                "PublicIp": "34.253.223.13"
                            },
                            "Attachment": {
                                "AttachTime": "2022-11-15T10:48:59+00:00",
                                "AttachmentId": "eni-attach-1234567890abcdefg",
                                "DeleteOnTermination": true,
                                "DeviceIndex": 0,
                                "Status": "attached",
                                "NetworkCardIndex": 0
                            },
                            "Description": "",
                            "Groups": [
                                {
                                    "GroupName": "launch-wizard-146",
                                    "GroupId": "sg-1234567890abcdefg"
                                }
                            ],
                            "Ipv6Addresses": [],
                            "MacAddress": "00:11:22:33:44:55",
                            "NetworkInterfaceId": "eni-1234567890abcdefg",
                            "OwnerId": "104024344472",
                            "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
                            "PrivateIpAddress": "10-0-0-157",
                            "PrivateIpAddresses": [
                                {
                                    "Association": {
```

```
                                    "IpOwnerId": "amazon",
                                    "PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
                                    "PublicIp": "34.253.223.13"
                                },
                                "Primary": true,
                                "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
                                "PrivateIpAddress": "10-0-0-157"
                            }
                        ],
                        "SourceDestCheck": true,
                        "Status": "in-use",
                        "SubnetId": "subnet-1234567890abcdefg",
                        "VpcId": "vpc-1234567890abcdefg",
                        "InterfaceType": "interface"
                    }
                ],
                "RootDeviceName": "/dev/xvda",
                "RootDeviceType": "ebs",
                "SecurityGroups": [
                    {
                        "GroupName": "launch-wizard-146",
                        "GroupId": "sg-1234567890abcdefg"
                    }
                ],
                "SourceDestCheck": true,
                "Tags": [
                    {
                        "Key": "Name",
                        "Value": "my-instance"
                    }
                ],
                "VirtualizationType": "hvm",
                "CpuOptions": {
                    "CoreCount": 1,
                    "ThreadsPerCore": 2
                },
                "CapacityReservationSpecification": {
                    "CapacityReservationPreference": "open"
                },
                "HibernationOptions": {
                    "Configured": false
                },
```

```
                    "MetadataOptions": {
                        "State": "applied",
                        "HttpTokens": "optional",
                        "HttpPutResponseHopLimit": 1,
                        "HttpEndpoint": "enabled",
                        "HttpProtocolIpv6": "disabled",
                        "InstanceMetadataTags": "enabled"
                    },
                    "EnclaveOptions": {
                        "Enabled": false
                    },
                    "PlatformDetails": "Linux/UNIX",
                    "UsageOperation": "RunInstances",
                    "UsageOperationUpdateTime": "2022-11-15T10:48:59+00:00",
                    "PrivateDnsNameOptions": {
                        "HostnameType": "ip-name",
                        "EnableResourceNameDnsARecord": true,
                        "EnableResourceNameDnsAAAARecord": false
                    },
                    "MaintenanceOptions": {
                        "AutoRecovery": "default"
                    }
                }
            ],
            "OwnerId": "111111111111",
            "ReservationId": "r-1234567890abcdefg"
        }
    ]
}
```

範例 2：篩選具有指定類型的執行個體

下列 `describe-instances` 範例會使用篩選條件，將結果範圍限定為指定類型的執行個體。

```
aws ec2 describe-instances \
    --filters Name=instance-type,Values=m5.large
```

如需輸出範例，請參閱範例 1。

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的使用 CLI 列出和篩選。

範例 3：篩選具有指定類型和可用區域的執行個體

下列 describe-instances 範例會使用多個篩選條件，將結果範圍限定為指定可用區域中具有指定類型的執行個體。

```
aws ec2 describe-instances \
    --filters Name=instance-type,Values=t2.micro,t3.micro Name=availability-zone,Values=us-east-2c
```

如需輸出範例，請參閱範例 1。

範例 4：使用 JSON 檔案篩選具有指定類型和可用區域的執行個體

下列 describe-instances 範例會使用 JSON 輸入檔案來執行與先前範例相同的篩選條件。若篩選條件變得更複雜，便可更輕鬆地在 JSON 檔案中指定這些條件。

```
aws ec2 describe-instances \
    --filters file://filters.json
```

filters.json 的內容：

```
[
    {
        "Name": "instance-type",
        "Values": ["t2.micro", "t3.micro"]
    },
    {
        "Name": "availability-zone",
        "Values": ["us-east-2c"]
    }
]
```

如需輸出範例，請參閱範例 1。

範例 5：篩選具有指定 Owner 標籤的執行個體

下列 describe-instances 範例會使用標籤篩選條件，將結果範圍限定為其標籤具有指定標籤索引鍵 (Owner) 的執行個體，不論標籤值為何。

```
aws ec2 describe-instances \
    --filters "Name=tag-key,Values=Owner"
```

如需輸出範例，請參閱範例 1。

範例 6：篩選具有指定 my-team 標籤值的執行個體

下列 `describe-instances` 範例會使用標籤篩選條件，將結果範圍限定為其標籤具有指定標籤值 (my-team) 的執行個體，不論標籤索引鍵為何。

```
aws ec2 describe-instances \
    --filters "Name=tag-value,Values=my-team"
```

如需輸出範例，請參閱範例 1。

範例 7：篩選具有指定 Owner 標籤和 my-team 值的執行個體

下列 `describe-instances` 範例會使用標籤篩選條件，將結果範圍限定為具有指定標籤 (Owner=my-team) 的執行個體。

```
aws ec2 describe-instances \
    --filters "Name=tag:Owner,Values=my-team"
```

如需輸出範例，請參閱範例 1。

範例 8：僅顯示所有執行個體的執行個體和子網路 ID

下列 `describe-instances` 範例會使用 `--query` 參數，以 JSON 格式僅顯示所有執行個體的執行個體和子網路 ID。

Linux 和 macOS：

```
aws ec2 describe-instances \
    --query 'Reservations[*].Instances[*].{Instance:InstanceId,Subnet:SubnetId}' \
    --output json
```

Windows：

```
aws ec2 describe-instances ^
    --query "Reservations[*].Instances[*].{Instance:InstanceId,Subnet:SubnetId}" ^
    --output json
```

輸出：

```
[
    {
```

```
        "Instance": "i-057750d42936e468a",
        "Subnet": "subnet-069beee9b12030077"
    },
    {
        "Instance": "i-001efd250faaa6ffa",
        "Subnet": "subnet-0b715c6b7db68927a"
    },
    {
        "Instance": "i-027552a73f021f3bd",
        "Subnet": "subnet-0250c25a1f4e15235"
    }
    ...
]
```

範例 9：篩選指定類型的執行個體，並僅顯示其執行個體 ID

下列 describe-instances 範例會使用篩選條件，將結果範圍限定為指定類型的執行個體以及 --query 參數，以便僅顯示執行個體 ID。

```
aws ec2 describe-instances \
    --filters "Name=instance-type,Values=t2.micro" \
    --query "Reservations[*].Instances[*].[InstanceId]" \
    --output text
```

輸出：

```
i-031c0dc19de2fb70c
i-00d8bff789a736b75
i-0b715c6b7db68927a
i-0626d4edd54f1286d
i-00b8ae04f9f99908e
i-0fc71c25d2374130c
```

範例 10：篩選指定類型的執行個體，並僅顯示其執行個體 ID、可用區域以及指定標籤值

下列 describe-instances 範例會針對其標籤具有名稱 tag-key 的執行個體，以表格格式顯示執行個體 ID、可用區域以及 Name 標籤的值。

Linux 和 macOS：

```
aws ec2 describe-instances \
    --filters Name=tag-key,Values=Name \
```

```
    --query 'Reservations[*].Instances[*].
{Instance:InstanceId,AZ:Placement.AvailabilityZone,Name:Tags[?Key==`Name`]|
[0].Value}' \
    --output table
```

Windows：

```
aws ec2 describe-instances ^
    --filters Name=tag-key,Values=Name ^
    --query "Reservations[*].Instances[*].
{Instance:InstanceId,AZ:Placement.AvailabilityZone,Name:Tags[?Key=='Name']|
[0].Value}" ^
    --output table
```

輸出：

```
-------------------------------------------------------------
|                     DescribeInstances                     |
+--------------+-----------------------+--------------------+
|      AZ      |       Instance        |        Name        |
+--------------+-----------------------+--------------------+
|  us-east-2b  |  i-057750d42936e468a  |  my-prod-server    |
|  us-east-2a  |  i-001efd250faaa6ffa  |  test-server-1     |
|  us-east-2a  |  i-027552a73f021f3bd  |  test-server-2     |
+--------------+-----------------------+--------------------+
```

範例 11：描述分區放置群組中的執行個體

下列 describe-instances 範例會描述指定的執行個體。輸出包含執行個體的放置資料，其中包括執行個體的放置群組名稱和分區號碼。

```
aws ec2 describe-instances \
    --instance-ids i-0123a456700123456 \
    --query "Reservations[*].Instances[*].Placement"
```

輸出：

```
[
    [
        {
            "AvailabilityZone": "us-east-1c",
```

```
            "GroupName": "HDFS-Group-A",
            "PartitionNumber": 3,
            "Tenancy": "default"
        }

    ]
]
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的描述放置群組中的執行個體。

範例 12：篩選為具有指定放置群組和分區號碼的執行個體

下列 describe-instances 範例會將結果篩選為僅顯示具有指定放置群組和分割區號碼的執行個體。

```
aws ec2 describe-instances \
    --filters "Name=placement-group-name,Values=HDFS-Group-A" "Name=placement-partition-number,Values=7"
```

以下內容僅顯示輸出的相關資訊。

```
"Instances": [
    {
        "InstanceId": "i-0123a456700123456",
        "InstanceType": "r4.large",
        "Placement": {
            "AvailabilityZone": "us-east-1c",
            "GroupName": "HDFS-Group-A",
            "PartitionNumber": 7,
            "Tenancy": "default"
        }
    },
    {
        "InstanceId": "i-9876a543210987654",
        "InstanceType": "r4.large",
        "Placement": {
            "AvailabilityZone": "us-east-1c",
            "GroupName": "HDFS-Group-A",
            "PartitionNumber": 7,
            "Tenancy": "default"
        }
    ],
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的描述放置群組中的執行個體。

範例 13：篩選出設定為允許從執行個體中繼資料存取標籤的執行個體

下列 describe-instances 範例會將結果篩選為僅顯示設定為允許從執行個體中繼資料存取執行個體標籤的執行個體。

```
aws ec2 describe-instances \
    --filters "Name=metadata-options.instance-metadata-tags,Values=enabled" \
    --query "Reservations[*].Instances[*].InstanceId" \
    --output text
```

以下內容顯示預期的輸出。

```
i-1234567890abcdefg
i-abcdefg1234567890
i-11111111aaaaaaaaa
i-aaaaaaaa111111111
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的使用執行個體中繼資料中的執行個體標籤。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeInstances。

describe-internet-gateways

以下程式碼範例顯示如何使用 describe-internet-gateways。

AWS CLI

描述網際網路閘道

下列describe-internet-gateways範例說明指定的網際網路閘道。

```
aws ec2 describe-internet-gateways \
    --internet-gateway-ids igw-0d0fb496b3EXAMPLE
```

輸出：

```
{
    "InternetGateways": [
        {
```

```
            "Attachments": [
                {
                    "State": "available",
                    "VpcId": "vpc-0a60eb65b4EXAMPLE"
                }
            ],
            "InternetGatewayId": "igw-0d0fb496b3EXAMPLE",
            "OwnerId": "123456789012",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "my-igw"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的網際網路閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInternetGateways。

describe-ipam-pools

以下程式碼範例顯示如何使用 describe-ipam-pools。

AWS CLI

檢視 IPAM 集區的詳細資訊

下列describe-ipam-pools範例顯示集區的詳細資訊。

(Linux)：

```
aws ec2 describe-ipam-pools \
    --filters Name=owner-id,Values=123456789012 Name=ipam-scope-id,Values=ipam-
scope-02fc38cd4c48e7d38
```

(Windows)：

```
aws ec2 describe-ipam-pools ^
    --filters Name=owner-id,Values=123456789012 Name=ipam-scope-id,Values=ipam-
scope-02fc38cd4c48e7d38
```

輸出：

```
{
    "IpamPools": [
        {
            "OwnerId": "123456789012",
            "IpamPoolId": "ipam-pool-02ec043a19bbe5d08",
            "IpamPoolArn": "arn:aws:ec2::123456789012:ipam-pool/ipam-
pool-02ec043a19bbe5d08",
            "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-02fc38cd4c48e7d38",
            "IpamScopeType": "private",
            "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
            "IpamRegion": "us-east-1",
            "Locale": "None",
            "PoolDepth": 1,
            "State": "create-complete",
            "AutoImport": true,
            "AddressFamily": "ipv4",
            "AllocationMinNetmaskLength": 16,
            "AllocationMaxNetmaskLength": 26,
            "AllocationDefaultNetmaskLength": 24,
            "AllocationResourceTags": [
                {
                    "Key": "Environment",
                    "Value": "Preprod"
                }
            ],
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Preprod pool"
                }
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeIpamPools。

describe-ipam-resource-discoveries

以下程式碼範例顯示如何使用 describe-ipam-resource-discoveries。

AWS CLI

範例 1：檢視資源探索的完整詳細資訊

在此範例中，您是委派 IPAM 管理員，想要建立資源探索並與另一個 AWS 組織中的 IPAM 管理員共用，以便管理員可以管理和監控組織中資源的 IP 地址。

如果出現下列情況，此範例可能很有用：

您嘗試建立資源探索，但收到錯誤，表示您已達到 1 的限制。您意識到您可能已建立資源探索，並且想要在帳戶中檢視它。您在 IPAM 未探索的區域中有資源。您想要檢視為 資源`--operating-regions`定義的 ，並確保已將正確的區域新增為操作區域，以便探索其中的資源。

下列`describe-ipam-resource-discoveries`範例列出您 AWS 帳戶中資源探索的詳細資訊。每個 AWS 區域可以有一個資源探索。

```
aws ec2 describe-ipam-resource-discoveries \
    --region us-east-1
```

輸出：

```
{
    "IpamResourceDiscoveries": [
        {
            "OwnerId": "149977607591",
            "IpamResourceDiscoveryId": "ipam-res-disco-0f8bdee9067137c0d",
            "IpamResourceDiscoveryArn": "arn:aws:ec2::149977607591:ipam-resource-
discovery/ipam-res-disco-0f8bdee9067137c0d",
            "IpamResourceDiscoveryRegion": "us-east-1",
            "OperatingRegions": [
                {
                    "RegionName": "us-east-1"
                }
            ],
            "IsDefault": false,
            "State": "create-complete",
            "Tags": []
    }
]
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的將 IPAM 與組織外部的帳戶整合。

範例 2：僅檢視資源探索 IDs

下列describe-ipam-resource-discoveries範例列出您 AWS 帳戶中資源探索的 ID。每個 AWS 區域可以有一個資源探索。

```
aws ec2 describe-ipam-resource-discoveries \
    --query "IpamResourceDiscoveries[*].IpamResourceDiscoveryId" \
    --output text
```

輸出：

```
ipam-res-disco-0481e39b242860333
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的將 IPAM 與組織外部的帳戶整合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeIpamResourceDiscoveries。

describe-ipam-resource-discovery-associations

以下程式碼範例顯示如何使用 describe-ipam-resource-discovery-associations。

AWS CLI

檢視與 IPAM 的所有資源探索關聯

在此範例中，您是 IPAM 委派的管理員，擁有與 IPAM 相關聯的資源探索，可將其他帳戶與您的 IPAM 整合。您已注意到 IPAM 未如預期般探索資源探索操作區域中的資源。您想要檢查資源探索的狀態和狀態，以確保建立資源探索的帳戶仍處於作用中狀態，且資源探索仍在共用中。

--region 必須是 IPAM 的主區域。

下列describe-ipam-resource-discovery-associations範例列出您 AWS 帳戶中的資源探索關聯。

```
aws ec2 describe-ipam-resource-discovery-associations \
    --region us-east-1
```

輸出：

```
{
```

```
    "IpamResourceDiscoveryAssociations": [
        {
            "OwnerId": "320805250157",
            "IpamResourceDiscoveryAssociationId": "ipam-res-disco-
assoc-05e6b45eca5bf5cf7",
            "IpamResourceDiscoveryAssociationArn": "arn:aws:ec2::320805250157:ipam-
resource-discovery-association/ipam-res-disco-assoc-05e6b45eca5bf5cf7",
            "IpamResourceDiscoveryId": "ipam-res-disco-0f4ef577a9f37a162",
            "IpamId": "ipam-005f921c17ebd5107",
            "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
            "IpamRegion": "us-east-1",
            "IsDefault": true,
            "ResourceDiscoveryStatus": "active",
            "State": "associate-complete",
            "Tags": []
        },
        {
            "OwnerId": "149977607591",
            "IpamResourceDiscoveryAssociationId": "ipam-res-disco-
assoc-0dfd21ae189ab5f62",
            "IpamResourceDiscoveryAssociationArn": "arn:aws:ec2::149977607591:ipam-
resource-discovery-association/ipam-res-disco-assoc-0dfd21ae189ab5f62",
            "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
            "IpamId": "ipam-005f921c17ebd5107",
            "IpamArn": "arn:aws:ec2::149977607591:ipam/ipam-005f921c17ebd5107",
            "IpamRegion": "us-east-1",
            "IsDefault": false,
            "ResourceDiscoveryStatus": "active",
            "State": "create-complete",
            "Tags": []
        }
    ]
}
```

在此範例中，執行此命令後，您注意到您有一個非預設資源探索 ("IsDefault": false ``) that is ``"ResourceDiscoveryStatus": "not-found" 和 "State": "create-complete"。資源探索擁有者的帳戶已關閉。在另一個案例中，如果您注意到 是 "ResourceDiscoveryStatus": "not-found"和 "State": "associate-complete"，這表示發生下列其中一種情況：

資源探索擁有者已刪除資源探索。資源探索擁有者未共用資源探索。

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的將 IPAM 與組織外部的帳戶整合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeIpamResourceDiscoveryAssociations。

describe-ipam-scopes

以下程式碼範例顯示如何使用 describe-ipam-scopes。

AWS CLI

檢視 IPAM 範圍的詳細資訊

下列describe-ipam-scopes範例顯示範圍的詳細資訊。

```
aws ec2 describe-ipam-scopes \
    --filters Name=owner-id,Values=123456789012 Name=ipam-
id,Values=ipam-08440e7a3acde3908
```

輸出：

```
{
    "IpamScopes": [
        {
            "OwnerId": "123456789012",
            "IpamScopeId": "ipam-scope-02fc38cd4c48e7d38",
            "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-02fc38cd4c48e7d38",
            "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
            "IpamRegion": "us-east-1",
            "IpamScopeType": "private",
            "IsDefault": true,
            "PoolCount": 2,
            "State": "create-complete",
            "Tags": []
        },
        {
            "OwnerId": "123456789012",
            "IpamScopeId": "ipam-scope-0b9eed026396dbc16",
            "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-0b9eed026396dbc16",
            "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
            "IpamRegion": "us-east-1",
            "IpamScopeType": "public",
```

```
            "IsDefault": true,
            "PoolCount": 0,
            "State": "create-complete",
            "Tags": []
        },
        {
            "OwnerId": "123456789012",
            "IpamScopeId": "ipam-scope-0f1aff29486355c22",
            "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-0f1aff29486355c22",
            "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
            "IpamRegion": "us-east-1",
            "IpamScopeType": "private",
            "IsDefault": false,
            "Description": "Example description",
            "PoolCount": 0,
            "State": "create-complete",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Example name value"
                }
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeIpamScopes。

describe-ipams

以下程式碼範例顯示如何使用 describe-ipams。

AWS CLI

檢視 IPAM 的詳細資訊

下列describe-ipams範例顯示 IPAM 的詳細資訊。

```
aws ec2 describe-ipams \
    --filters Name=owner-id,Values=123456789012
```

輸出：

```
{
    "Ipams": [
        {
            "OwnerId": "123456789012",
            "IpamId": "ipam-08440e7a3acde3908",
            "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
            "IpamRegion": "us-east-1",
            "PublicDefaultScopeId": "ipam-scope-0b9eed026396dbc16",
            "PrivateDefaultScopeId": "ipam-scope-02fc38cd4c48e7d38",
            "ScopeCount": 3,
            "OperatingRegions": [
                {
                    "RegionName": "us-east-1"
                },
                {
                    "RegionName": "us-east-2"
                },
                {
                    "RegionName": "us-west-1"
                }
            ],
            "State": "create-complete",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "ExampleIPAM"
                }
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeIpams。

describe-ipv6-pools

以下程式碼範例顯示如何使用 `describe-ipv6-pools`。

AWS CLI

描述您的 IPv6 地址集區

下列`describe-ipv6-pools`範例顯示所有 IPv6 地址集區的詳細資訊。

```
aws ec2 describe-ipv6-pools
```

輸出：

```
{
    "Ipv6Pools": [
        {
            "PoolId": "ipv6pool-ec2-012345abc12345abc",
            "PoolCidrBlocks": [
                {
                    "Cidr": "2001:db8:123::/48"
                }
            ],
            "Tags": [
                {
                    "Key": "pool-1",
                    "Value": "public"
                }
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeIpv6Pools。

describe-key-pairs

以下程式碼範例顯示如何使用 describe-key-pairs。

AWS CLI

顯示金鑰對

下列 describe-key-pairs 範例顯示指定金鑰對的相關資訊。

```
aws ec2 describe-key-pairs \
    --key-names my-key-pair
```

輸出：

```
{
    "KeyPairs": [
```

```
        {
            "KeyPairId": "key-0b94643da6EXAMPLE",
            "KeyFingerprint":
 "1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f",
            "KeyName": "my-key-pair",
            "KeyType": "rsa",
            "Tags": [],
            "CreateTime": "2022-05-27T21:51:16.000Z"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的描述公有金鑰。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeKeyPairs。

describe-launch-template-versions

以下程式碼範例顯示如何使用 describe-launch-template-versions。

AWS CLI

描述啟動範本版本

此範例說明指定啟動範本的版本。

命令：

```
aws ec2 describe-launch-template-versions --launch-template-id lt-068f72b72934aff71
```

輸出：

```
{
  "LaunchTemplateVersions": [
      {
          "LaunchTemplateId": "lt-068f72b72934aff71",
          "LaunchTemplateName": "Webservers",
          "VersionNumber": 3,
          "CreatedBy": "arn:aws:iam::123456789102:root",
          "LaunchTemplateData": {
              "KeyName": "kp-us-east",
              "ImageId": "ami-6057e21a",
              "InstanceType": "t2.small",
```

```
            "NetworkInterfaces": [
                {
                    "SubnetId": "subnet-7b16de0c",
                    "DeviceIndex": 0,
                    "Groups": [
                        "sg-7c227019"
                    ]
                }
            ]
        },
        "DefaultVersion": false,
        "CreateTime": "2017-11-20T13:19:54.000Z"
    },
    {
        "LaunchTemplateId": "lt-068f72b72934aff71",
        "LaunchTemplateName": "Webservers",
        "VersionNumber": 2,
        "CreatedBy": "arn:aws:iam::123456789102:root",
        "LaunchTemplateData": {
            "KeyName": "kp-us-east",
            "ImageId": "ami-6057e21a",
            "InstanceType": "t2.medium",
            "NetworkInterfaces": [
                {
                    "SubnetId": "subnet-1a2b3c4d",
                    "DeviceIndex": 0,
                    "Groups": [
                        "sg-7c227019"
                    ]
                }
            ]
        },
        "DefaultVersion": false,
        "CreateTime": "2017-11-20T13:12:32.000Z"
    },
    {
        "LaunchTemplateId": "lt-068f72b72934aff71",
        "LaunchTemplateName": "Webservers",
        "VersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789102:root",
        "LaunchTemplateData": {
            "UserData": "",
            "KeyName": "kp-us-east",
            "ImageId": "ami-aabbcc11",
```

```
                "InstanceType": "t2.medium",
                "NetworkInterfaces": [
                    {
                        "SubnetId": "subnet-7b16de0c",
                        "DeviceIndex": 0,
                        "DeleteOnTermination": false,
                        "Groups": [
                            "sg-7c227019"
                        ],
                        "AssociatePublicIpAddress": true
                    }
                ]
            },
            "DefaultVersion": true,
            "CreateTime": "2017-11-20T12:52:33.000Z"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLaunchTemplateVersions。

describe-launch-templates

以下程式碼範例顯示如何使用 describe-launch-templates。

AWS CLI

描述啟動範本

此範例說明您的啟動範本。

命令：

```
aws ec2 describe-launch-templates
```

輸出：

```
{
    "LaunchTemplates": [
        {
            "LatestVersionNumber": 2,
            "LaunchTemplateId": "lt-0e06d290751193123",
```

```
            "LaunchTemplateName": "TemplateForWebServer",
            "DefaultVersionNumber": 2,
            "CreatedBy": "arn:aws:iam::123456789012:root",
            "CreateTime": "2017-11-27T09:30:23.000Z"
        },
        {
            "LatestVersionNumber": 6,
            "LaunchTemplateId": "lt-0c45b5e061ec98456",
            "LaunchTemplateName": "DBServersTemplate",
            "DefaultVersionNumber": 1,
            "CreatedBy": "arn:aws:iam::123456789012:root",
            "CreateTime": "2017-11-20T09:25:22.000Z"
        },
        {
            "LatestVersionNumber": 1,
            "LaunchTemplateId": "lt-0d47d774e8e52dabc",
            "LaunchTemplateName": "MyLaunchTemplate2",
            "DefaultVersionNumber": 1,
            "CreatedBy": "arn:aws:iam::123456789012:root",
            "CreateTime": "2017-11-02T12:06:21.000Z"
        },
        {
            "LatestVersionNumber": 3,
            "LaunchTemplateId": "lt-01e5f948eb4f589d6",
            "LaunchTemplateName": "testingtemplate2",
            "DefaultVersionNumber": 1,
            "CreatedBy": "arn:aws:sts::123456789012:assumed-role/AdminRole/
i-03ee35176e2e5aabc",
            "CreateTime": "2017-12-01T08:19:48.000Z"
        },
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLaunchTemplates。

describe-local-gateway-route-table-virtual-interface-group-associations

以下程式碼範例顯示如何使用 describe-local-gateway-route-table-virtual-interface-group-associations。

AWS CLI

描述虛擬介面群組與本機閘道路由表之間的關聯

下列describe-local-gateway-route-table-virtual-interface-group-associations範例說明虛擬介面群組與 AWS 帳戶中本機閘道路由表之間的關聯。

```
aws ec2 describe-local-gateway-route-table-virtual-interface-group-associations
```

輸出：

```
{
    "LocalGatewayRouteTableVirtualInterfaceGroupAssociations": [
        {
            "LocalGatewayRouteTableVirtualInterfaceGroupAssociationId": "lgw-vif-grp-assoc-07145b276bEXAMPLE",
            "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
            "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
            "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
            "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:123456789012:local-gateway-route-table/lgw-rtb-059615ef7dEXAMPLE",
            "OwnerId": "123456789012",
            "State": "associated",
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Outposts 使用者指南中的使用本機閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLocalGatewayRouteTableVirtualInterfaceGroupAssociations。

describe-local-gateway-route-table-vpc-associations

以下程式碼範例顯示如何使用 describe-local-gateway-route-table-vpc-associations。

AWS CLI

描述 VPCs與本機閘道路由表之間的關聯

下列describe-local-gateway-route-table-vpc-associations範例顯示 VPCs 與本機閘道路由表之間指定關聯的相關資訊。

```
aws ec2 describe-local-gateway-route-table-vpc-associations \
    --local-gateway-route-table-vpc-association-ids lgw-vpc-assoc-0e0f27af15EXAMPLE
```

輸出：

```
{
    "LocalGatewayRouteTableVpcAssociation": {
        "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-0e0f27af1EXAMPLE",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
        "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
        "VpcId": "vpc-0efe9bde08EXAMPLE",
        "State": "associated"
    }
}
```

如需詳細資訊，請參閱《 Outposts 使用者指南》中的本機閘道路油表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLocalGatewayRouteTableVpcAssociations。

describe-local-gateway-route-tables

以下程式碼範例顯示如何使用 describe-local-gateway-route-tables。

AWS CLI

描述本機閘道路由表

下列describe-local-gateway-route-tables範例顯示本機閘道路由表的詳細資訊。

```
aws ec2 describe-local-gateway-route-tables
```

輸出：

```
{
    "LocalGatewayRouteTables": [
        {
            "LocalGatewayRouteTableId": "lgw-rtb-059615ef7deEXAMPLE",
            "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
            "OutpostArn": "arn:aws:outposts:us-west-2:111122223333:outpost/
op-0dc11b66edEXAMPLE",
            "State": "available"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLocalGatewayRouteTables。

describe-local-gateway-virtual-interface-groups

以下程式碼範例顯示如何使用 describe-local-gateway-virtual-interface-groups。

AWS CLI

描述本機閘道虛擬介面群組

下列describe-local-gateway-virtual-interface-groups範例說明您 AWS 帳戶中的本機閘道虛擬介面群組。

```
aws ec2 describe-local-gateway-virtual-interface-groups
```

輸出：

```
{
    "LocalGatewayVirtualInterfaceGroups": [
        {
            "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
            "LocalGatewayVirtualInterfaceIds": [
                "lgw-vif-01a23bc4d5EXAMPLE",
                "lgw-vif-543ab21012EXAMPLE"
            ],
            "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
            "OwnerId": "123456789012",
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Outposts 使用者指南中的使用本機閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLocalGatewayVirtualInterfaceGroups。

describe-local-gateway-virtual-interfaces

以下程式碼範例顯示如何使用 describe-local-gateway-virtual-interfaces。

AWS CLI

描述本機閘道虛擬介面

下列describe-local-gateway-virtual-interfaces範例說明您 AWS 帳戶中的本機閘道虛擬介面。

```
aws ec2 describe-local-gateway-virtual-interfaces
```

輸出：

```
{
    "LocalGatewayVirtualInterfaces": [
        {
            "LocalGatewayVirtualInterfaceId": "lgw-vif-01a23bc4d5EXAMPLE",
            "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
            "Vlan": 2410,
            "LocalAddress": "0.0.0.0/0",
            "PeerAddress": "0.0.0.0/0",
            "LocalBgpAsn": 65010,
            "PeerBgpAsn": 65000,
            "OwnerId": "123456789012",
            "Tags": []
        },
        {
            "LocalGatewayVirtualInterfaceId": "lgw-vif-543ab21012EXAMPLE",
            "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
            "Vlan": 2410,
            "LocalAddress": "0.0.0.0/0",
            "PeerAddress": "0.0.0.0/0",
            "LocalBgpAsn": 65010,
            "PeerBgpAsn": 65000,
            "OwnerId": "123456789012",
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Outposts 使用者指南中的使用本機閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLocalGatewayVirtualInterfaces。

describe-local-gateways

以下程式碼範例顯示如何使用 describe-local-gateways。

AWS CLI

描述您的本機閘道

下列describe-local-gateways範例顯示可供您使用的本機閘道詳細資訊。

```
aws ec2 describe-local-gateways
```

輸出：

```
{
    "LocalGateways": [
        {
            "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
            "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-0dc11b66ed59f995a",
            "OwnerId": "123456789012",
            "State": "available"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLocalGateways。

describe-locked-snapshots

以下程式碼範例顯示如何使用 describe-locked-snapshots。

AWS CLI

描述快照的鎖定狀態

下列describe-locked-snapshots範例說明指定快照的鎖定狀態。

```
aws ec2 describe-locked-snapshots \
    --snapshot-ids snap-0b5e733b4a8df6e0d
```

輸出：

```
{
    "Snapshots": [
        {
            "OwnerId": "123456789012",
            "SnapshotId": "snap-0b5e733b4a8df6e0d",
            "LockState": "governance",
            "LockDuration": 365,
            "LockCreatedOn": "2024-05-05T00:56:06.208000+00:00",
            "LockDurationStartTime": "2024-05-05T00:56:06.208000+00:00",
            "LockExpiresOn": "2025-05-05T00:56:06.208000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EBS 使用者指南》中的快照鎖定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLockedSnapshots。

describe-managed-prefix-lists

以下程式碼範例顯示如何使用 describe-managed-prefix-lists。

AWS CLI

描述受管字首清單

下列describe-managed-prefix-lists範例說明 AWS 帳戶 擁有的字首清單123456789012。

```
aws ec2 describe-managed-prefix-lists \
    --filters Name=owner-id,Values=123456789012
```

輸出：

```
{
    "PrefixLists": [
        {
            "PrefixListId": "pl-11223344556677aab",
            "AddressFamily": "IPv6",
            "State": "create-complete",
            "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-11223344556677aab",
```

```
            "PrefixListName": "vpc-ipv6-cidrs",
            "MaxEntries": 25,
            "Version": 1,
            "Tags": [],
            "OwnerId": "123456789012"
        },
        {
            "PrefixListId": "pl-0123456abcabcabc1",
            "AddressFamily": "IPv4",
            "State": "active",
            "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-0123456abcabcabc1",
            "PrefixListName": "vpc-cidrs",
            "MaxEntries": 10,
            "Version": 1,
            "Tags": [],
            "OwnerId": "123456789012"
      }
  ]
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的受管字首清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeManagedPrefixLists。

describe-moving-addresses

以下程式碼範例顯示如何使用 describe-moving-addresses。

AWS CLI

描述您的移動地址

此範例說明所有移動中的彈性 IP 地址。

命令：

```
aws ec2 describe-moving-addresses
```

輸出：

```
{
  "MovingAddressStatuses": [
```

```
    {
      "PublicIp": "198.51.100.0",
      "MoveStatus": "MovingToVpc"
    }
  ]
}
```

此範例說明要移至 EC2-VPC 平台的所有地址。

命令：

```
aws ec2 describe-moving-addresses --filters Name=moving-status,Values=MovingToVpc
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeMovingAddresses。

describe-nat-gateways

以下程式碼範例顯示如何使用 describe-nat-gateways。

AWS CLI

範例 1：描述公有 NAT 閘道

下列describe-nat-gateways範例說明指定的公有 NAT 閘道。

```
aws ec2 describe-nat-gateways \
    --nat-gateway-id nat-01234567890abcdef
```

輸出：

```
{
    "NatGateways": [
        {
            "CreateTime": "2023-08-25T01:56:51.000Z",
            "NatGatewayAddresses": [
                {
                    "AllocationId": "eipalloc-0790180cd2EXAMPLE",
                    "NetworkInterfaceId": "eni-09cc4b2558794f7f9",
                    "PrivateIp": "10.0.0.211",
                    "PublicIp": "54.85.121.213",
                    "AssociationId": "eipassoc-04d295cc9b8815b24",
                    "IsPrimary": true,
```

```
                    "Status": "succeeded"
                },
                {
                    "AllocationId": "eipalloc-0be6ecac95EXAMPLE",
                    "NetworkInterfaceId": "eni-09cc4b2558794f7f9",
                    "PrivateIp": "10.0.0.74",
                    "PublicIp": "3.211.231.218",
                    "AssociationId": "eipassoc-0f96bdca17EXAMPLE",
                    "IsPrimary": false,
                    "Status": "succeeded"
                }
            ],
            "NatGatewayId": "nat-01234567890abcdef",
            "State": "available",
            "SubnetId": "subnet-655eab5f08EXAMPLE",
            "VpcId": "vpc-098eb5ef58EXAMPLE",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "public-nat"
                }
            ],
            "ConnectivityType": "public"
        }
    ]
}
```

範例 2：描述私有 NAT 閘道

下列describe-nat-gateways範例說明指定的私有 NAT 閘道。

```
aws ec2 describe-nat-gateways \
    --nat-gateway-id nat-1234567890abcdef0
```

輸出：

```
{
    "NatGateways": [
        {
            "CreateTime": "2023-08-25T00:50:05.000Z",
            "NatGatewayAddresses": [
                {
                    "NetworkInterfaceId": "eni-0065a61b324d1897a",
```

```
                    "PrivateIp": "10.0.20.240",
                    "IsPrimary": true,
                    "Status": "succeeded"
                },
                {
                    "NetworkInterfaceId": "eni-0065a61b324d1897a",
                    "PrivateIp": "10.0.20.33",
                    "IsPrimary": false,
                    "Status": "succeeded"
                },
                {
                    "NetworkInterfaceId": "eni-0065a61b324d1897a",
                    "PrivateIp": "10.0.20.197",
                    "IsPrimary": false,
                    "Status": "succeeded"
                }
            ],
            "NatGatewayId": "nat-1234567890abcdef0",
            "State": "available",
            "SubnetId": "subnet-08fc749671EXAMPLE",
            "VpcId": "vpc-098eb5ef58EXAMPLE",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "private-nat"
                }
            ],
            "ConnectivityType": "private"
        }
    ]
}
```

如需詳細資訊，請參閱 《Amazon VPC 使用者指南》中的 NAT 閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeNatGateways。

`describe-network-acls`

以下程式碼範例顯示如何使用 `describe-network-acls`。

AWS CLI

描述您的網路 ACLs

下列describe-network-acls範例會擷取網路 ACLs的詳細資訊。

```
aws ec2 describe-network-acls
```

輸出：

```
{
    "NetworkAcls": [
        {
            "Associations": [
                {
                    "NetworkAclAssociationId": "aclassoc-0c1679dc41EXAMPLE",
                    "NetworkAclId": "acl-0ea1f54ca7EXAMPLE",
                    "SubnetId": "subnet-0931fc2fa5EXAMPLE"
                }
            ],
            "Entries": [
                {
                    "CidrBlock": "0.0.0.0/0",
                    "Egress": true,
                    "Protocol": "-1",
                    "RuleAction": "allow",
                    "RuleNumber": 100
                },
                {
                    "CidrBlock": "0.0.0.0/0",
                    "Egress": true,
                    "Protocol": "-1",
                    "RuleAction": "deny",
                    "RuleNumber": 32767
                },
                {
                    "CidrBlock": "0.0.0.0/0",
                    "Egress": false,
                    "Protocol": "-1",
                    "RuleAction": "allow",
                    "RuleNumber": 100
                },
                {
                    "CidrBlock": "0.0.0.0/0",
                    "Egress": false,
                    "Protocol": "-1",
                    "RuleAction": "deny",
```

```
                "RuleNumber": 32767
            }
        ],
        "IsDefault": true,
        "NetworkAclId": "acl-0ea1f54ca7EXAMPLE",
        "Tags": [],
        "VpcId": "vpc-06e4ab6c6cEXAMPLE",
        "OwnerId": "111122223333"
    },
    {
        "Associations": [],
        "Entries": [
            {
                "CidrBlock": "0.0.0.0/0",
                "Egress": true,
                "Protocol": "-1",
                "RuleAction": "allow",
                "RuleNumber": 100
            },
            {
                "Egress": true,
                "Ipv6CidrBlock": "::/0",
                "Protocol": "-1",
                "RuleAction": "allow",
                "RuleNumber": 101
            },
            {
                "CidrBlock": "0.0.0.0/0",
                "Egress": true,
                "Protocol": "-1",
                "RuleAction": "deny",
                "RuleNumber": 32767
            },
            {
                "Egress": true,
                "Ipv6CidrBlock": "::/0",
                "Protocol": "-1",
                "RuleAction": "deny",
                "RuleNumber": 32768
            },
            {
                "CidrBlock": "0.0.0.0/0",
                "Egress": false,
                "Protocol": "-1",
```

```
                    "RuleAction": "allow",
                    "RuleNumber": 100
                },
                {
                    "Egress": false,
                    "Ipv6CidrBlock": "::/0",
                    "Protocol": "-1",
                    "RuleAction": "allow",
                    "RuleNumber": 101
                },
                {
                    "CidrBlock": "0.0.0.0/0",
                    "Egress": false,
                    "Protocol": "-1",
                    "RuleAction": "deny",
                    "RuleNumber": 32767
                },
                {
                    "Egress": false,
                    "Ipv6CidrBlock": "::/0",
                    "Protocol": "-1",
                    "RuleAction": "deny",
                    "RuleNumber": 32768
                }
            ],
            "IsDefault": true,
            "NetworkAclId": "acl-0e2a78e4e2EXAMPLE",
            "Tags": [],
            "VpcId": "vpc-03914afb3eEXAMPLE",
            "OwnerId": "111122223333"
        }
    ]
}
```

如需詳細資訊，請參閱AWS 《VPC 使用者指南》中的網路 ACLs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeNetworkAcls。

describe-network-insights-access-scope-analyses

以下程式碼範例顯示如何使用 describe-network-insights-access-scope-analyses。

AWS CLI

描述 Network Insights 存取範圍分析

下列`describe-network-insights-access-scope-analyses`範例說明您 AWS 帳戶中的存取範圍分析。

```
aws ec2 describe-network-insights-access-scope-analyses \
    --region us-east-1
```

輸出：

```
{
    "NetworkInsightsAccessScopeAnalyses": [
        {
            "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789111",
            "NetworkInsightsAccessScopeAnalysisArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope-analysis/nisa-123456789111",
            "NetworkInsightsAccessScopeId": "nis-123456789222",
            "Status": "succeeded",
            "StartDate": "2022-01-25T19:45:36.842000+00:00",
            "FindingsFound": "true",
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱《Network Access Analyzer 指南》中的使用 AWS CLI 開始使用 Network Access Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeNetworkInsightsAccessScopeAnalyses。

`describe-network-insights-access-scopes`

以下程式碼範例顯示如何使用 `describe-network-insights-access-scopes`。

AWS CLI

描述 Network Insights 存取範圍

下列describe-network-insights-access-scopes範例說明您 AWS 帳戶中的存取範圍分析。

```
aws ec2 describe-network-insights-access-scopes \
    --region us-east-1
```

輸出：

```
{
    "NetworkInsightsAccessScopes": [
        {
            "NetworkInsightsAccessScopeId": "nis-123456789111",
            "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope/nis-123456789111",
            "CreatedDate": "2021-11-29T21:12:41.416000+00:00",
            "UpdatedDate": "2021-11-29T21:12:41.416000+00:00",
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱《Network Access Analyzer 指南》中的使用 AWS CLI 開始使用 Network Access Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeNetworkInsightsAccessScopes。

describe-network-insights-analyses

以下程式碼範例顯示如何使用 describe-network-insights-analyses。

AWS CLI

檢視路徑分析的結果

下列describe-network-insights-analyses範例說明指定的分析。在此範例中，來源是網際網路閘道，目的地是 EC2 執行個體，而通訊協定是 TCP。分析成功 (Status 為 succeeded)，且路徑無法連線 (NetworkPathFound 為 false)。說明程式碼ENI_SG_RULES_MISMATCH表示執行個體的安全群組不包含允許目的地連接埠流量的規則。

```
aws ec2 describe-network-insights-analyses \
```

```
    --network-insights-analysis-ids nia-02207aa13eb480c7a
```

輸出：

```
{
    "NetworkInsightsAnalyses": [
        {
            "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a",
            "NetworkInsightsAnalysisArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-analysis/nia-02207aa13eb480c7a",
            "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",
            "StartDate": "2021-01-20T22:58:37.495Z",
            "Status": "succeeded",
            "NetworkPathFound": false,
            "Explanations": [
                {
                    "Direction": "ingress",
                    "ExplanationCode": "ENI_SG_RULES_MISMATCH",
                    "NetworkInterface": {
                        "Id": "eni-0a25edef15a6cc08c",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:network-
interface/eni-0a25edef15a6cc08c"
                    },
                    "SecurityGroups": [
                        {
                            "Id": "sg-02f0d35a850ba727f",
                            "Arn": "arn:aws:ec2:us-east-1:123456789012:security-
group/sg-02f0d35a850ba727f"
                        }
                    ],
                    "Subnet": {
                        "Id": "subnet-004ff41eccb4d1194",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:subnet/
subnet-004ff41eccb4d1194"
                    },
                    "Vpc": {
                        "Id": "vpc-f1663d98ad28331c7",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-
f1663d98ad28331c7"
                    }
                }
            ],
            "Tags": []
```

```
        }
    ]
}
```

如需詳細資訊，請參閱 Reachability Analyzer 指南中的開始使用 AWS CLI。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeNetworkInsightsAnalyses。

`describe-network-insights-paths`

以下程式碼範例顯示如何使用 `describe-network-insights-paths`。

AWS CLI

描述路徑

下列`describe-network-insights-paths`範例說明指定的路徑。

```
aws ec2 describe-network-insights-paths \
    --network-insights-path-ids nip-0b26f224f1d131fa8
```

輸出：

```
{
    "NetworkInsightsPaths": [
        {
            "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",
            "NetworkInsightsPathArn": "arn:aws:ec2:us-east-1:123456789012:network-
insights-path/nip-0b26f224f1d131fa8",
            "CreatedDate": "2021-01-20T22:43:46.933Z",
            "Source": "igw-0797cccdc9d73b0e5",
            "Destination": "i-0495d385ad28331c7",
            "Protocol": "tcp"
        }
    ]
}
```

如需詳細資訊，請參閱 Reachability Analyzer 指南中的開始使用 AWS CLI。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeNetworkInsightsPaths。

describe-network-interface-attribute

以下程式碼範例顯示如何使用 describe-network-interface-attribute。

AWS CLI

描述網路界面的附件屬性

此範例命令說明指定網路界面的attachment屬性。

命令：

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --attribute attachment
```

輸出：

```
{
  "NetworkInterfaceId": "eni-686ea200",
  "Attachment": {
      "Status": "attached",
      "DeviceIndex": 0,
      "AttachTime": "2015-05-21T20:02:20.000Z",
      "InstanceId": "i-1234567890abcdef0",
      "DeleteOnTermination": true,
      "AttachmentId": "eni-attach-43348162",
      "InstanceOwnerId": "123456789012"
  }
}
```

描述網路界面的描述屬性

此範例命令說明指定網路界面的description屬性。

命令：

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --attribute description
```

輸出：

```
{
```

```
  "NetworkInterfaceId": "eni-686ea200",
  "Description": {
      "Value": "My description"
  }
}
```

描述網路界面的 groupSet 屬性

此範例命令說明指定網路界面的groupSet屬性。

命令：

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --
attribute groupSet
```

輸出：

```
{
  "NetworkInterfaceId": "eni-686ea200",
  "Groups": [
      {
          "GroupName": "my-security-group",
          "GroupId": "sg-903004f8"
      }
  ]
}
```

描述網路界面的 sourceDestCheck 屬性

此範例命令說明指定網路界面的sourceDestCheck屬性。

命令：

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --
attribute sourceDestCheck
```

輸出：

```
{
  "NetworkInterfaceId": "eni-686ea200",
  "SourceDestCheck": {
      "Value": true
```

```
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeNetworkInterfaceAttribute。

describe-network-interface-permissions

以下程式碼範例顯示如何使用 describe-network-interface-permissions。

AWS CLI

描述您的網路界面許可

此範例說明您的所有網路介面許可。

命令：

```
aws ec2 describe-network-interface-permissions
```

輸出：

```
{
  "NetworkInterfacePermissions": [
      {
          "PermissionState": {
              "State": "GRANTED"
          },
          "NetworkInterfacePermissionId": "eni-perm-06fd19020ede149ea",
          "NetworkInterfaceId": "eni-b909511a",
          "Permission": "INSTANCE-ATTACH",
          "AwsAccountId": "123456789012"
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeNetworkInterfacePermissions。

describe-network-interfaces

以下程式碼範例顯示如何使用 describe-network-interfaces。

AWS CLI

描述您的網路界面

此範例說明您的所有網路介面。

命令：

```
aws ec2 describe-network-interfaces
```

輸出：

```
{
  "NetworkInterfaces": [
      {
          "Status": "in-use",
          "MacAddress": "02:2f:8f:b0:cf:75",
          "SourceDestCheck": true,
          "VpcId": "vpc-a01106c2",
          "Description": "my network interface",
          "Association": {
              "PublicIp": "203.0.113.12",
              "AssociationId": "eipassoc-0fbb766a",
              "PublicDnsName": "ec2-203-0-113-12.compute-1.amazonaws.com",
              "IpOwnerId": "123456789012"
          },
          "NetworkInterfaceId": "eni-e5aa89a3",
          "PrivateIpAddresses": [
              {
                  "PrivateDnsName": "ip-10-0-1-17.ec2.internal",
                  "Association": {
                      "PublicIp": "203.0.113.12",
                      "AssociationId": "eipassoc-0fbb766a",
                      "PublicDnsName": "ec2-203-0-113-12.compute-1.amazonaws.com",
                      "IpOwnerId": "123456789012"
                  },
                  "Primary": true,
                  "PrivateIpAddress": "10.0.1.17"
              }
          ],
          "RequesterManaged": false,
          "Ipv6Addresses": [],
          "PrivateDnsName": "ip-10-0-1-17.ec2.internal",
          "AvailabilityZone": "us-east-1d",
```

```
            "Attachment": {
                "Status": "attached",
                "DeviceIndex": 1,
                "AttachTime": "2013-11-30T23:36:42.000Z",
                "InstanceId": "i-1234567890abcdef0",
                "DeleteOnTermination": false,
                "AttachmentId": "eni-attach-66c4350a",
                "InstanceOwnerId": "123456789012"
            },
            "Groups": [
                {
                    "GroupName": "default",
                    "GroupId": "sg-8637d3e3"
                }
            ],
            "SubnetId": "subnet-b61f49f0",
            "OwnerId": "123456789012",
            "TagSet": [],
            "PrivateIpAddress": "10.0.1.17"
        },
        {
            "Status": "in-use",
            "MacAddress": "02:58:f5:ef:4b:06",
            "SourceDestCheck": true,
            "VpcId": "vpc-a01106c2",
            "Description": "Primary network interface",
            "Association": {
                "PublicIp": "198.51.100.0",
                "IpOwnerId": "amazon"
            },
            "NetworkInterfaceId": "eni-f9ba99bf",
            "PrivateIpAddresses": [
                {
                    "Association": {
                        "PublicIp": "198.51.100.0",
                        "IpOwnerId": "amazon"
                    },
                    "Primary": true,
                    "PrivateIpAddress": "10.0.1.149"
                }
            ],
            "RequesterManaged": false,
            "Ipv6Addresses": [],
            "AvailabilityZone": "us-east-1d",
```

```
            "Attachment": {
                "Status": "attached",
                "DeviceIndex": 0,
                "AttachTime": "2013-11-30T23:35:33.000Z",
                "InstanceId": "i-0598c7d356eba48d7",
                "DeleteOnTermination": true,
                "AttachmentId": "eni-attach-1b9db777",
                "InstanceOwnerId": "123456789012"
            },
            "Groups": [
                {
                    "GroupName": "default",
                    "GroupId": "sg-8637d3e3"
                }
            ],
            "SubnetId": "subnet-b61f49f0",
            "OwnerId": "123456789012",
            "TagSet": [],
            "PrivateIpAddress": "10.0.1.149"
        }
    ]
}
```

此範例說明具有索引鍵為 Purpose且值為 之標籤的網路介面Prod。

命令：

```
aws ec2 describe-network-interfaces --filters Name=tag:Purpose,Values=Prod
```

輸出：

```
{
    "NetworkInterfaces": [
        {
            "Status": "available",
            "MacAddress": "12:2c:bd:f9:bf:17",
            "SourceDestCheck": true,
            "VpcId": "vpc-8941ebec",
            "Description": "ProdENI",
            "NetworkInterfaceId": "eni-b9a5ac93",
            "PrivateIpAddresses": [
                {
                    "PrivateDnsName": "ip-10-0-1-55.ec2.internal",
```

```
                "Primary": true,
                "PrivateIpAddress": "10.0.1.55"
            },
            {
                "PrivateDnsName": "ip-10-0-1-117.ec2.internal",
                "Primary": false,
                "PrivateIpAddress": "10.0.1.117"
            }
        ],
        "RequesterManaged": false,
        "PrivateDnsName": "ip-10-0-1-55.ec2.internal",
        "AvailabilityZone": "us-east-1d",
        "Ipv6Addresses": [],
        "Groups": [
            {
                "GroupName": "MySG",
                "GroupId": "sg-905002f5"
            }
        ],
        "SubnetId": "subnet-31d6c219",
        "OwnerId": "123456789012",
        "TagSet": [
            {
                "Value": "Prod",
                "Key": "Purpose"
            }
        ],
        "PrivateIpAddress": "10.0.1.55"
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeNetworkInterfaces。

describe-placement-groups

以下程式碼範例顯示如何使用 describe-placement-groups。

AWS CLI

描述您的置放群組

此範例命令說明所有置放群組。

命令：

```
aws ec2 describe-placement-groups
```

輸出：

```
{
    "PlacementGroups": [
        {
            "GroupName": "my-cluster",
            "State": "available",
            "Strategy": "cluster"
        },
        ...
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePlacementGroups。

describe-prefix-lists

以下程式碼範例顯示如何使用 describe-prefix-lists。

AWS CLI

描述字首清單

此範例列出區域的所有可用字首清單。

命令：

```
aws ec2 describe-prefix-lists
```

輸出：

```
{
  "PrefixLists": [
    {
      "PrefixListName": "com.amazonaws.us-east-1.s3",
      "Cidrs": [
        "54.231.0.0/17"
```

```
      ],
      "PrefixListId": "pl-63a5400a"
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePrefixLists。

`describe-principal-id-format`

以下程式碼範例顯示如何使用 `describe-principal-id-format`。

AWS CLI

描述已啟用長 ID 格式之 IAM 使用者和角色的 ID 格式

下列`describe-principal-id-format`範例說明根使用者的 ID 格式、所有 IAM 角色，以及啟用長 ID 格式的所有 IAM 使用者。

```
aws ec2 describe-principal-id-format \
    --resource instance
```

輸出：

```
{
    "Principals": [
        {
            "Arn": "arn:aws:iam::123456789012:root",
            "Statuses": [
                {
                    "Deadline": "2016-12-15T00:00:00.000Z",
                    "Resource": "reservation",
                    "UseLongIds": true
                },
                {
                    "Deadline": "2016-12-15T00:00:00.000Z",
                    "Resource": "instance",
                    "UseLongIds": true
                },
                {
                    "Deadline": "2016-12-15T00:00:00.000Z",
                    "Resource": "volume",
```

```
                    "UseLongIds": true
                },
            ]
        },
        ...
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePrincipalIdFormat。

describe-public-ipv4-pools

以下程式碼範例顯示如何使用 describe-public-ipv4-pools。

AWS CLI

描述您的公有 IPv4 地址集區

下列describe-public-ipv4-pools範例顯示使用自攜 IP 地址 (BYOIP) 佈建公有 IPv4 地址範圍時所建立之地址集區的詳細資訊。

```
aws ec2 describe-public-ipv4-pools
```

輸出：

```
{
    "PublicIpv4Pools": [
        {
            "PoolId": "ipv4pool-ec2-1234567890abcdef0",
            "PoolAddressRanges": [
                {
                    "FirstAddress": "203.0.113.0",
                    "LastAddress": "203.0.113.255",
                    "AddressCount": 256,
                    "AvailableAddressCount": 256
                }
            ],
            "TotalAddressCount": 256,
            "TotalAvailableAddressCount": 256
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePublicIpv4Pools。

describe-regions

以下程式碼範例顯示如何使用 describe-regions。

AWS CLI

範例 1：描述所有已啟用的區域

以下 describe-regions 範例說明為您帳戶啟用的所有區域。

```
aws ec2 describe-regions
```

輸出：

```
{
    "Regions": [
        {
            "Endpoint": "ec2.eu-north-1.amazonaws.com",
            "RegionName": "eu-north-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-south-1.amazonaws.com",
            "RegionName": "ap-south-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-west-3.amazonaws.com",
            "RegionName": "eu-west-3",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-west-2.amazonaws.com",
            "RegionName": "eu-west-2",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-west-1.amazonaws.com",
            "RegionName": "eu-west-1",
            "OptInStatus": "opt-in-not-required"
        },
```

```
        {
            "Endpoint": "ec2.ap-northeast-3.amazonaws.com",
            "RegionName": "ap-northeast-3",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-northeast-2.amazonaws.com",
            "RegionName": "ap-northeast-2",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-northeast-1.amazonaws.com",
            "RegionName": "ap-northeast-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.sa-east-1.amazonaws.com",
            "RegionName": "sa-east-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ca-central-1.amazonaws.com",
            "RegionName": "ca-central-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-southeast-1.amazonaws.com",
            "RegionName": "ap-southeast-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-southeast-2.amazonaws.com",
            "RegionName": "ap-southeast-2",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-central-1.amazonaws.com",
            "RegionName": "eu-central-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.us-east-1.amazonaws.com",
            "RegionName": "us-east-1",
            "OptInStatus": "opt-in-not-required"
```

```
        },
        {
            "Endpoint": "ec2.us-east-2.amazonaws.com",
            "RegionName": "us-east-2",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.us-west-1.amazonaws.com",
            "RegionName": "us-west-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.us-west-2.amazonaws.com",
            "RegionName": "us-west-2",
            "OptInStatus": "opt-in-not-required"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的區域 (Region) 和區域 (Zone)。

範例 2：說明使用名稱包含特定字串的端點，且已啟用的區域

下列 `describe-regions` 範例會描述您已啟用，且其端點中具有字串「美國」(us) 的所有區域。

```
aws ec2 describe-regions \
    --filters "Name=endpoint,Values=*us*"
```

輸出：

```
{
    "Regions": [
        {
            "Endpoint": "ec2.us-east-1.amazonaws.com",
            "RegionName": "us-east-1"
        },
        {
            "Endpoint": "ec2.us-east-2.amazonaws.com",
            "RegionName": "us-east-2"
        },
        {
            "Endpoint": "ec2.us-west-1.amazonaws.com",
            "RegionName": "us-west-1"
```

```
        },
        {
            "Endpoint": "ec2.us-west-2.amazonaws.com",
            "RegionName": "us-west-2"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的區域 (Region) 和區域 (Zone)。

範例 3：描述所有區域

下列 `describe-regions` 範例會描述所有可用的區域，包括已停用的區域。

```
aws ec2 describe-regions \
    --all-regions
```

輸出：

```
{
    "Regions": [
        {
            "Endpoint": "ec2.eu-north-1.amazonaws.com",
            "RegionName": "eu-north-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-south-1.amazonaws.com",
            "RegionName": "ap-south-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-west-3.amazonaws.com",
            "RegionName": "eu-west-3",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-west-2.amazonaws.com",
            "RegionName": "eu-west-2",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-west-1.amazonaws.com",
```

```
            "RegionName": "eu-west-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-northeast-3.amazonaws.com",
            "RegionName": "ap-northeast-3",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.me-south-1.amazonaws.com",
            "RegionName": "me-south-1",
            "OptInStatus": "not-opted-in"
        },
        {
            "Endpoint": "ec2.ap-northeast-2.amazonaws.com",
            "RegionName": "ap-northeast-2",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-northeast-1.amazonaws.com",
            "RegionName": "ap-northeast-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.sa-east-1.amazonaws.com",
            "RegionName": "sa-east-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ca-central-1.amazonaws.com",
            "RegionName": "ca-central-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-east-1.amazonaws.com",
            "RegionName": "ap-east-1",
            "OptInStatus": "not-opted-in"
        },
        {
            "Endpoint": "ec2.ap-southeast-1.amazonaws.com",
            "RegionName": "ap-southeast-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
```

```
            "Endpoint": "ec2.ap-southeast-2.amazonaws.com",
            "RegionName": "ap-southeast-2",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-central-1.amazonaws.com",
            "RegionName": "eu-central-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.us-east-1.amazonaws.com",
            "RegionName": "us-east-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.us-east-2.amazonaws.com",
            "RegionName": "us-east-2",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.us-west-1.amazonaws.com",
            "RegionName": "us-west-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.us-west-2.amazonaws.com",
            "RegionName": "us-west-2",
            "OptInStatus": "opt-in-not-required"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的區域 (Region) 和區域 (Zone)。

範例 4：僅列出區域名稱

下列 `describe-regions` 範例會使用 `--query` 參數來篩選輸出，並以文字形式僅傳回區域 (Region) 的名稱。

```
aws ec2 describe-regions \
    --all-regions \
    --query "Regions[].{Name:RegionName}" \
    --output text
```

輸出：

```
eu-north-1
ap-south-1
eu-west-3
eu-west-2
eu-west-1
ap-northeast-3
ap-northeast-2
me-south-1
ap-northeast-1
sa-east-1
ca-central-1
ap-east-1
ap-southeast-1
ap-southeast-2
eu-central-1
us-east-1
us-east-2
us-west-1
us-west-2
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的區域 (Region) 和區域 (Zone)。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeRegions。

describe-replace-root-volume-tasks

以下程式碼範例顯示如何使用 describe-replace-root-volume-tasks。

AWS CLI

範例 1：檢視特定根磁碟區替換任務的相關資訊

下列describe-replace-root-volume-tasks範例說明根磁碟區取代任務 replacevol-0111122223333abcd。

```
aws ec2 describe-replace-root-volume-tasks \
    --replace-root-volume-task-ids replacevol-0111122223333abcd
```

輸出：

```
{
```

```
    "ReplaceRootVolumeTasks": [
        {
            "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",
            "Tags": [],
            "InstanceId": "i-0123456789abcdefa",
            "TaskState": "succeeded",
            "StartTime": "2022-03-14T15:16:28Z",
            "CompleteTime": "2022-03-14T15:16:52Z"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的取代根磁碟區。

範例 2：檢視特定執行個體所有根磁碟區替換任務的相關資訊

下列`describe-replace-root-volume-tasks`範例說明執行個體 i-0123456789abcdefa 的所有根磁碟區替換任務。

```
aws ec2 describe-replace-root-volume-tasks \
    --filters Name=instance-id,Values=i-0123456789abcdefa
```

輸出：

```
{
    "ReplaceRootVolumeTasks": [
        {
            "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",
            "Tags": [],
            "InstanceId": "i-0123456789abcdefa",
            "TaskState": "succeeded",
            "StartTime": "2022-03-14T15:06:38Z",
            "CompleteTime": "2022-03-14T15:07:03Z"
        },
        {
            "ReplaceRootVolumeTaskId": "replacevol-0444455555555abcd",
            "Tags": [],
            "InstanceId": "i-0123456789abcdefa",
            "TaskState": "succeeded",
            "StartTime": "2022-03-14T15:16:28Z",
            "CompleteTime": "2022-03-14T15:16:52Z"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的取代根磁碟區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReplaceRootVolumeTasks。

describe-reserved-instances-listings

以下程式碼範例顯示如何使用 describe-reserved-instances-listings。

AWS CLI

描述預留執行個體清單

下列describe-reserved-instances-listings範例會擷取指定預留執行個體清單的相關資訊。

```
aws ec2 describe-reserved-instances-listings \
    --reserved-instances-listing-id 5ec28771-05ff-4b9b-aa31-9e57dexample
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReservedInstancesListings。

describe-reserved-instances-modifications

以下程式碼範例顯示如何使用 describe-reserved-instances-modifications。

AWS CLI

描述預留執行個體修改

此範例命令說明為您的帳戶提交的所有預留執行個體修改請求。

命令：

```
aws ec2 describe-reserved-instances-modifications
```

輸出：

```
{
```

```
    "ReservedInstancesModifications": [
        {
            "Status": "fulfilled",
            "ModificationResults": [
                {
                    "ReservedInstancesId": "93bbbca2-62f1-4d9d-b225-16bada29e6c7",
                    "TargetConfiguration": {
                        "AvailabilityZone": "us-east-1b",
                        "InstanceType": "m1.large",
                        "InstanceCount": 3
                    }
                },
                {
                     "ReservedInstancesId": "1ba8e2e3-aabb-46c3-bcf5-3fe2fda922e6",
                     "TargetConfiguration": {
                         "AvailabilityZone": "us-east-1d",
                         "InstanceType": "m1.xlarge",
                         "InstanceCount": 1
                     }
                 }
            ],
            "EffectiveDate": "2015-08-12T17:00:00.000Z",
            "CreateDate": "2015-08-12T17:52:52.630Z",
            "UpdateDate": "2015-08-12T18:08:06.698Z",
            "ClientToken": "c9adb218-3222-4889-8216-0cf0e52dc37e:
            "ReservedInstancesModificationId": "rimod-d3ed4335-b1d3-4de6-
ab31-0f13aaf46687",
            "ReservedInstancesIds": [
                {
                    "ReservedInstancesId": "b847fa93-e282-4f55-b59a-1342f5bd7c02"
                }
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReservedInstancesModifications。

describe-reserved-instances-offerings

以下程式碼範例顯示如何使用 describe-reserved-instances-offerings。

AWS CLI

描述預留執行個體方案

此範例命令說明所有可在 區域中購買的預留執行個體。

命令：

```
aws ec2 describe-reserved-instances-offerings
```

輸出：

```
{
  "ReservedInstancesOfferings": [
      {
          "OfferingType": "Partial Upfront",
          "AvailabilityZone": "us-east-1b",
          "InstanceTenancy": "default",
          "PricingDetails": [],
          "ProductDescription": "Red Hat Enterprise Linux",
          "UsagePrice": 0.0,
          "RecurringCharges": [
              {
                  "Amount": 0.088,
                  "Frequency": "Hourly"
              }
          ],
          "Marketplace": false,
          "CurrencyCode": "USD",
          "FixedPrice": 631.0,
          "Duration": 94608000,
          "ReservedInstancesOfferingId": "9a06095a-bdc6-47fe-a94a-2a382f016040",
          "InstanceType": "c1.medium"
      },
      {
          "OfferingType": "PartialUpfront",
          "AvailabilityZone": "us-east-1b",
          "InstanceTenancy": "default",
          "PricingDetails": [],
          "ProductDescription": "Linux/UNIX",
          "UsagePrice": 0.0,
          "RecurringCharges": [
              {
```

```
                "Amount": 0.028,
                "Frequency": "Hourly"
            }
        ],
        "Marketplace": false,
        "CurrencyCode": "USD",
        "FixedPrice": 631.0,
        "Duration": 94608000,
        "ReservedInstancesOfferingId": "bfbefc6c-0d10-418d-b144-7258578d329d",
        "InstanceType": "c1.medium"
    },
  ...
}
```

使用 選項描述您的預留執行個體方案

此範例列出 提供的預留執行個體 AWS ，其規格如下：t1.micro 執行個體類型、Windows (Amazon VPC) 產品和重度使用率產品。

命令：

```
aws ec2 describe-reserved-instances-offerings --no-include-marketplace --instance-type t1.micro --product-description "Windows (Amazon VPC)" --offering-type "no upfront"
```

輸出：

```
{
    "ReservedInstancesOfferings": [
      {
        "OfferingType": "No Upfront",
        "AvailabilityZone": "us-east-1b",
        "InstanceTenancy": "default",
        "PricingDetails": [],
        "ProductDescription": "Windows",
        "UsagePrice": 0.0,
        "RecurringCharges": [
            {
                "Amount": 0.015,
                "Frequency": "Hourly"
            }
        ],
        "Marketplace": false,
```

```
            "CurrencyCode": "USD",
            "FixedPrice": 0.0,
            "Duration": 31536000,
            "ReservedInstancesOfferingId": "c48ab04c-fe69-4f94-8e39-a23842292823",
            "InstanceType": "t1.micro"
        },

            ...
        {
            "OfferingType": "No Upfront",
            "AvailabilityZone": "us-east-1d",
            "InstanceTenancy": "default",
            "PricingDetails": [],
            "ProductDescription": "Windows (Amazon VPC)",
            "UsagePrice": 0.0,
            "RecurringCharges": [
                {
                    "Amount": 0.015,
                    "Frequency": "Hourly"
                }
            ],
            "Marketplace": false,
            "CurrencyCode": "USD",
            "FixedPrice": 0.0,
            "Duration": 31536000,
            "ReservedInstancesOfferingId": "3a98bf7d-2123-42d4-b4f5-8dbec4b06dc6",
            "InstanceType": "t1.micro"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReservedInstancesOfferings。

describe-reserved-instances

以下程式碼範例顯示如何使用 describe-reserved-instances。

AWS CLI

描述您的預留執行個體

此範例命令說明您擁有的預留執行個體。

命令：

```
aws ec2 describe-reserved-instances
```

輸出：

```
{
  "ReservedInstances": [
      {
          "ReservedInstancesId": "b847fa93-e282-4f55-b59a-1342fexample",
          "OfferingType": "No Upfront",
          "AvailabilityZone": "us-west-1c",
          "End": "2016-08-14T21:34:34.000Z",
          "ProductDescription": "Linux/UNIX",
          "UsagePrice": 0.00,
          "RecurringCharges": [
              {
                  "Amount": 0.104,
                  "Frequency": "Hourly"
              }
          ],
          "Start": "2015-08-15T21:34:35.086Z",
          "State": "active",
          "FixedPrice": 0.0,
          "CurrencyCode": "USD",
          "Duration": 31536000,
          "InstanceTenancy": "default",
          "InstanceType": "m3.medium",
          "InstanceCount": 2
      },
      ...
  ]
}
```

使用篩選條件描述您的預留執行個體

此範例會篩選回應，以在 us-west-1c 中僅包含三年的 t2.micro Linux/UNIX 預留執行個體。

命令：

```
aws ec2 describe-reserved-instances --
filters Name=duration,Values=94608000 Name=instance-
```

```
type,Values=t2.micro Name=product-description,Values=Linux/UNIX Name=availability-
zone,Values=us-east-1e
```

輸出：

```
{
    "ReservedInstances": [
        {
            "ReservedInstancesId": "f127bd27-edb7-44c9-a0eb-0d7e09259af0",
            "OfferingType": "All Upfront",
            "AvailabilityZone": "us-east-1e",
            "End": "2018-03-26T21:34:34.000Z",
            "ProductDescription": "Linux/UNIX",
            "UsagePrice": 0.00,
            "RecurringCharges": [],
            "Start": "2015-03-27T21:34:35.848Z",
            "State": "active",
            "FixedPrice": 151.0,
            "CurrencyCode": "USD",
            "Duration": 94608000,
            "InstanceTenancy": "default",
            "InstanceType": "t2.micro",
            "InstanceCount": 1
        }
    ]
}
```

如需詳細資訊，請參閱《AWS 命令行介面使用者指南》中的「Amazon EC2 執行個體」。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReservedInstances。

describe-route-tables

以下程式碼範例顯示如何使用 describe-route-tables。

AWS CLI

描述您的路由表

下列describe-route-tables範例會擷取路由表的詳細資訊

```
aws ec2 describe-route-tables
```

輸出：

```
{
    "RouteTables": [
        {
            "Associations": [
                {
                    "Main": true,
                    "RouteTableAssociationId": "rtbassoc-0df3f54e06EXAMPLE",
                    "RouteTableId": "rtb-09ba434c1bEXAMPLE"
                }
            ],
            "PropagatingVgws": [],
            "RouteTableId": "rtb-09ba434c1bEXAMPLE",
            "Routes": [
                {
                    "DestinationCidrBlock": "10.0.0.0/16",
                    "GatewayId": "local",
                    "Origin": "CreateRouteTable",
                    "State": "active"
                },
                {
                    "DestinationCidrBlock": "0.0.0.0/0",
                    "NatGatewayId": "nat-06c018cbd8EXAMPLE",
                    "Origin": "CreateRoute",
                    "State": "blackhole"
                }
            ],
            "Tags": [],
            "VpcId": "vpc-0065acced4EXAMPLE",
            "OwnerId": "111122223333"
        },
        {
            "Associations": [
                {
                    "Main": true,
                    "RouteTableAssociationId": "rtbassoc-9EXAMPLE",
                    "RouteTableId": "rtb-a1eec7de"
                }
            ],
            "PropagatingVgws": [],
            "RouteTableId": "rtb-a1eec7de",
            "Routes": [
                {
```

```
                "DestinationCidrBlock": "172.31.0.0/16",
                "GatewayId": "local",
                "Origin": "CreateRouteTable",
                "State": "active"
            },
            {
                "DestinationCidrBlock": "0.0.0.0/0",
                "GatewayId": "igw-fEXAMPLE",
                "Origin": "CreateRoute",
                "State": "active"
            }
        ],
        "Tags": [],
        "VpcId": "vpc-3EXAMPLE",
        "OwnerId": "111122223333"
    },
    {
        "Associations": [
            {
                "Main": false,
                "RouteTableAssociationId": "rtbassoc-0b100c28b2EXAMPLE",
                "RouteTableId": "rtb-07a98f76e5EXAMPLE",
                "SubnetId": "subnet-0d3d002af8EXAMPLE"
            }
        ],
        "PropagatingVgws": [],
        "RouteTableId": "rtb-07a98f76e5EXAMPLE",
        "Routes": [
            {
                "DestinationCidrBlock": "10.0.0.0/16",
                "GatewayId": "local",
                "Origin": "CreateRouteTable",
                "State": "active"
            },
            {
                "DestinationCidrBlock": "0.0.0.0/0",
                "GatewayId": "igw-06cf664d80EXAMPLE",
                "Origin": "CreateRoute",
                "State": "active"
            }
        ],
        "Tags": [],
        "VpcId": "vpc-0065acced4EXAMPLE",
        "OwnerId": "111122223333"
```

```
        }
    ]
}
```

如需詳細資訊，請參閱《AWS VPC 使用者指南》中的使用路由表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRouteTables。

describe-scheduled-instance-availability

以下程式碼範例顯示如何使用 describe-scheduled-instance-availability。

AWS CLI

描述可用的排程

此範例說明從指定日期開始，每週週日發生的排程。

命令：

```
aws ec2 describe-scheduled-instance-availability --
recurrence Frequency=Weekly,Interval=1,OccurrenceDays=[1] --first-slot-start-time-
range EarliestTime=2016-01-31T00:00:00Z,LatestTime=2016-01-31T04:00:00Z
```

輸出：

```
{
  "ScheduledInstanceAvailabilitySet": [
    {
        "AvailabilityZone": "us-west-2b",
        "TotalScheduledInstanceHours": 1219,
        "PurchaseToken": "eyJ2IjoiMSIsInMiOjEsImMiOi...",
        "MinTermDurationInDays": 366,
        "AvailableInstanceCount": 20,
        "Recurrence": {
            "OccurrenceDaySet": [
                1
            ],
            "Interval": 1,
            "Frequency": "Weekly",
            "OccurrenceRelativeToEnd": false
        },
        "Platform": "Linux/UNIX",
```

```
        "FirstSlotStartTime": "2016-01-31T00:00:00Z",
        "MaxTermDurationInDays": 366,
        "SlotDurationInHours": 23,
        "NetworkPlatform": "EC2-VPC",
        "InstanceType": "c4.large",
        "HourlyPrice": "0.095"
    },
    ...
  ]
}
```

若要縮小結果範圍，您可以新增篩選條件來指定作業系統、網路和執行個體類型。

命令：

--filters Name=platform，Values=Linux/UNIX Name=network-platform，Values=EC2-VPC Name=instance-type，Values=c4.large

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeScheduledInstanceAvailability。

describe-scheduled-instances

以下程式碼範例顯示如何使用 describe-scheduled-instances。

AWS CLI

描述您的排程執行個體

此範例說明指定的排程執行個體。

命令：

```
aws ec2 describe-scheduled-instances --scheduled-instance-
ids sci-1234-1234-1234-1234-123456789012
```

輸出：

```
{
  "ScheduledInstanceSet": [
      {
          "AvailabilityZone": "us-west-2b",
          "ScheduledInstanceId": "sci-1234-1234-1234-1234-123456789012",
```

```
            "HourlyPrice": "0.095",
            "CreateDate": "2016-01-25T21:43:38.612Z",
            "Recurrence": {
                "OccurrenceDaySet": [
                    1
                ],
                "Interval": 1,
                "Frequency": "Weekly",
                "OccurrenceRelativeToEnd": false,
                "OccurrenceUnit": ""
            },
            "Platform": "Linux/UNIX",
            "TermEndDate": "2017-01-31T09:00:00Z",
            "InstanceCount": 1,
            "SlotDurationInHours": 32,
            "TermStartDate": "2016-01-31T09:00:00Z",
            "NetworkPlatform": "EC2-VPC",
            "TotalScheduledInstanceHours": 1696,
            "NextSlotStartTime": "2016-01-31T09:00:00Z",
            "InstanceType": "c4.large"
        }
    ]
}
```

此範例說明所有已排程的執行個體。

命令：

```
aws ec2 describe-scheduled-instances
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeScheduledInstances。

describe-security-group-references

以下程式碼範例顯示如何使用 describe-security-group-references。

AWS CLI

描述安全群組參考

此範例說明 的安全群組參考sg-bbbb2222。回應表示 VPC 中的安全群組sg-bbbb2222正在參考安全群組vpc-aaaaaaaa。

命令：

```
aws ec2 describe-security-group-references --group-id sg-bbbbb22222
```

輸出：

```
{
  "SecurityGroupsReferenceSet": [
    {
      "ReferencingVpcId": "vpc-aaaaaaaa ",
      "GroupId": "sg-bbbbb22222",
      "VpcPeeringConnectionId": "pcx-b04deed9"
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSecurityGroupReferences。

describe-security-group-rules

以下程式碼範例顯示如何使用 describe-security-group-rules。

AWS CLI

範例 1：描述安全群組的安全群組規則

下列describe-security-group-rules範例說明指定安全群組的安全群組規則。使用filters選項將結果範圍限定為特定安全群組。

```
aws ec2 describe-security-group-rules \
    --filters Name="group-id",Values="sg-1234567890abcdef0"
```

輸出：

```
{
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-abcdef01234567890",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "111122223333",
            "IsEgress": false,
            "IpProtocol": "-1",
```

```
            "FromPort": -1,
            "ToPort": -1,
            "ReferencedGroupInfo": {
                "GroupId": "sg-1234567890abcdef0",
                "UserId": "111122223333"
            },
            "Tags": []
        },
        {
            "SecurityGroupRuleId": "sgr-bcdef01234567890a",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "111122223333",
            "IsEgress": true,
            "IpProtocol": "-1",
            "FromPort": -1,
            "ToPort": -1,
            "CidrIpv6": "::/0",
            "Tags": []
        },
        {
            "SecurityGroupRuleId": "sgr-cdef01234567890ab",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "111122223333",
            "IsEgress": true,
            "IpProtocol": "-1",
            "FromPort": -1,
            "ToPort": -1,
            "CidrIpv4": "0.0.0.0/0",
            "Tags": []
        }
    ]
}
```

範例 2：描述安全群組規則

下列describe-security-group-rules範例說明指定的安全群組規則。

```
aws ec2 describe-security-group-rules \
    --security-group-rule-ids sgr-cdef01234567890ab
```

輸出：

```
{
```

```
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-cdef01234567890ab",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "111122223333",
            "IsEgress": true,
            "IpProtocol": "-1",
            "FromPort": -1,
            "ToPort": -1,
            "CidrIpv4": "0.0.0.0/0",
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的安全群組規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSecurityGroupRules。

describe-security-group-vpc-associations

以下程式碼範例顯示如何使用 describe-security-group-vpc-associations。

AWS CLI

描述 VPC 關聯

下列describe-security-group-vpc-associations範例說明指定安全群組的 VPC 關聯。

```
aws ec2 describe-security-group-vpc-associations \
    --filters Name=group-id,Values=sg-04dbb43907d3f8a78
```

輸出：

```
{
    "SecurityGroupVpcAssociations": [
        {
            "GroupId": "sg-04dbb43907d3f8a78",
            "VpcId": "vpc-0bf4c2739bc05a694",
            "VpcOwnerId": "123456789012",
            "State": "associated"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon VPCs》中的將安全群組與多個 VPC 建立關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSecurityGroupVpcAssociations。

describe-security-groups

以下程式碼範例顯示如何使用 describe-security-groups。

AWS CLI

範例 1：描述安全群組

下列 describe-security-groups 範例會描述指定的安全群組。

```
aws ec2 describe-security-groups \
    --group-ids sg-903004f8
```

輸出：

```
{
    "SecurityGroups": [
        {
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "UserIdGroupPairs": [],
                    "PrefixListIds": []
                }
            ],
            "Description": "My security group",
            "Tags": [
                {
                    "Value": "SG1",
                    "Key": "Name"
```

```
                }
            ],
            "IpPermissions": [
                {
                    "IpProtocol": "-1",
                    "IpRanges": [],
                    "UserIdGroupPairs": [
                        {
                            "UserId": "123456789012",
                            "GroupId": "sg-903004f8"
                        }
                    ],
                    "PrefixListIds": []
                },
                {
                    "PrefixListIds": [],
                    "FromPort": 22,
                    "IpRanges": [
                        {
                            "Description": "Access from NY office",
                            "CidrIp": "203.0.113.0/24"
                        }
                    ],
                    "ToPort": 22,
                    "IpProtocol": "tcp",
                    "UserIdGroupPairs": []
                    }
            ],
            "GroupName": "MySecurityGroup",
            "VpcId": "vpc-1a2b3c4d",
            "OwnerId": "123456789012",
            "GroupId": "sg-903004f8",
        }
    ]
}
```

範例 2：描述具有特定規則的安全群組

下列`describe-security-groups`範例使用篩選條件，將結果範圍限定為具有允許 SSH 流量（連接埠 22) 之規則的安全群組，以及允許來自所有地址 () 之流量的規則`0.0.0.0/0`。此範例使用 `--query` 參數，僅顯示安全群組的名稱。安全群組必須符合所有篩選條件才能在結果中傳回；不過，單一規則不需要符合所有篩選條件。例如，輸出會傳回一個安全群組，其中包含允許來自特定 IP 地址之 SSH 流量的一個規則，以及允許來自所有地址之 HTTP 流量的另一個規則。

```
aws ec2 describe-security-groups \
    --filters Name=ip-permission.from-port,Values=22 Name=ip-permission.to-
port,Values=22 Name=ip-permission.cidr,Values='0.0.0.0/0' \
    --query "SecurityGroups[*].[GroupName]" \
    --output text
```

輸出：

```
default
my-security-group
web-servers
launch-wizard-1
```

範例 3：根據標籤描述安全群組

下列 describe-security-groups 範例會使用篩選條件，將結果範圍限定為在安全群組名稱中加入 test，且具有標籤 Test=To-delete 的安全群組。此範例使用 --query 參數，僅顯示安全群組的名稱和 ID。

```
aws ec2 describe-security-groups \
    --filters Name=group-name,Values=*test* Name=tag:Test,Values=To-delete \
    --query "SecurityGroups[*].{Name:GroupName,ID:GroupId}"
```

輸出：

```
[
    {
        "Name": "testfornewinstance",
        "ID": "sg-33bb22aa"
    },
    {
        "Name": "newgrouptest",
        "ID": "sg-1a2b3c4d"
    }
]
```

如需使用標籤篩選條件的其他範例，請參閱《Amazon EC2 使用者指南》中的使用標籤。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeSecurityGroups。

describe-snapshot-attribute

以下程式碼範例顯示如何使用 describe-snapshot-attribute。

AWS CLI

描述快照的快照屬性

下列describe-snapshot-attribute範例列出與快照共用的帳戶。

```
aws ec2 describe-snapshot-attribute \
    --snapshot-id snap-01234567890abcedf \
    --attribute createVolumePermission
```

輸出：

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "CreateVolumePermissions": [
        {
            "UserId": "123456789012"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的共用 Amazon EBS 快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSnapshotAttribute。

describe-snapshot-tier-status

以下程式碼範例顯示如何使用 describe-snapshot-tier-status。

AWS CLI

檢視封存快照的封存資訊

下列describe-snapshot-tier-status範例提供封存快照的封存資訊。

```
aws ec2 describe-snapshot-tier-status \
```

```
    --filters "Name=snapshot-id, Values=snap-01234567890abcedf"
```

輸出：

```
{
    "SnapshotTierStatuses": [
        {
            "Status": "completed",
            "ArchivalCompleteTime": "2021-09-15T17:33:16.147Z",
            "LastTieringProgress": 100,
            "Tags": [],
            "VolumeId": "vol-01234567890abcedf",
            "LastTieringOperationState": "archival-completed",
            "StorageTier": "archive",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-01234567890abcedf",
            "LastTieringStartTime": "2021-09-15T16:44:37.574Z"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的檢視封存的快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSnapshotTierStatus。

describe-snapshots

以下程式碼範例顯示如何使用 describe-snapshots。

AWS CLI

範例 1：描述快照

下列 describe-snapshots 範例會描述指定的快照。

```
aws ec2 describe-snapshots \
    --snapshot-ids snap-1234567890abcdef0
```

輸出：

```
{
```

```
    "Snapshots": [
        {
            "Description": "This is my snapshot",
            "Encrypted": false,
            "VolumeId": "vol-049df61146c4d7901",
            "State": "completed",
            "VolumeSize": 8,
            "StartTime": "2019-02-28T21:28:32.000Z",
            "Progress": "100%",
            "OwnerId": "012345678910",
            "SnapshotId": "snap-01234567890abcdef",
            "Tags": [
                {
                    "Key": "Stack",
                    "Value": "test"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Amazon EBS 加密。

範例 2：根據篩選條件描述快照

下列describe-snapshots範例使用篩選條件，將結果範圍限定為 AWS 您的帳戶擁有且pending處於 狀態的快照。此範例使用 --query 參數，僅顯示快照 ID 和快照啟動時間。

```
aws ec2 describe-snapshots \
    --owner-ids self \
    --filters Name=status,Values=pending \
    --query "Snapshots[*].{ID:SnapshotId,Time:StartTime}"
```

輸出：

```
[
    {
        "ID": "snap-1234567890abcdef0",
        "Time": "2019-08-04T12:48:18.000Z"
    },
    {
        "ID": "snap-066877671789bd71b",
```

```
        "Time": "2019-08-04T02:45:16.000Z
    },
    ...
]
```

下列 describe-snapshots 範例會使用篩選條件，將結果範圍限制為從指定磁碟區建立的快照。此範例使用 --query 參數，僅顯示快照 ID。

```
aws ec2 describe-snapshots \
    --filters Name=volume-id,Values=049df61146c4d7901 \
    --query "Snapshots[*].[SnapshotId]" \
    --output text
```

輸出：

```
snap-1234567890abcdef0
snap-08637175a712c3fb9
...
```

如需使用篩選條件的其他範例，請參閱《Amazon EC2 使用者指南》中的列出與篩選您的資源。

範例 3：根據標籤描述快照

下列 describe-snapshots 範例會使用標籤篩選條件，將結果範圍設定為具有標籤 Stack=Prod 的快照。

```
aws ec2 describe-snapshots \
    --filters Name=tag:Stack,Values=prod
```

如需 describe-snapshots 的輸出範例，請參閱範例 1。

如需使用標籤篩選條件的其他範例，請參閱《Amazon EC2 使用者指南》中的使用標籤。

範例 4：根據年齡描述快照

下列describe-snapshots範例使用 JMESPath 表達式來描述 AWS 帳戶在指定日期之前建立的所有快照。此範例僅顯示快照 ID。

```
aws ec2 describe-snapshots \
    --owner-ids 012345678910 \
```

```
    --query "Snapshots[?(StartTime<='2020-03-31')].[SnapshotId]"
```

如需使用篩選條件的其他範例，請參閱《Amazon EC2 使用者指南》中的列出與篩選您的資源。

範例 5：僅檢視封存的快照

以下 describe-snapshots 範例只列出儲存在封存層中的快照。

```
aws ec2 describe-snapshots \
    --filters "Name=storage-tier,Values=archive"
```

輸出：

```
{
    "Snapshots": [
        {
            "Description": "Snap A",
            "Encrypted": false,
            "VolumeId": "vol-01234567890aaaaaa",
            "State": "completed",
            "VolumeSize": 8,
            "StartTime": "2021-09-07T21:00:00.000Z",
            "Progress": "100%",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-01234567890aaaaaa",
            "StorageTier": "archive",
            "Tags": []
        },
    ]
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的檢視封存的快照。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeSnapshots。

describe-spot-datafeed-subscription

以下程式碼範例顯示如何使用 describe-spot-datafeed-subscription。

AWS CLI

描述帳戶的 Spot 執行個體資料饋送訂閱

此範例命令說明 帳戶的資料饋送。

命令：

```
aws ec2 describe-spot-datafeed-subscription
```

輸出：

```
{
    "SpotDatafeedSubscription": {
        "OwnerId": "123456789012",
        "Prefix": "spotdata",
        "Bucket": "amzn-s3-demo-bucket",
        "State": "Active"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSpotDatafeedSubscription。

describe-spot-fleet-instances

以下程式碼範例顯示如何使用 describe-spot-fleet-instances。

AWS CLI

描述與 Spot 機群相關聯的 Spot 執行個體

此範例命令會列出與指定 Spot 機群相關聯的 Spot 執行個體。

命令：

```
aws ec2 describe-spot-fleet-instances --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

輸出：

```
{
  "ActiveInstances": [
      {
          "InstanceId": "i-1234567890abcdef0",
```

```
            "InstanceType": "m3.medium",
            "SpotInstanceRequestId": "sir-08b93456"
        },
        ...
    ],
    "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSpotFleetInstances。

describe-spot-fleet-request-history

以下程式碼範例顯示如何使用 describe-spot-fleet-request-history。

AWS CLI

描述 Spot 機群歷史記錄

此範例命令會從指定的時間開始傳回指定 Spot 機群的歷史記錄。

命令：

```
aws ec2 describe-spot-fleet-request-history --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --start-time 2015-05-26T00:00:00Z
```

下列範例輸出顯示 Spot 機群成功啟動兩個 Spot 執行個體。

輸出：

```
{
    "HistoryRecords": [
        {
            "Timestamp": "2015-05-26T23:17:20.697Z",
            "EventInformation": {
                "EventSubType": "submitted"
            },
            "EventType": "fleetRequestChange"
        },
        {
            "Timestamp": "2015-05-26T23:17:20.873Z",
            "EventInformation": {
                "EventSubType": "active"
```

```
            },
            "EventType": "fleetRequestChange"
        },
        {
            "Timestamp": "2015-05-26T23:21:21.712Z",
            "EventInformation": {
                "InstanceId": "i-1234567890abcdef0",
                "EventSubType": "launched"
            },
            "EventType": "instanceChange"
        },
        {
            "Timestamp": "2015-05-26T23:21:21.816Z",
            "EventInformation": {
                "InstanceId": "i-1234567890abcdef1",
                "EventSubType": "launched"
            },
            "EventType": "instanceChange"
        }
    ],
    "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
    "NextToken": "CpHNsscimcV5oH7bSbub03CI2Qms5+ypNpNm
+53MNlR0YcXAkp0xFlfKf91yVxSExmbtma3awYxMFzNA663ZskT0AHtJ6TCb2Z8bQC2EnZgyELbymtWPfpZ1ZbauVg
+P+TfGlWxWWB/Vr5dk5d4LfdgA/DRAHUrYgxzrEXAMPLE=",
    "StartTime": "2015-05-26T00:00:00Z"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSpotFleetRequestHistory。

describe-spot-fleet-requests

以下程式碼範例顯示如何使用 describe-spot-fleet-requests。

AWS CLI

描述您的 Spot 機群請求

此範例說明所有 Spot 機群請求。

命令：

```
aws ec2 describe-spot-fleet-requests
```

輸出：

```
{
  "SpotFleetRequestConfigs": [
      {
          "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
          "SpotFleetRequestConfig": {
              "TargetCapacity": 20,
              "LaunchSpecifications": [
                  {
                      "EbsOptimized": false,
                      "NetworkInterfaces": [
                          {
                              "SubnetId": "subnet-a61dafcf",
                              "DeviceIndex": 0,
                              "DeleteOnTermination": false,
                              "AssociatePublicIpAddress": true,
                              "SecondaryPrivateIpAddressCount": 0
                          }
                      ],
                      "InstanceType": "cc2.8xlarge",
                      "ImageId": "ami-1a2b3c4d"
                  },
                  {
                      "EbsOptimized": false,
                      "NetworkInterfaces": [
                          {
                              "SubnetId": "subnet-a61dafcf",
                              "DeviceIndex": 0,
                              "DeleteOnTermination": false,
                              "AssociatePublicIpAddress": true,
                              "SecondaryPrivateIpAddressCount": 0
                          }
                      ],
                      "InstanceType": "r3.8xlarge",
                      "ImageId": "ami-1a2b3c4d"
                  }
              ],
              "SpotPrice": "0.05",
              "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role"
          },
          "SpotFleetRequestState": "active"
      },
      {
```

```
            "SpotFleetRequestId": "sfr-306341ed-9739-402e-881b-ce47bEXAMPLE",
            "SpotFleetRequestConfig": {
                "TargetCapacity": 20,
                "LaunchSpecifications": [
                    {
                        "EbsOptimized": false,
                        "NetworkInterfaces": [
                            {
                                "SubnetId": "subnet-6e7f829e",
                                "DeviceIndex": 0,
                                "DeleteOnTermination": false,
                                "AssociatePublicIpAddress": true,
                                "SecondaryPrivateIpAddressCount": 0
                            }
                        ],
                        "InstanceType": "m3.medium",
                        "ImageId": "ami-1a2b3c4d"
                    }
                ],
                "SpotPrice": "0.05",
                "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role"
            },
            "SpotFleetRequestState": "active"
        }
    ]
}
```

描述 Spot 機群請求

此範例說明指定的 Spot 機群請求。

命令：

```
aws ec2 describe-spot-fleet-requests --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

輸出：

```
{
    "SpotFleetRequestConfigs": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
```

```
            "SpotFleetRequestConfig": {
                "TargetCapacity": 20,
                "LaunchSpecifications": [
                    {
                        "EbsOptimized": false,
                        "NetworkInterfaces": [
                            {
                                "SubnetId": "subnet-a61dafcf",
                                "DeviceIndex": 0,
                                "DeleteOnTermination": false,
                                "AssociatePublicIpAddress": true,
                                "SecondaryPrivateIpAddressCount": 0
                            }
                        ],
                        "InstanceType": "cc2.8xlarge",
                        "ImageId": "ami-1a2b3c4d"
                    },
                    {
                        "EbsOptimized": false,
                        "NetworkInterfaces": [
                            {
                                "SubnetId": "subnet-a61dafcf",
                                "DeviceIndex": 0,
                                "DeleteOnTermination": false,
                                "AssociatePublicIpAddress": true,
                                "SecondaryPrivateIpAddressCount": 0
                            }
                        ],
                        "InstanceType": "r3.8xlarge",
                        "ImageId": "ami-1a2b3c4d"
                    }
                ],
                "SpotPrice": "0.05",
                "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role"
            },
            "SpotFleetRequestState": "active"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSpotFleetRequests。

describe-spot-instance-requests

以下程式碼範例顯示如何使用 describe-spot-instance-requests。

AWS CLI

範例 1：描述 Spot 執行個體請求

下列describe-spot-instance-requests範例說明指定的 Spot 執行個體請求。

```
aws ec2 describe-spot-instance-requests \
    --spot-instance-request-ids sir-08b93456
```

輸出：

```
{
    "SpotInstanceRequests": [
        {
            "CreateTime": "2018-04-30T18:14:55.000Z",
            "InstanceId": "i-1234567890abcdef1",
            "LaunchSpecification": {
                "InstanceType": "t2.micro",
                "ImageId": "ami-003634241a8fcdec0",
                "KeyName": "my-key-pair",
                "SecurityGroups": [
                    {
                        "GroupName": "default",
                        "GroupId": "sg-e38f24a7"
                    }
                ],
                "BlockDeviceMappings": [
                    {
                        "DeviceName": "/dev/sda1",
                        "Ebs": {
                            "DeleteOnTermination": true,
                            "SnapshotId": "snap-0e54a519c999adbbd",
                            "VolumeSize": 8,
                            "VolumeType": "standard",
                            "Encrypted": false
                        }
                    }
                ],
                "NetworkInterfaces": [
                    {
```

```
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "SubnetId": "subnet-049df61146c4d7901"
                    }
                ],
                "Placement": {
                    "AvailabilityZone": "us-east-2b",
                    "Tenancy": "default"
                },
                "Monitoring": {
                    "Enabled": false
                }
            },
            "LaunchedAvailabilityZone": "us-east-2b",
            "ProductDescription": "Linux/UNIX",
            "SpotInstanceRequestId": "sir-08b93456",
            "SpotPrice": "0.010000"
            "State": "active",
            "Status": {
                "Code": "fulfilled",
                "Message": "Your Spot request is fulfilled.",
                "UpdateTime": "2018-04-30T18:16:21.000Z"
            },
            "Tags": [],
            "Type": "one-time",
            "InstanceInterruptionBehavior": "terminate"
        }
    ]
}
```

範例 2：根據篩選條件描述 Spot 執行個體請求

下列`describe-spot-instance-requests`範例使用篩選條件，將結果範圍限定為指定可用區域中具有指定執行個體類型的 Spot 執行個體請求。此範例使用 `--query` 參數僅顯示執行個體 IDs。

```
aws ec2 describe-spot-instance-requests \
    --filters Name=launch.instance-type,Values=m3.medium Name=launched-availability-zone,Values=us-east-2a \
    --query "SpotInstanceRequests[*].[InstanceId]" \
    --output text
```

輸出：

```
i-057750d42936e468a
i-001efd250faaa6ffa
i-027552a73f021f3bd
...
```

如需使用篩選條件的其他範例，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的列出和篩選您的資源。

範例 3：根據標籤描述 Spot 執行個體請求

下列describe-spot-instance-requests範例使用標籤篩選條件，將結果範圍限定為具有標籤 的 Spot 執行個體請求cost-center=cc123。

```
aws ec2 describe-spot-instance-requests \
    --filters Name=tag:cost-center,Values=cc123
```

如需 describe-spot-instance-requests 的輸出範例，請參閱範例 1。

如需使用標籤篩選條件的其他範例，請參閱《Amazon EC2 使用者指南》中的使用標籤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSpotInstanceRequests。

describe-spot-price-history

以下程式碼範例顯示如何使用 describe-spot-price-history。

AWS CLI

描述 Spot 價格歷史記錄

此範例命令會傳回 1 月特定日期的 m1.xlarge 執行個體 Spot 價格歷史記錄。

命令：

```
aws ec2 describe-spot-price-history --instance-types m1.xlarge --start-
time 2014-01-06T07:08:09 --end-time 2014-01-06T08:09:10
```

輸出：

```
{
```

```
  "SpotPriceHistory": [
          {
              "Timestamp": "2014-01-06T07:10:55.000Z",
              "ProductDescription": "SUSE Linux",
              "InstanceType": "m1.xlarge",
              "SpotPrice": "0.087000",
              "AvailabilityZone": "us-west-1b"
          },
          {
              "Timestamp": "2014-01-06T07:10:55.000Z",
              "ProductDescription": "SUSE Linux",
              "InstanceType": "m1.xlarge",
              "SpotPrice": "0.087000",
              "AvailabilityZone": "us-west-1c"
          },
          {
              "Timestamp": "2014-01-06T05:42:36.000Z",
              "ProductDescription": "SUSE Linux (Amazon VPC)",
              "InstanceType": "m1.xlarge",
              "SpotPrice": "0.087000",
              "AvailabilityZone": "us-west-1a"
      },
      ...
}
```

描述 Linux/UNIX Amazon VPC 的 Spot 價格歷史記錄

此範例命令會傳回 1 月特定日期的 m1.xlarge、Linux/UNIX Amazon VPC 執行個體 Spot 價格歷史記錄。

命令：

```
aws ec2 describe-spot-price-history --instance-types m1.xlarge --product-
description "Linux/UNIX (Amazon VPC)" --start-time 2014-01-06T07:08:09 --end-
time 2014-01-06T08:09:10
```

輸出：

```
{
  "SpotPriceHistory": [
      {
          "Timestamp": "2014-01-06T04:32:53.000Z",
          "ProductDescription": "Linux/UNIX (Amazon VPC)",
```

```
            "InstanceType": "m1.xlarge",
            "SpotPrice": "0.080000",
            "AvailabilityZone": "us-west-1a"
        },
        {
            "Timestamp": "2014-01-05T11:28:26.000Z",
            "ProductDescription": "Linux/UNIX (Amazon VPC)",
            "InstanceType": "m1.xlarge",
            "SpotPrice": "0.080000",
            "AvailabilityZone": "us-west-1c"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSpotPriceHistory。

describe-stale-security-groups

以下程式碼範例顯示如何使用 describe-stale-security-groups。

AWS CLI

描述過時的 安全群組

此範例說明 的過時安全群組規則vpc-11223344。回應顯示您帳戶中的 sg-5fa68d3a 具有在對等 VPC sg-279ab042中參考的過時傳入 SSH 規則，而您sg-fe6fba9a帳戶中的 sg-5fa68d3a 具有在對等 VPC sg-ef6fba8b中參考的過時傳出 SSH 規則。

命令：

```
aws ec2 describe-stale-security-groups --vpc-id vpc-11223344
```

輸出：

```
{
  "StaleSecurityGroupSet": [
      {
          "VpcId": "vpc-11223344",
          "StaleIpPermissionsEgress": [
              {
                  "ToPort": 22,
```

```
                "FromPort": 22,
                "UserIdGroupPairs": [
                    {
                        "VpcId": "vpc-7a20e51f",
                        "GroupId": "sg-ef6fba8b",
                        "VpcPeeringConnectionId": "pcx-b04deed9",
                        "PeeringStatus": "active"
                    }
                ],
                "IpProtocol": "tcp"
            }
        ],
        "GroupName": "MySG1",
        "StaleIpPermissions": [],
        "GroupId": "sg-fe6fba9a",
        "Description": MySG1"
    },
    {
        "VpcId": "vpc-11223344",
        "StaleIpPermissionsEgress": [],
        "GroupName": "MySG2",
        "StaleIpPermissions": [
            {
                "ToPort": 22,
                "FromPort": 22,
                "UserIdGroupPairs": [
                    {
                        "VpcId": "vpc-7a20e51f",
                        "GroupId": "sg-279ab042",
                        "Description": "Access from pcx-b04deed9",
                        "VpcPeeringConnectionId": "pcx-b04deed9",
                        "PeeringStatus": "active"
                    }
                ],
                "IpProtocol": "tcp"
            }
        ],
        "GroupId": "sg-5fa68d3a",
        "Description": "MySG2"
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStaleSecurityGroups。

describe-store-image-tasks

以下程式碼範例顯示如何使用 describe-store-image-tasks。

AWS CLI

描述 AMI 存放區任務的進度

下列describe-store-image-tasks範例說明 AMI 存放區任務的進度。

```
aws ec2 describe-store-image-tasks
```

輸出：

```
{
    "StoreImageTaskResults": [
        {
            "AmiId": "ami-1234567890abcdef0",
            "Bucket": "my-ami-bucket",
            "ProgressPercentage": 17,
            "S3objectKey": "ami-1234567890abcdef0.bin",
            "StoreTaskState": "InProgress",
            "StoreTaskFailureReason": null,
            "TaskStartTime": "2022-01-01T01:01:01.001Z"
        }
    ]
}
```

如需使用 S3 存放和還原 AMI 的詳細資訊，請參閱《Amazon EC2 使用者指南》中的使用 S3 <https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/ami-store-restore.html> 存放和還原 AMI。EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStoreImageTasks。

describe-subnets

以下程式碼範例顯示如何使用 describe-subnets。

AWS CLI

範例 1：描述所有子網路

以下 describe-subnets 範例顯示子網路的詳細資訊。

```
aws ec2 describe-subnets
```

輸出：

```
{
    "Subnets": [
        {
            "AvailabilityZone": "us-east-1d",
            "AvailabilityZoneId": "use1-az2",
            "AvailableIpAddressCount": 4089,
            "CidrBlock": "172.31.80.0/20",
            "DefaultForAz": true,
            "MapPublicIpOnLaunch": false,
            "MapCustomerOwnedIpOnLaunch": true,
            "State": "available",
            "SubnetId": "subnet-0bb1c79de3EXAMPLE",
            "VpcId": "vpc-0ee975135dEXAMPLE",
            "OwnerId": "111122223333",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "CustomerOwnedIpv4Pool:": 'pool-2EXAMPLE',
            "SubnetArn": "arn:aws:ec2:us-east-2:111122223333:subnet/
subnet-0bb1c79de3EXAMPLE",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            }
        },
        {
            "AvailabilityZone": "us-east-1d",
            "AvailabilityZoneId": "use1-az2",
            "AvailableIpAddressCount": 4089,
            "CidrBlock": "172.31.80.0/20",
            "DefaultForAz": true,
            "MapPublicIpOnLaunch": true,
            "MapCustomerOwnedIpOnLaunch": false,
            "State": "available",
            "SubnetId": "subnet-8EXAMPLE",
            "VpcId": "vpc-3EXAMPLE",
            "OwnerId": "1111222233333",
```

```
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "MySubnet"
                }
            ],
            "SubnetArn": "arn:aws:ec2:us-east-1:111122223333:subnet/
subnet-8EXAMPLE",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            }
        }
    ]
}
```

如需詳細資訊，請參閱《AWS VPC 使用者指南》中的使用 VPC 和子網路。

範例 2：描述特定 VPC 的子網路

下列 describe-subnets 範例會使用篩選條件來擷取指定 VPC 得子網路詳細資訊。

```
aws ec2 describe-subnets \
    --filters "Name=vpc-id,Values=vpc-3EXAMPLE"
```

輸出：

```
{
    "Subnets": [
        {
            "AvailabilityZone": "us-east-1d",
            "AvailabilityZoneId": "use1-az2",
            "AvailableIpAddressCount": 4089,
            "CidrBlock": "172.31.80.0/20",
            "DefaultForAz": true,
            "MapPublicIpOnLaunch": true,
            "MapCustomerOwnedIpOnLaunch": false,
            "State": "available",
```

```
            "SubnetId": "subnet-8EXAMPLE",
            "VpcId": "vpc-3EXAMPLE",
            "OwnerId": "1111222233333",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "MySubnet"
                }
            ],
            "SubnetArn": "arn:aws:ec2:us-east-1:111122223333:subnet/
subnet-8EXAMPLE",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            }
        }
    ]
}
```

如需詳細資訊，請參閱《AWS VPC 使用者指南》中的使用 VPC 和子網路。

範例 3：描述具有特定標籤的子網路

下列 `describe-subnets` 範例會使用篩選條件來擷取這些子網路 (其中包含標籤 `CostCenter=123` 和 `--query` 參數) 的詳細資訊，以顯示具有此標籤之子網路的子網路 ID。

```
aws ec2 describe-subnets \
    --filters "Name=tag:CostCenter,Values=123" \
    --query "Subnets[*].SubnetId" \
    --output text
```

輸出：

```
subnet-0987a87c8b37348ef
subnet-02a95061c45f372ee
subnet-03f720e7de2788d73
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的使用 VPC 和子網路。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeSubnets。

describe-tags

以下程式碼範例顯示如何使用 describe-tags。

AWS CLI

範例 1：描述單一資源的所有標籤

下列describe-tags範例說明指定執行個體的標籤。

```
aws ec2 describe-tags \
    --filters "Name=resource-id,Values=i-1234567890abcdef8"
```

輸出：

```
{
    "Tags": [
        {
            "ResourceType": "instance",
            "ResourceId": "i-1234567890abcdef8",
            "Value": "Test",
            "Key": "Stack"
        },
        {
            "ResourceType": "instance",
            "ResourceId": "i-1234567890abcdef8",
            "Value": "Beta Server",
            "Key": "Name"
        }
    ]
}
```

範例 2：描述資源類型的所有標籤

下列describe-tags範例說明 磁碟區的標籤。

```
aws ec2 describe-tags \
    --filters "Name=resource-type,Values=volume"
```

輸出：

```
{
    "Tags": [
        {
            "ResourceType": "volume",
            "ResourceId": "vol-1234567890abcdef0",
            "Value": "Project1",
            "Key": "Purpose"
        },
        {
            "ResourceType": "volume",
            "ResourceId": "vol-049df61146c4d7901",
            "Value": "Logs",
            "Key": "Purpose"
        }
    ]
}
```

範例 3：描述您的所有標籤

下列describe-tags範例說明所有 資源的標籤。

```
aws ec2 describe-tags
```

範例 4：根據標籤索引鍵描述資源的標籤

下列describe-tags範例說明具有金鑰 標籤之 資源的標籤Stack。

```
aws ec2 describe-tags \
    --filters Name=key,Values=Stack
```

輸出：

```
{
    "Tags": [
        {
            "ResourceType": "volume",
            "ResourceId": "vol-027552a73f021f3b",
            "Value": "Production",
            "Key": "Stack"
        },
        {
            "ResourceType": "instance",
```

```
            "ResourceId": "i-1234567890abcdef8",
            "Value": "Test",
            "Key": "Stack"
        }
    ]
}
```

範例 5：根據標籤索引鍵和標籤值描述資源的標籤

下列describe-tags範例說明具有標籤 之 資源的標籤Stack=Test。

```
aws ec2 describe-tags \
    --filters Name=key,Values=Stack Name=value,Values=Test
```

輸出：

```
{
    "Tags": [
        {
            "ResourceType": "image",
            "ResourceId": "ami-3ac336533f021f3bd",
            "Value": "Test",
            "Key": "Stack"
        },
        {
            "ResourceType": "instance",
            "ResourceId": "i-1234567890abcdef8",
            "Value": "Test",
            "Key": "Stack"
        }
    ]
}
```

下列describe-tags範例使用替代語法來描述具有標籤 的資源Stack=Test。

```
aws ec2 describe-tags \
    --filters "Name=tag:Stack,Values=Test"
```

下列describe-tags範例說明所有執行個體的標籤，其標籤具有 索引鍵Purpose且無值。

```
aws ec2 describe-tags \
```

```
    --filters "Name=resource-
type,Values=instance" "Name=key,Values=Purpose" "Name=value,Values="
```

輸出：

```
{
    "Tags": [
        {
            "ResourceType": "instance",
            "ResourceId": "i-1234567890abcdef5",
            "Value": null,
            "Key": "Purpose"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTags。

describe-traffic-mirror-filters

以下程式碼範例顯示如何使用 describe-traffic-mirror-filters。

AWS CLI

檢視流量鏡像篩選條件

下列describe-traffic-mirror-filters範例顯示所有流量鏡像篩選條件的詳細資訊。

```
aws ec2 describe-traffic-mirror-filters
```

輸出：

```
{
    "TrafficMirrorFilters": [
        {
            "TrafficMirrorFilterId": "tmf-0293f26e86EXAMPLE",
            "IngressFilterRules": [
                {
                    "TrafficMirrorFilterRuleId": "tmfr-0ca76e0e08EXAMPLE",
                    "TrafficMirrorFilterId": "tmf-0293f26e86EXAMPLE",
                    "TrafficDirection": "ingress",
                    "RuleNumber": 100,
```

```
                    "RuleAction": "accept",
                    "Protocol": 6,
                    "DestinationCidrBlock": "10.0.0.0/24",
                    "SourceCidrBlock": "10.0.0.0/24",
                    "Description": "TCP Rule"
                }
            ],
            "EgressFilterRules": [],
            "NetworkServices": [],
            "Description": "Example filter",
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱《流量鏡射指南》中的檢視流量鏡射篩選條件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTrafficMirrorFilters。

describe-traffic-mirror-sessions

以下程式碼範例顯示如何使用 describe-traffic-mirror-sessions。

AWS CLI

描述流量鏡射工作階段

下列describe-traffic-mirror-sessions範例顯示流量鏡射工作階段的詳細資訊。

```
aws ec2 describe-traffic-mirror-sessions
```

輸出：

```
{
    "TrafficMirrorSessions": [
        {
            "Tags": [],
            "VirtualNetworkId": 42,
            "OwnerId": "111122223333",
            "Description": "TCP Session",
            "NetworkInterfaceId": "eni-0a471a5cf3EXAMPLE",
            "TrafficMirrorTargetId": "tmt-0dabe9b0a6EXAMPLE",
            "TrafficMirrorFilterId": "tmf-083e18f985EXAMPLE",
```

```
            "PacketLength": 20,
            "SessionNumber": 1,
            "TrafficMirrorSessionId": "tms-0567a4c684EXAMPLE"
        },
        {
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "tag test"
                }
            ],
            "VirtualNetworkId": 13314501,
            "OwnerId": "111122223333",
            "Description": "TCP Session",
            "NetworkInterfaceId": "eni-0a471a5cf3EXAMPLE",
            "TrafficMirrorTargetId": "tmt-03665551cbEXAMPLE",
            "TrafficMirrorFilterId": "tmf-06c787846cEXAMPLE",
            "SessionNumber": 2,
            "TrafficMirrorSessionId": "tms-0060101cf8EXAMPLE"
        }
    ]
}
```

如需詳細資訊，請參閱《流量鏡射指南》中的檢視流量鏡射工作階段詳細資訊。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTrafficMirrorSessions。

describe-traffic-mirror-targets

以下程式碼範例顯示如何使用 describe-traffic-mirror-targets。

AWS CLI

描述流量鏡像目標

下列describe-traffic-mirror-targets範例顯示指定流量鏡像目標的相關資訊。

```
aws ec2 describe-traffic-mirror-targets \
    --traffic-mirror-target-ids tmt-0dabe9b0a6EXAMPLE
```

輸出：

```
{
```

```
    "TrafficMirrorTargets": [
        {
            "TrafficMirrorTargetId": "tmt-0dabe9b0a6EXAMPLE",
            "NetworkLoadBalancerArn": "arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/net/NLB/7cdec873fEXAMPLE",
            "Type": "network-load-balancer",
            "Description": "Example Network Load Balancer target",
            "OwnerId": "111122223333",
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC 流量鏡射指南》中的流量鏡射目標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTrafficMirrorTargets。

describe-transit-gateway-attachments

以下程式碼範例顯示如何使用 describe-transit-gateway-attachments。

AWS CLI

檢視您的傳輸閘道附件

下列describe-transit-gateway-attachments範例顯示傳輸閘道附件的詳細資訊。

```
aws ec2 describe-transit-gateway-attachments
```

輸出：

```
{
    "TransitGatewayAttachments": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-01f8100bc7EXAMPLE",
            "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
            "TransitGatewayOwnerId": "123456789012",
            "ResourceOwnerId": "123456789012",
            "ResourceType": "vpc",
            "ResourceId": "vpc-3EXAMPLE",
            "State": "available",
            "Association": {
                "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
```

```
                "State": "associated"
            },
            "CreationTime": "2019-08-26T14:59:25.000Z",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Example"
                }
            ]
        },
        {
            "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
            "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
            "TransitGatewayOwnerId": "123456789012",
            "ResourceOwnerId": "123456789012",
            "ResourceType": "vpc",
            "ResourceId": "vpc-0065acced4EXAMPLE",
            "State": "available",
            "Association": {
                "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
                "State": "associated"
            },
            "CreationTime": "2019-08-07T17:03:07.000Z",
            "Tags": []
        },
        {
            "TransitGatewayAttachmentId": "tgw-attach-08e0bc912cEXAMPLE",
            "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
            "TransitGatewayOwnerId": "123456789012",
            "ResourceOwnerId": "123456789012",
            "ResourceType": "direct-connect-gateway",
            "ResourceId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
            "State": "available",
            "Association": {
                "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
                "State": "associated"
            },
            "CreationTime": "2019-08-14T20:27:44.000Z",
            "Tags": []
        },
        {
            "TransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
            "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
            "TransitGatewayOwnerId": "123456789012",
```

```
            "ResourceOwnerId": "123456789012",
            "ResourceType": "direct-connect-gateway",
            "ResourceId": "8384da05-13ce-4a91-aada-5a1baEXAMPLE",
            "State": "available",
            "Association": {
                "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
                "State": "associated"
            },
            "CreationTime": "2019-08-14T20:33:02.000Z",
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的使用傳輸閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTransitGatewayAttachments。

describe-transit-gateway-connect-peers

以下程式碼範例顯示如何使用 describe-transit-gateway-connect-peers。

AWS CLI

描述 Transit Gateway Connect 對等

下列describe-transit-gateway-connect-peers範例說明指定的 Connect 對等。

```
aws ec2 describe-transit-gateway-connect-peers \
    --transit-gateway-connect-peer-ids tgw-connect-peer-0666adbac4EXAMPLE
```

輸出：

```
{
    "TransitGatewayConnectPeers": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-0f0927767cEXAMPLE",
            "TransitGatewayConnectPeerId": "tgw-connect-peer-0666adbac4EXAMPLE",
            "State": "available",
            "CreationTime": "2021-10-13T03:35:17.000Z",
            "ConnectPeerConfiguration": {
```

```
                "TransitGatewayAddress": "10.0.0.234",
                "PeerAddress": "172.31.1.11",
                "InsideCidrBlocks": [
                    "169.254.6.0/29"
                ],
                "Protocol": "gre",
                "BgpConfigurations": [
                    {
                        "TransitGatewayAsn": 64512,
                        "PeerAsn": 64512,
                        "TransitGatewayAddress": "169.254.6.2",
                        "PeerAddress": "169.254.6.1",
                        "BgpStatus": "down"
                    },
                    {
                        "TransitGatewayAsn": 64512,
                        "PeerAsn": 64512,
                        "TransitGatewayAddress": "169.254.6.3",
                        "PeerAddress": "169.254.6.1",
                        "BgpStatus": "down"
                    }
                ]
            },
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的 Transit Gateway Connect 連接和 Transit Gateway Connect 對等。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTransitGatewayConnectPeers。

`describe-transit-gateway-connects`

以下程式碼範例顯示如何使用 `describe-transit-gateway-connects`。

AWS CLI

描述傳輸閘道 Connect 連接

下列`describe-transit-gateway-connects`範例說明指定的 Connect 連接。

```
aws ec2 describe-transit-gateway-connects \
    --transit-gateway-attachment-ids tgw-attach-037012e5dcEXAMPLE
```

輸出：

```
{
    "TransitGatewayConnects": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-037012e5dcEXAMPLE",
            "TransportTransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
            "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
            "State": "available",
            "CreationTime": "2021-03-09T19:59:17+00:00",
            "Options": {
                "Protocol": "gre"
            },
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的 Transit Gateway Connect 連接和 Transit Gateway Connect 對等。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTransitGatewayConnects。

describe-transit-gateway-multicast-domains

以下程式碼範例顯示如何使用 describe-transit-gateway-multicast-domains。

AWS CLI

描述您的傳輸閘道多點傳送網域

下列describe-transit-gateway-multicast-domains範例顯示所有傳輸閘道多點傳送網域的詳細資訊。

```
aws ec2 describe-transit-gateway-multicast-domains
```

輸出：

```
{
```

```
    "TransitGatewayMulticastDomains": [
        {
            "TransitGatewayMulticastDomainId": "tgw-mcast-domain-000fb24d04EXAMPLE",
            "TransitGatewayId": "tgw-0bf0bffefaEXAMPLE",
            "TransitGatewayMulticastDomainArn": "arn:aws:ec2:us-
east-1:123456789012:transit-gateway-multicast-domain/tgw-mcast-
domain-000fb24d04EXAMPLE",
            "OwnerId": "123456789012",
            "Options": {
                "Igmpv2Support": "disable",
                "StaticSourcesSupport": "enable",
                "AutoAcceptSharedAssociations": "disable"
            },
            "State": "available",
            "CreationTime": "2019-12-10T18:32:50+00:00",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "mc1"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的管理多點傳送網域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTransitGatewayMulticastDomains。

describe-transit-gateway-peering-attachments

以下程式碼範例顯示如何使用 describe-transit-gateway-peering-attachments。

AWS CLI

描述您的傳輸閘道對等互連附件

下列describe-transit-gateway-peering-attachments範例顯示所有傳輸閘道對等互連附件的詳細資訊。

```
aws ec2 describe-transit-gateway-peering-attachments
```

輸出：

```
{
    "TransitGatewayPeeringAttachments": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
            "RequesterTgwInfo": {
                "TransitGatewayId": "tgw-123abc05e04123abc",
                "OwnerId": "123456789012",
                "Region": "us-west-2"
            },
            "AccepterTgwInfo": {
                "TransitGatewayId": "tgw-11223344aabbcc112",
                "OwnerId": "123456789012",
                "Region": "us-east-2"
            },
            "State": "pendingAcceptance",
            "CreationTime": "2019-12-09T11:38:05.000Z",
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的 Transit Gateway 對等連接。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTransitGatewayPeeringAttachments。

describe-transit-gateway-policy-tables

以下程式碼範例顯示如何使用 describe-transit-gateway-policy-tables。

AWS CLI

描述傳輸閘道政策表

下列describe-transit-gateway-policy-tables範例說明指定的傳輸閘道政策表。

```
aws ec2 describe-transit-gateway-policy-tables \
    --transit-gateway-policy-table-ids tgw-ptb-0a16f134b78668a81
```

輸出：

```
{
    "TransitGatewayPolicyTables": [
        {
            "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
            "TransitGatewayId": "tgw-067f8505c18f0bd6e",
            "State": "available",
            "CreationTime": "2023-11-28T16:36:43+00:00",
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱《傳輸閘道使用者指南》中的傳輸閘道政策表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTransitGatewayPolicyTables。

describe-transit-gateway-route-tables

以下程式碼範例顯示如何使用 describe-transit-gateway-route-tables。

AWS CLI

描述您的傳輸閘道路由表

下列describe-transit-gateway-route-tables範例顯示傳輸閘道路由表的詳細資訊。

```
aws ec2 describe-transit-gateway-route-tables
```

輸出：

```
{
    "TransitGatewayRouteTables": [
        {
            "TransitGatewayRouteTableId": "tgw-rtb-0ca78a549EXAMPLE",
            "TransitGatewayId": "tgw-0bc994abffEXAMPLE",
            "State": "available",
            "DefaultAssociationRouteTable": true,
            "DefaultPropagationRouteTable": true,
            "CreationTime": "2018-11-28T14:24:49.000Z",
            "Tags": []
        },
```

```
        {
            "TransitGatewayRouteTableId": "tgw-rtb-0e8f48f148EXAMPLE",
            "TransitGatewayId": "tgw-0043d72bb4EXAMPLE",
            "State": "available",
            "DefaultAssociationRouteTable": true,
            "DefaultPropagationRouteTable": true,
            "CreationTime": "2018-11-28T14:24:00.000Z",
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱《傳輸閘道指南》中的檢視傳輸閘道路由表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTransitGatewayRouteTables。

describe-transit-gateway-vpc-attachments

以下程式碼範例顯示如何使用 describe-transit-gateway-vpc-attachments。

AWS CLI

描述您的傳輸閘道 VPC 連接

下列describe-transit-gateway-vpc-attachments範例顯示傳輸閘道 VPC 連接的詳細資訊。

```
aws ec2 describe-transit-gateway-vpc-attachments
```

輸出：

```
{
    "TransitGatewayVpcAttachments": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-0a08e88308EXAMPLE",
            "TransitGatewayId": "tgw-0043d72bb4EXAMPLE",
            "VpcId": "vpc-0f501f7ee8EXAMPLE",
            "VpcOwnerId": "111122223333",
            "State": "available",
            "SubnetIds": [
                "subnet-045d586432EXAMPLE",
```

```
                "subnet-0a0ad478a6EXAMPLE"
            ],
            "CreationTime": "2019-02-13T11:04:02.000Z",
            "Options": {
                "DnsSupport": "enable",
                "Ipv6Support": "disable"
            },
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "attachment name"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的檢視您的 VPC 附件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTransitGatewayVpcAttachments。

describe-transit-gateways

以下程式碼範例顯示如何使用 describe-transit-gateways。

AWS CLI

描述您的傳輸閘道

下列describe-transit-gateways範例會擷取傳輸閘道的詳細資訊。

```
aws ec2 describe-transit-gateways
```

輸出：

```
{
    "TransitGateways": [
        {
            "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
            "TransitGatewayArn": "arn:aws:ec2:us-east-2:111122223333:transit-
gateway/tgw-0262a0e521EXAMPLE",
```

```
            "State": "available",
            "OwnerId": "111122223333",
            "Description": "MyTGW",
            "CreationTime": "2019-07-10T14:02:12.000Z",
            "Options": {
                "AmazonSideAsn": 64516,
                "AutoAcceptSharedAttachments": "enable",
                "DefaultRouteTableAssociation": "enable",
                "AssociationDefaultRouteTableId": "tgw-rtb-018774adf3EXAMPLE",
                "DefaultRouteTablePropagation": "enable",
                "PropagationDefaultRouteTableId": "tgw-rtb-018774adf3EXAMPLE",
                "VpnEcmpSupport": "enable",
                "DnsSupport": "enable"
            },
            "Tags": []
        },
        {
            "TransitGatewayId": "tgw-0fb8421e2dEXAMPLE",
            "TransitGatewayArn": "arn:aws:ec2:us-east-2:111122223333:transit-
gateway/tgw-0fb8421e2da853bf3",
            "State": "available",
            "OwnerId": "111122223333",
            "CreationTime": "2019-03-15T22:57:33.000Z",
            "Options": {
                "AmazonSideAsn": 65412,
                "AutoAcceptSharedAttachments": "disable",
                "DefaultRouteTableAssociation": "enable",
                "AssociationDefaultRouteTableId": "tgw-rtb-06a241a3d8EXAMPLE",
                "DefaultRouteTablePropagation": "enable",
                "PropagationDefaultRouteTableId": "tgw-rtb-06a241a3d8EXAMPLE",
                "VpnEcmpSupport": "enable",
                "DnsSupport": "enable"
            },
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "TGW1"
                }
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTransitGateways。

describe-verified-access-endpoints

以下程式碼範例顯示如何使用 describe-verified-access-endpoints。

AWS CLI

描述 Verified Access 端點

下列describe-verified-access-endpoints範例說明指定的 Verified Access 端點。

```
aws ec2 describe-verified-access-endpoints \
    --verified-access-endpoint-ids vae-066fac616d4d546f2
```

輸出：

```
{
    "VerifiedAccessEndpoints": [
        {
            "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
            "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
            "VerifiedAccessEndpointId": "vae-066fac616d4d546f2",
            "ApplicationDomain": "example.com",
            "EndpointType": "network-interface",
            "AttachmentType": "vpc",
            "DomainCertificateArn": "arn:aws:acm:us-east-2:123456789012:certificate/
eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE",
            "EndpointDomain": "my-ava-
app.edge-00c3372d53b1540bb.vai-0ce000c0b7643abea.prod.verified-access.us-
east-2.amazonaws.com",
            "SecurityGroupIds": [
                "sg-004915970c4c8f13a"
            ],
            "NetworkInterfaceOptions": {
                "NetworkInterfaceId": "eni-0aec70418c8d87a0f",
                "Protocol": "https",
                "Port": 443
            },
            "Status": {
                "Code": "active"
            },
            "Description": "",
            "CreationTime": "2023-08-25T20:54:43",
            "LastUpdatedTime": "2023-08-25T22:17:26",
```

```
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "my-va-endpoint"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Verified Access 使用者指南》中的 Verified Access 端點。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVerifiedAccessEndpoints。

describe-verified-access-groups

以下程式碼範例顯示如何使用 describe-verified-access-groups。

AWS CLI

描述 Verified Access 群組

下列describe-verified-access-groups範例說明指定的 Verified Access 群組。

```
aws ec2 describe-verified-access-groups \
    --verified-access-group-ids vagr-0dbe967baf14b7235
```

輸出：

```
{
    "VerifiedAccessGroups": [
        {
            "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
            "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
            "Description": "Testing Verified Access",
            "Owner": "123456789012",
            "VerifiedAccessGroupArn": "arn:aws:ec2:us-east-2:123456789012:verified-
access-group/vagr-0dbe967baf14b7235",
            "CreationTime": "2023-08-25T19:55:19",
            "LastUpdatedTime": "2023-08-25T22:17:25",
            "Tags": [
                {
                    "Key": "Name",
```

```
                    "Value": "my-va-group"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱 Verified Access 使用者指南中的 Verified Access 群組。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVerifiedAccessGroups。

describe-verified-access-instance-logging-configurations

以下程式碼範例顯示如何使用 describe-verified-access-instance-logging-configurations。

AWS CLI

描述 Verified Access 執行個體的記錄組態

下列describe-verified-access-instance-logging-configurations範例說明指定 Verified Access 執行個體的記錄組態。

```
aws ec2 describe-verified-access-instance-logging-configurations \
    --verified-access-instance-ids vai-0ce000c0b7643abea
```

輸出：

```
{
    "LoggingConfigurations": [
        {
            "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
            "AccessLogs": {
                "S3": {
                    "Enabled": false
                },
                "CloudWatchLogs": {
                    "Enabled": true,
                    "DeliveryStatus": {
                        "Code": "success"
                    },
                    "LogGroup": "my-log-group"
                },
```

```
                "KinesisDataFirehose": {
                    "Enabled": false
                },
                "LogVersion": "ocsf-1.0.0-rc.2",
                "IncludeTrustContext": false
            }
        }
    ]
}
```

如需詳細資訊，請參閱 Verified Access 使用者指南AWS 中的 Verified Access Logs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVerifiedAccessInstanceLoggingConfigurations。

describe-verified-access-instances

以下程式碼範例顯示如何使用 describe-verified-access-instances。

AWS CLI

描述 Verified Access 執行個體

下列describe-verified-access-instances範例說明指定的 Verified Access 執行個體。

```
aws ec2 describe-verified-access-instances \
    --verified-access-instance-ids vai-0ce000c0b7643abea
```

輸出：

```
{
    "VerifiedAccessInstances": [
        {
            "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
            "Description": "Testing Verified Access",
            "VerifiedAccessTrustProviders": [
                {
                    "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
                    "TrustProviderType": "user",
                    "UserTrustProviderType": "iam-identity-center"
                }
            ],
            "CreationTime": "2023-08-25T18:27:56",
```

```
            "LastUpdatedTime": "2023-08-25T19:03:32",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "my-ava-instance"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱 Verified Access 使用者指南中的 Verified Access 執行個體。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVerifiedAccessInstances。

describe-verified-access-trust-providers

以下程式碼範例顯示如何使用 describe-verified-access-trust-providers。

AWS CLI

描述 Verified Access 信任提供者

下列describe-verified-access-trust-providers範例說明指定的 Verified Access 信任提供者。

```
aws ec2 describe-verified-access-trust-providers \
    --verified-access-trust-provider-ids vatp-0bb32de759a3e19e7
```

輸出：

```
{
    "VerifiedAccessTrustProviders": [
        {
            "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
            "Description": "Testing Verified Access",
            "TrustProviderType": "user",
            "UserTrustProviderType": "iam-identity-center",
            "PolicyReferenceName": "idc",
            "CreationTime": "2023-08-25T19:00:38",
            "LastUpdatedTime": "2023-08-25T19:03:32",
            "Tags": [
                {
```

```
                    "Key": "Name",
                    "Value": "my-va-trust-provider"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Verified Access 使用者指南》中的 Verified Access 的信任提供者。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVerifiedAccessTrustProviders。

describe-volume-attribute

以下程式碼範例顯示如何使用 describe-volume-attribute。

AWS CLI

描述磁碟區屬性

此範例命令說明 ID 為 之磁碟區的autoEnableIo屬性vol-049df61146c4d7901。

命令：

```
aws ec2 describe-volume-attribute --volume-id vol-049df61146c4d7901 --
attribute autoEnableIO
```

輸出：

```
{
    "AutoEnableIO": {
        "Value": false
    },
    "VolumeId": "vol-049df61146c4d7901"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVolumeAttribute。

describe-volume-status

以下程式碼範例顯示如何使用 describe-volume-status。

AWS CLI

描述單一磁碟區的狀態

此範例命令說明磁碟區 的狀態vol-1234567890abcdef0。

命令：

```
aws ec2 describe-volume-status --volume-ids vol-1234567890abcdef0
```

輸出：

```
{
    "VolumeStatuses": [
        {
            "VolumeStatus": {
                "Status": "ok",
                "Details": [
                    {
                        "Status": "passed",
                        "Name": "io-enabled"
                    },
                    {
                        "Status": "not-applicable",
                        "Name": "io-performance"
                    }
                ]
            },
            "AvailabilityZone": "us-east-1a",
            "VolumeId": "vol-1234567890abcdef0",
            "Actions": [],
            "Events": []
        }
    ]
}
```

描述磁碟區受損的狀態

此範例命令說明所有磁碟區受損的狀態。在此範例輸出中，沒有磁碟區受損。

命令：

```
aws ec2 describe-volume-status --filters Name=volume-status.status,Values=impaired
```

輸出：

```
{
    "VolumeStatuses": []
}
```

如果您有狀態檢查失敗的磁碟區（狀態受損），請參閱《Amazon EC2 使用者指南》中的使用受損磁碟區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVolumeStatus。

describe-volumes-modifications

以下程式碼範例顯示如何使用 describe-volumes-modifications。

AWS CLI

描述磁碟區的修改狀態

下列describe-volumes-modifications範例說明指定磁碟區的磁碟區修改狀態。

```
aws ec2 describe-volumes-modifications \
    --volume-ids vol-1234567890abcdef0
```

輸出：

```
{
    "VolumeModification": {
        "TargetSize": 150,
        "TargetVolumeType": "io1",
        "ModificationState": "optimizing",
        "VolumeId": " vol-1234567890abcdef0",
        "TargetIops": 100,
        "StartTime": "2019-05-17T11:27:19.000Z",
        "Progress": 70,
        "OriginalVolumeType": "io1",
        "OriginalIops": 100,
        "OriginalSize": 100
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVolumesModifications。

describe-volumes

以下程式碼範例顯示如何使用 describe-volumes。

AWS CLI

範例 1：描述磁碟區

下列describe-volumes範例說明目前區域中指定的磁碟區。

```
aws ec2 describe-volumes \
    --volume-ids vol-049df61146c4d7901 vol-1234567890abcdef0
```

輸出：

```
{
    "Volumes": [
        {
            "AvailabilityZone": "us-east-1a",
            "Attachments": [
                {
                    "AttachTime": "2013-12-18T22:35:00.000Z",
                    "InstanceId": "i-1234567890abcdef0",
                    "VolumeId": "vol-049df61146c4d7901",
                    "State": "attached",
                    "DeleteOnTermination": true,
                    "Device": "/dev/sda1"
                }
            ],
            "Encrypted": true,
            "KmsKeyId": "arn:aws:kms:us-east-2a:123456789012:key/8c5b2c63-b9bc-45a3-
a87a-5513eEXAMPLE,
            "VolumeType": "gp2",
            "VolumeId": "vol-049df61146c4d7901",
            "State": "in-use",
            "Iops": 100,
            "SnapshotId": "snap-1234567890abcdef0",
            "CreateTime": "2019-12-18T22:35:00.084Z",
            "Size": 8
        },
        {
            "AvailabilityZone": "us-east-1a",
            "Attachments": [],
            "Encrypted": false,
```

```
            "VolumeType": "gp2",
            "VolumeId": "vol-1234567890abcdef0",
            "State": "available",
            "Iops": 300,
            "SnapshotId": "",
            "CreateTime": "2020-02-27T00:02:41.791Z",
            "Size": 100
        }
    ]
}
```

範例 2：描述連接至特定執行個體的磁碟區

下列describe-volumes範例說明連接至指定執行個體的所有磁碟區，並設定為在執行個體終止時刪除。

```
aws ec2 describe-volumes \
    --region us-east-1 \
    --filters Name=attachment.instance-
id,Values=i-1234567890abcdef0 Name=attachment.delete-on-termination,Values=true
```

如需 describe-volumes 的輸出範例，請參閱範例 1。

範例 3：描述特定可用區域中的可用磁碟區

下列describe-volumes範例說明狀態為 available且位於指定可用區域內的所有磁碟區。

```
aws ec2 describe-volumes \
    --filters Name=status,Values=available Name=availability-zone,Values=us-east-1a
```

如需 describe-volumes 的輸出範例，請參閱範例 1。

範例 4：根據標籤描述磁碟區

下列describe-volumes範例說明具有標籤索引鍵Name和開頭為 的值的所有磁碟區Test。然後，系統會使用僅顯示磁碟區標籤和 IDs的查詢來篩選輸出。

```
aws ec2 describe-volumes \
    --filters Name=tag:Name,Values=Test* \
    --query "Volumes[*].{ID:VolumeId,Tag:Tags}"
```

輸出：

```
[
    {
        "Tag": [
            {
                "Value": "Test2",
                "Key": "Name"
            }
        ],
        "ID": "vol-1234567890abcdef0"
    },
    {
        "Tag": [
            {
                "Value": "Test1",
                "Key": "Name"
            }
        ],
        "ID": "vol-049df61146c4d7901"
    }
]
```

如需使用標籤篩選條件的其他範例，請參閱《Amazon EC2 使用者指南》中的使用標籤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVolumes。

describe-vpc-attribute

以下程式碼範例顯示如何使用 `describe-vpc-attribute`。

AWS CLI

描述 enableDnsSupport 屬性

此範例說明 `enableDnsSupport` 屬性。此屬性指出是否已啟用 VPC 的 DNS 解析。如果此屬性為 `true`，Amazon DNS 伺服器會將您的執行個體的 DNS 主機名稱解析為對應的 IP 地址；否則將不會進行解析。

命令：

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute enableDnsSupport
```

輸出：

```
{
    "VpcId": "vpc-a01106c2",
    "EnableDnsSupport": {
        "Value": true
    }
}
```

描述 enableDnsHostnames 屬性

此範例說明 enableDnsHostnames 屬性。此屬性指出在 VPC 中啟動的執行個體是否取得 DNS 主機名稱。如果此屬性為 true，該 VPC 中的執行個體會取得 DNS 主機名稱；否則將不會取得。

命令：

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute enableDnsHostnames
```

輸出：

```
{
    "VpcId": "vpc-a01106c2",
    "EnableDnsHostnames": {
        "Value": true
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVpcAttribute。

describe-vpc-classic-link-dns-support

以下程式碼範例顯示如何使用 describe-vpc-classic-link-dns-support。

AWS CLI

描述 VPCs的 ClassicLink DNS 支援

此範例說明所有 VPCs的 ClassicLink DNS 支援狀態。

命令：

```
aws ec2 describe-vpc-classic-link-dns-support
```

輸出：

```
{
  "Vpcs": [
    {
      "VpcId": "vpc-88888888",
      "ClassicLinkDnsSupported": true
    },
    {
      "VpcId": "vpc-1a2b3c4d",
      "ClassicLinkDnsSupported": false
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVpcClassicLinkDnsSupport。

describe-vpc-classic-link

以下程式碼範例顯示如何使用 describe-vpc-classic-link。

AWS CLI

描述 VPCs的 ClassicLink 狀態

此範例列出 vpc-88888888 的 ClassicLink 狀態。

命令：

```
aws ec2 describe-vpc-classic-link --vpc-id vpc-88888888
```

輸出：

```
{
  "Vpcs": [
    {
      "ClassicLinkEnabled": true,
      "VpcId": "vpc-88888888",
      "Tags": [
        {
          "Value": "classiclinkvpc",
          "Key": "Name"
        }
      ]
```

```
    }
  ]
}
```

此範例只會列出針對 Classiclink 啟用VPCs (的篩選條件值`is-classic-link-enabled`設定為`true`)。

命令：

```
aws ec2 describe-vpc-classic-link --filter "Name=is-classic-link-
enabled,Values=true"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVpcClassicLink。

describe-vpc-endpoint-associations

以下程式碼範例顯示如何使用 `describe-vpc-endpoint-associations`。

AWS CLI

描述 VPC 端點關聯

下列`describe-vpc-endpoint-associations`範例說明您的 VPC 端點關聯。

```
aws ec2 describe-vpc-endpoint-associations
```

輸出：

```
{
    "VpcEndpointAssociations": [
        {
            "Id": "vpce-rsc-asc-0a810ca6ac8866bf9",
            "VpcEndpointId": "vpce-019b90d6f16d4f958",
            "AssociatedResourceAccessibility": "Accessible",
            "DnsEntry": {
                "DnsName":
 "vpce-019b90d6f16d4f958.rcfg-07129f3acded87625.4232ccc.vpc-lattice-rsc.us-
east-2.on.aws",
                "HostedZoneId": "Z03265862FOUNWMZOKUF4"
            },
            "AssociatedResourceArn": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceconfiguration/rcfg-07129f3acded87625"
```

```
        }
    ]
}
```

如需詳細資訊，請參閱《 AWS PrivateLink 使用者指南》中的管理 VPC 端點關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVpcEndpointAssociations。

describe-vpc-endpoint-connection-notifications

以下程式碼範例顯示如何使用 describe-vpc-endpoint-connection-notifications。

AWS CLI

描述端點連線通知

下列describe-vpc-endpoint-connection-notifications範例說明您的所有端點連線通知。

```
aws ec2 describe-vpc-endpoint-connection-notifications
```

輸出：

```
{
   "ConnectionNotificationSet": [
       {
           "ConnectionNotificationState": "Enabled",
           "ConnectionNotificationType": "Topic",
           "ConnectionEvents": [
               "Accept",
               "Reject",
               "Delete",
               "Connect"
           ],
           "ConnectionNotificationId": "vpce-nfn-04bcb952bc8af7abc",
           "ConnectionNotificationArn": "arn:aws:sns:us-
east-1:123456789012:VpceNotification",
           "VpcEndpointId": "vpce-0324151a02f327123"
       }
   ]
 }
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVpcEndpointConnectionNotifications。

describe-vpc-endpoint-connections

以下程式碼範例顯示如何使用 describe-vpc-endpoint-connections。

AWS CLI

描述 VPC 端點連線

此範例說明與端點服務的介面端點連線，並篩選結果以顯示 的端點PendingAcceptance。

命令：

```
aws ec2 describe-vpc-endpoint-connections --filters Name=vpc-endpoint-
state,Values=pendingAcceptance
```

輸出：

```
{
  "VpcEndpointConnections": [
      {
          "VpcEndpointId": "vpce-0abed31004e618123",
          "ServiceId": "vpce-svc-0abced088d20def56",
          "CreationTimestamp": "2017-11-30T10:00:24.350Z",
          "VpcEndpointState": "pendingAcceptance",
          "VpcEndpointOwner": "123456789012"
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVpcEndpointConnections。

describe-vpc-endpoint-service-configurations

以下程式碼範例顯示如何使用 describe-vpc-endpoint-service-configurations。

AWS CLI

描述端點服務組態

下列describe-vpc-endpoint-service-configurations範例說明您的端點服務組態。

```
aws ec2 describe-vpc-endpoint-service-configurations
```

輸出：

```
{
    "ServiceConfigurations": [
        {
            "ServiceType": [
                {
                    "ServiceType": "GatewayLoadBalancer"
                }
            ],
            "ServiceId": "vpce-svc-012d33a1c4321cabc",
            "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-
svc-012d33a1c4321cabc",
            "ServiceState": "Available",
            "AvailabilityZones": [
                "us-east-1d"
            ],
            "AcceptanceRequired": false,
            "ManagesVpcEndpoints": false,
            "GatewayLoadBalancerArns": [
                "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/
gwy/GWLBService/123210844e429123"
            ],
            "Tags": []
        },
        {
            "ServiceType": [
                {
                    "ServiceType": "Interface"
                }
            ],
            "ServiceId": "vpce-svc-123cabc125efa123",
            "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-123cabc125efa123",
            "ServiceState": "Available",
            "AvailabilityZones": [
                "us-east-1a"
            ],
            "AcceptanceRequired": true,
            "ManagesVpcEndpoints": false,
```

```
            "NetworkLoadBalancerArns": [
                "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/
net/NLBforService/1238753950b25123"
            ],
            "BaseEndpointDnsNames": [
                "vpce-svc-123cabc125efa123.us-east-1.vpce.amazonaws.com"
            ],
            "PrivateDnsName": "example.com",
            "PrivateDnsNameConfiguration": {
                "State": "failed",
                "Type": "TXT",
                "Value": "vpce:qUAth3FdeABCApUiXabc",
                "Name": "_1d367jvbg34znqvyefrj"
            },
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱 AWS PrivateLink 使用者指南中的概念。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVpcEndpointServiceConfigurations。

describe-vpc-endpoint-service-permissions

以下程式碼範例顯示如何使用 describe-vpc-endpoint-service-permissions。

AWS CLI

描述端點服務許可

此範例說明指定端點服務的許可。

命令：

```
aws ec2 describe-vpc-endpoint-service-permissions --service-id vpce-
svc-03d5ebb7d9579a2b3
```

輸出：

```
{
   "AllowedPrincipals": [
```

```
        {
            "PrincipalType": "Account",
            "Principal": "arn:aws:iam::123456789012:root"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVpcEndpointServicePermissions。

describe-vpc-endpoint-services

以下程式碼範例顯示如何使用 describe-vpc-endpoint-services。

AWS CLI

範例 1：描述所有 VPC 端點服務

下列describe-vpc-endpoint-services範例列出 AWS 區域的所有 VPC 端點服務。

```
aws ec2 describe-vpc-endpoint-services
```

輸出：

```
{
    "ServiceDetails": [
        {
            "ServiceType": [
                {
                    "ServiceType": "Gateway"
                }
            ],
            "AcceptanceRequired": false,
            "ServiceName": "com.amazonaws.us-east-1.dynamodb",
            "VpcEndpointPolicySupported": true,
            "Owner": "amazon",
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1c",
                "us-east-1d",
                "us-east-1e",
```

```
                "us-east-1f"
            ],
            "BaseEndpointDnsNames": [
                "dynamodb.us-east-1.amazonaws.com"
            ]
        },
        {
            "ServiceType": [
                {
                    "ServiceType": "Interface"
                }
            ],
            "PrivateDnsName": "ec2.us-east-1.amazonaws.com",
            "ServiceName": "com.amazonaws.us-east-1.ec2",
            "VpcEndpointPolicySupported": false,
            "Owner": "amazon",
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1c",
                "us-east-1d",
                "us-east-1e",
                "us-east-1f"
            ],
            "AcceptanceRequired": false,
            "BaseEndpointDnsNames": [
                "ec2.us-east-1.vpce.amazonaws.com"
            ]
        },
        {
            "ServiceType": [
                {
                    "ServiceType": "Interface"
                }
            ],
            "PrivateDnsName": "ssm.us-east-1.amazonaws.com",
            "ServiceName": "com.amazonaws.us-east-1.ssm",
            "VpcEndpointPolicySupported": true,
            "Owner": "amazon",
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1c",
                "us-east-1d",
```

```
                "us-east-1e"
            ],
            "AcceptanceRequired": false,
            "BaseEndpointDnsNames": [
                "ssm.us-east-1.vpce.amazonaws.com"
            ]
        }
    ],
    "ServiceNames": [
        "com.amazonaws.us-east-1.dynamodb",
        "com.amazonaws.us-east-1.ec2",
        "com.amazonaws.us-east-1.ec2messages",
        "com.amazonaws.us-east-1.elasticloadbalancing",
        "com.amazonaws.us-east-1.kinesis-streams",
        "com.amazonaws.us-east-1.s3",
        "com.amazonaws.us-east-1.ssm"
    ]
}
```

範例 2：描述端點服務的詳細資訊

下列`describe-vpc-endpoint-services`範例列出 Amazon S3 介面端點服務的詳細資訊。

```
aws ec2 describe-vpc-endpoint-services \
    --filter 'Name=service-type,Values=Interface' Name=service-
name,Values=com.amazonaws.us-east-1.s3
```

輸出：

```
{
    "ServiceDetails": [
        {
            "ServiceName": "com.amazonaws.us-east-1.s3",
            "ServiceId": "vpce-svc-081d84efcdEXAMPLE",
            "ServiceType": [
                {
                    "ServiceType": "Interface"
                }
            ],
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1c",
```

```
                "us-east-1d",
                "us-east-1e",
            "us-east-1f"
            ],
            "Owner": "amazon",
            "BaseEndpointDnsNames": [
                "s3.us-east-1.vpce.amazonaws.com"
            ],
            "VpcEndpointPolicySupported": true,
            "AcceptanceRequired": false,
            "ManagesVpcEndpoints": false,
            "Tags": []
        }
    ],
    "ServiceNames": [
        "com.amazonaws.us-east-1.s3"
    ]
}
```

如需詳細資訊，請參閱《 AWS PrivateLink 使用者指南》中的檢視可用的 AWS 服務名稱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVpcEndpointServices。

`describe-vpc-endpoints`

以下程式碼範例顯示如何使用 `describe-vpc-endpoints`。

AWS CLI

描述您的 VPC 端點

下列`describe-vpc-endpoints`範例顯示所有 VPC 端點的詳細資訊。

```
aws ec2 describe-vpc-endpoints
```

輸出：

```
{
    "VpcEndpoints": [
        {
            "PolicyDocument": "{\"Version\":\"2008-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Principal\":\"*\",\"Action\":\"*\",\"Resource\":\"*\"}]}",
            "VpcId": "vpc-aabb1122",
```

```
            "NetworkInterfaceIds": [],
            "SubnetIds": [],
            "PrivateDnsEnabled": true,
            "State": "available",
            "ServiceName": "com.amazonaws.us-east-1.dynamodb",
            "RouteTableIds": [
                "rtb-3d560345"
            ],
            "Groups": [],
            "VpcEndpointId": "vpce-032a826a",
            "VpcEndpointType": "Gateway",
            "CreationTimestamp": "2017-09-05T20:41:28Z",
            "DnsEntries": [],
            "OwnerId": "123456789012"
        },
        {
            "PolicyDocument": "{\n  \"Statement\": [\n    {\n      \"Action\": \"*
\", \n      \"Effect\": \"Allow\", \n      \"Principal\": \"*\", \n      \"Resource
\": \"*\"\n    }\n  ]\n}",
            "VpcId": "vpc-1a2b3c4d",
            "NetworkInterfaceIds": [
                "eni-2ec2b084",
                "eni-1b4a65cf"
            ],
            "SubnetIds": [
                "subnet-d6fcaa8d",
                "subnet-7b16de0c"
            ],
            "PrivateDnsEnabled": false,
            "State": "available",
            "ServiceName": "com.amazonaws.us-east-1.elasticloadbalancing",
            "RouteTableIds": [],
            "Groups": [
                {
                    "GroupName": "default",
                    "GroupId": "sg-54e8bf31"
                }
            ],
            "VpcEndpointId": "vpce-0f89a33420c1931d7",
            "VpcEndpointType": "Interface",
            "CreationTimestamp": "2017-09-05T17:55:27.583Z",
            "DnsEntries": [
                {
                    "HostedZoneId": "Z7HUB22UULQXV",
```

```
                    "DnsName": "vpce-0f89a33420c1931d7-
bluzidnv.elasticloadbalancing.us-east-1.vpce.amazonaws.com"
                },
                {
                    "HostedZoneId": "Z7HUB22UULQXV",
                    "DnsName": "vpce-0f89a33420c1931d7-bluzidnv-us-
east-1b.elasticloadbalancing.us-east-1.vpce.amazonaws.com"
                },
                {
                    "HostedZoneId": "Z7HUB22UULQXV",
                    "DnsName": "vpce-0f89a33420c1931d7-bluzidnv-us-
east-1a.elasticloadbalancing.us-east-1.vpce.amazonaws.com"
                }
            ],
            "OwnerId": "123456789012"
        },
        {
            "VpcEndpointId": "vpce-aabbaabbaabbaabba",
            "VpcEndpointType": "GatewayLoadBalancer",
            "VpcId": "vpc-111122223333aabbc",
            "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-
svc-123123a1c43abc123",
            "State": "available",
            "SubnetIds": [
                "subnet-0011aabbcc2233445"
            ],
            "RequesterManaged": false,
            "NetworkInterfaceIds": [
                "eni-01010120203030405"
            ],
            "CreationTimestamp": "2020-11-11T08:06:03.522Z",
            "Tags": [],
            "OwnerId": "123456789012"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS PrivateLink 使用者指南中的概念。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVpcEndpoints。

describe-vpc-peering-connections

以下程式碼範例顯示如何使用 describe-vpc-peering-connections。

AWS CLI

描述您的 VPC 互連連線

此範例說明所有 VPC 對等互連。

命令：

```
aws ec2 describe-vpc-peering-connections
```

輸出：

```
{
    "VpcPeeringConnections": [
        {
            "Status": {
                "Message": "Active",
                "Code": "active"
            },
            "Tags": [
                {
                    "Value": "Peering-1",
                    "Key": "Name"
                }
            ],
            "AccepterVpcInfo": {
                "OwnerId": "111122223333",
                "VpcId": "vpc-1a2b3c4d",
                "CidrBlock": "10.0.1.0/28"
            },
            "VpcPeeringConnectionId": "pcx-11122233",
            "RequesterVpcInfo": {
                "PeeringOptions": {
                    "AllowEgressFromLocalVpcToRemoteClassicLink": false,
                    "AllowEgressFromLocalClassicLinkToRemoteVpc": false
                },
                "OwnerId": "444455556666",
                "VpcId": "vpc-123abc45",
                "CidrBlock": "192.168.0.0/16"
```

```
            }
        },
        {
            "Status": {
                "Message": "Pending Acceptance by 444455556666",
                "Code": "pending-acceptance"
            },
            "Tags": [],
            "RequesterVpcInfo": {
                "PeeringOptions": {
                    "AllowEgressFromLocalVpcToRemoteClassicLink": false,
                    "AllowEgressFromLocalClassicLinkToRemoteVpc": false
                },
                "OwnerId": "444455556666",
                "VpcId": "vpc-11aa22bb",
                "CidrBlock": "10.0.0.0/28"
            },
            "VpcPeeringConnectionId": "pcx-abababab",
            "ExpirationTime": "2014-04-03T09:12:43.000Z",
            "AccepterVpcInfo": {
                "OwnerId": "444455556666",
                "VpcId": "vpc-33cc44dd"
            }
        }
    ]
}
```

描述特定 VPC 對等互連

此範例說明處於待接受狀態的所有 VPC 對等互連。

命令：

```
aws ec2 describe-vpc-peering-connections --filters Name=status-code,Values=pending-acceptance
```

此範例說明標籤為 Owner=Finance 的所有 VPC 對等互連。

命令：

```
aws ec2 describe-vpc-peering-connections --filters Name=tag:Owner,Values=Finance
```

此範例說明您為指定的 VPC vpc-1a2b3c4d 請求的所有 VPC 對等互連。

命令：

```
aws ec2 describe-vpc-peering-connections --filters Name=requester-vpc-info.vpc-id,Values=vpc-1a2b3c4d
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVpcPeeringConnections。

describe-vpcs

以下程式碼範例顯示如何使用 describe-vpcs。

AWS CLI

範例 1：描述所有 VPC

下列 describe-vpcs 範例會擷取有關您 VPC 的詳細資訊。

```
aws ec2 describe-vpcs
```

輸出：

```
{
    "Vpcs": [
        {
            "CidrBlock": "30.1.0.0/16",
            "DhcpOptionsId": "dopt-19edf471",
            "State": "available",
            "VpcId": "vpc-0e9801d129EXAMPLE",
            "OwnerId": "111122223333",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-062c64cfafEXAMPLE",
                    "CidrBlock": "30.1.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": false,
            "Tags": [
                {
```

```
                    "Key": "Name",
                    "Value": "Not Shared"
                }
            ]
        },
        {
            "CidrBlock": "10.0.0.0/16",
            "DhcpOptionsId": "dopt-19edf471",
            "State": "available",
            "VpcId": "vpc-06e4ab6c6cEXAMPLE",
            "OwnerId": "222222222222",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-00b17b4eddEXAMPLE",
                    "CidrBlock": "10.0.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": false,
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Shared VPC"
                }
            ]
        }
    ]
}
```

範例 2：描述指定的 VPC

下列 `describe-vpcs` 範例會擷取指定 VPC 的詳細資訊。

```
aws ec2 describe-vpcs \
    --vpc-ids vpc-06e4ab6c6cEXAMPLE
```

輸出：

```
{
    "Vpcs": [
```

```
        {
            "CidrBlock": "10.0.0.0/16",
            "DhcpOptionsId": "dopt-19edf471",
            "State": "available",
            "VpcId": "vpc-06e4ab6c6cEXAMPLE",
            "OwnerId": "111122223333",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-00b17b4eddEXAMPLE",
                    "CidrBlock": "10.0.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": false,
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Shared VPC"
                }
            ]
        }
    ]
}
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeVpcs。

describe-vpn-connections

以下程式碼範例顯示如何使用 `describe-vpn-connections`。

AWS CLI

範例 1：描述您的 VPN 連線

下列`describe-vpn-connections`範例說明所有 Site-to-Site VPN 連線。

```
aws ec2 describe-vpn-connections
```

輸出：

```
{
    "VpnConnections": [
        {
            "CustomerGatewayConfiguration": "...configuration information...",
            "CustomerGatewayId": "cgw-01234567abcde1234",
            "Category": "VPN",
            "State": "available",
            "Type": "ipsec.1",
            "VpnConnectionId": "vpn-1122334455aabbccd",
            "TransitGatewayId": "tgw-00112233445566aab",
            "Options": {
                "EnableAcceleration": false,
                "StaticRoutesOnly": true,
                "LocalIpv4NetworkCidr": "0.0.0.0/0",
                "RemoteIpv4NetworkCidr": "0.0.0.0/0",
                "TunnelInsideIpVersion": "ipv4"
            },
            "Routes": [],
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "CanadaVPN"
                }
            ],
            "VgwTelemetry": [
                {
                    "AcceptedRouteCount": 0,
                    "LastStatusChange": "2020-07-29T10:35:11.000Z",
                    "OutsideIpAddress": "203.0.113.3",
                    "Status": "DOWN",
                    "StatusMessage": ""
                },
                {
                    "AcceptedRouteCount": 0,
                    "LastStatusChange": "2020-09-02T09:09:33.000Z",
                    "OutsideIpAddress": "203.0.113.5",
                    "Status": "UP",
                    "StatusMessage": ""
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱AWS 《 Site-to-Site VPN 使用者指南》中的 HowSite-to-Site VPN 運作。AWS Site-to-Site

範例 2：描述您可用的 VPN 連線

下列describe-vpn-connections範例說明狀態為 的Site-to-Site VPN 連接available。

```
aws ec2 describe-vpn-connections \
    --filters "Name=state,Values=available"
```

如需詳細資訊，請參閱AWS 《 Site-to-Site VPN 使用者指南》中的 HowSite-to-Site VPN 運作。AWS Site-to-Site

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVpnConnections。

describe-vpn-gateways

以下程式碼範例顯示如何使用 describe-vpn-gateways。

AWS CLI

描述您的虛擬私有閘道

此範例說明您的虛擬私有閘道。

命令：

```
aws ec2 describe-vpn-gateways
```

輸出：

```
{
    "VpnGateways": [
        {
            "State": "available",
            "Type": "ipsec.1",
            "VpnGatewayId": "vgw-f211f09b",
            "VpcAttachments": [
                {
                    "State": "attached",
                    "VpcId": "vpc-98eb5ef5"
```

```
                }
            ]
        },
        {
            "State": "available",
            "Type": "ipsec.1",
            "VpnGatewayId": "vgw-9a4cacf3",
            "VpcAttachments": [
                {
                    "State": "attaching",
                    "VpcId": "vpc-a01106c2"
                }
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVpnGateways。

`detach-classic-link-vpc`

以下程式碼範例顯示如何使用 `detach-classic-link-vpc`。

AWS CLI

從 VPC 取消連結（分離）EC2-Classic 執行個體

此範例會從 VPC vpc-88888888 取消連結執行個體 i-0598c7d356eba48d7。

命令：

```
aws ec2 detach-classic-link-vpc --instance-id i-0598c7d356eba48d7 --vpc-
id vpc-88888888
```

輸出：

```
{
  "Return": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetachClassicLinkVpc。

detach-internet-gateway

以下程式碼範例顯示如何使用 detach-internet-gateway。

AWS CLI

從 VPC 分離網際網路閘道

下列detach-internet-gateway範例會將指定的網際網路閘道與特定 VPC 分離。

```
aws ec2 detach-internet-gateway \
    --internet-gateway-id igw-0d0fb496b3EXAMPLE \
    --vpc-id vpc-0a60eb65b4EXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的網際網路閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetachInternetGateway。

detach-network-interface

以下程式碼範例顯示如何使用 detach-network-interface。

AWS CLI

從執行個體分離網路界面

此範例會將指定的網路介面與指定的執行個體分離。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 detach-network-interface --attachment-id eni-attach-66c4350a
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetachNetworkInterface。

detach-verified-access-trust-provider

以下程式碼範例顯示如何使用 detach-verified-access-trust-provider。

AWS CLI

從執行個體分離信任提供者

下列detach-verified-access-trust-provider範例會將指定的 Verified Access 信任提供者與指定的 Verified Access 執行個體分離。

```
aws ec2 detach-verified-access-trust-provider \
    --verified-access-instance-id vai-0ce000c0b7643abea \
    --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

輸出：

```
{
    "VerifiedAccessTrustProvider": {
        "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
        "Description": "Testing Verified Access",
        "TrustProviderType": "user",
        "UserTrustProviderType": "iam-identity-center",
        "PolicyReferenceName": "idc",
        "CreationTime": "2023-08-25T19:00:38",
        "LastUpdatedTime": "2023-08-25T19:00:38"
    },
    "VerifiedAccessInstance": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "Description": "Testing Verified Access",
        "VerifiedAccessTrustProviders": [],
        "CreationTime": "2023-08-25T18:27:56",
        "LastUpdatedTime": "2023-08-25T18:27:56"
    }
}
```

如需詳細資訊，請參閱 Verified Access 使用者指南中的 Verified Access 執行個體。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetachVerifiedAccessTrustProvider。

detach-volume

以下程式碼範例顯示如何使用 detach-volume。

AWS CLI

從執行個體分離磁碟區

此範例命令會將磁碟區 (vol-049df61146c4d7901) 與其連接的執行個體分離。

命令：

```
aws ec2 detach-volume --volume-id vol-1234567890abcdef0
```

輸出：

```
{
    "AttachTime": "2014-02-27T19:23:06.000Z",
    "InstanceId": "i-1234567890abcdef0",
    "VolumeId": "vol-049df61146c4d7901",
    "State": "detaching",
    "Device": "/dev/sdb"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetachVolume。

detach-vpn-gateway

以下程式碼範例顯示如何使用 detach-vpn-gateway。

AWS CLI

從 VPC 分離虛擬私有閘道

此範例會將指定的虛擬私有閘道與指定的 VPC 分離。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 detach-vpn-gateway --vpn-gateway-id vgw-9a4cacf3 --vpc-id vpc-a01106c2
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetachVpnGateway。

disable-address-transfer

以下程式碼範例顯示如何使用 disable-address-transfer。

AWS CLI

停用彈性 IP 地址傳輸

下列disable-address-transfer範例會停用指定彈性 IP 地址的彈性 IP 地址轉移。

```
aws ec2 disable-address-transfer \
```

```
    --allocation-id eipalloc-09ad461b0d03f6aaf
```

輸出：

```
{
    "AddressTransfer": {
        "PublicIp": "100.21.184.216",
        "AllocationId": "eipalloc-09ad461b0d03f6aaf",
        "AddressTransferStatus": "disabled"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的轉移彈性 IP 地址。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableAddressTransfer。

disable-aws-network-performance-metric-subscription

以下程式碼範例顯示如何使用 disable-aws-network-performance-metric-subscription。

AWS CLI

停用指標訂閱

下列disable-aws-network-performance-metric-subscription範例會停用指定來源和目的地區域之間彙總網路延遲的監控。

```
aws ec2 disable-aws-network-performance-metric-subscription \
    --source us-east-1 \
    --destination eu-west-1 \
    --metric aggregate-latency \
    --statistic p50
```

輸出：

```
{
    "Output": true
}
```

如需詳細資訊，請參閱《 基礎設施效能使用者指南》中的使用 CLI 管理 CloudWatch 訂閱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableAwsNetworkPerformanceMetricSubscription。

disable-ebs-encryption-by-default

以下程式碼範例顯示如何使用 disable-ebs-encryption-by-default。

AWS CLI

預設停用 EBS 加密

下列disable-ebs-encryption-by-default範例預設會停用目前區域中 AWS 您帳戶的 EBS 加密。

```
aws ec2 disable-ebs-encryption-by-default
```

輸出：

```
{
    "EbsEncryptionByDefault": false
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableEbsEncryptionByDefault。

disable-fast-launch

以下程式碼範例顯示如何使用 disable-fast-launch。

AWS CLI

停止快速啟動映像

下列disable-fast-launch範例會停止指定 AMI 的快速啟動，並清除現有的預先佈建快照。

```
aws ec2 disable-fast-launch \
    --image-id ami-01234567890abcedf
```

輸出：

```
{
    "ImageId": "ami-01234567890abcedf",
```

```
    "ResourceType": "snapshot",
    "SnapshotConfiguration": {},
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-01234567890abcedf",
        "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-
a8c6215d-94e6-441b-9272-dbd1f87b07e2",
        "Version": "1"
    },
    "MaxParallelLaunches": 6,
    "OwnerId": "0123456789123",
    "State": "disabling",
    "StateTransitionReason": "Client.UserInitiated",
    "StateTransitionTime": "2022-01-27T22:47:29.265000+00:00"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的設定 Windows AMI 的 EC2 Fast Launch 設定。 Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableFastLaunch。

disable-fast-snapshot-restores

以下程式碼範例顯示如何使用 disable-fast-snapshot-restores。

AWS CLI

停用快速快照還原

下列disable-fast-snapshot-restores範例會停用指定可用區域中指定快照的快速快照還原。

```
aws ec2 disable-fast-snapshot-restores \
    --availability-zones us-east-2a \
    --source-snapshot-ids snap-1234567890abcdef0
```

輸出：

```
{
    "Successful": [
        {
            "SnapshotId": "snap-1234567890abcdef0"
            "AvailabilityZone": "us-east-2a",
```

```
            "State": "disabling",
            "StateTransitionReason": "Client.UserInitiated",
            "OwnerId": "123456789012",
            "EnablingTime": "2020-01-25T23:57:49.602Z"
        }
    ],
    "Unsuccessful": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableFastSnapshotRestores。

disable-image-block-public-access

以下程式碼範例顯示如何使用 disable-image-block-public-access。

AWS CLI

停用封鎖指定區域中 AMIs公開存取

下列disable-image-block-public-access範例會在指定區域中的帳戶層級停用 AMIs的封鎖公開存取。

```
aws ec2 disable-image-block-public-access \
    --region us-east-1
```

輸出：

```
{
    "ImageBlockPublicAccessState": "unblocked"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的封鎖對 AMIs公開存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableImageBlockPublicAccess。

disable-image-deprecation

以下程式碼範例顯示如何使用 disable-image-deprecation。

AWS CLI

取消 AMI 的棄用

下列disable-image-deprecation範例會取消 AMI 的棄用，這會從describe-images輸出中移除 DeprecationTime 欄位。您必須是 AMI 擁有者才能執行此程序。

```
aws ec2 disable-image-deprecation \
    --image-id ami-1234567890abcdef0
```

輸出：

```
{
    "RequestID": "11aabb229-4eac-35bd-99ed-be587EXAMPLE",
    "Return": "true"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的棄用 AMI。 Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableImageDeprecation。

disable-image-deregistration-protection

以下程式碼範例顯示如何使用 disable-image-deregistration-protection。

AWS CLI

停用取消註冊保護

下列disable-image-deregistration-protection範例會停用指定映像的取消註冊保護。

```
aws ec2 disable-image-deregistration-protection \
    --image-id ami-0b1a928a144a74ec9
```

輸出：

```
{
    "Return": "disabled"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的保護 AMI 免於取消註冊。 Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableImageDeregistrationProtection。

disable-image

以下程式碼範例顯示如何使用 disable-image。

AWS CLI

停用 AMI

下列disable-image範例會停用指定的 AMI。

```
aws ec2 disable-image \
    --image-id ami-1234567890abcdef0
```

輸出：

```
{
    "Return": "true"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的停用 AMI。 Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableImage。

disable-ipam-organization-admin-account

以下程式碼範例顯示如何使用 disable-ipam-organization-admin-account。

AWS CLI

停用委派 IPAM 管理員

在某些情況下，您會將 IPAM 與 AWS Organizations 整合。當您這樣做時， AWS Organizations 管理帳戶會將 AWS Organizations 成員帳戶委派為 IPAM 管理員。

在此範例中，您是委派 IPAM 管理員帳戶的 AWS Organizations 管理帳戶，而且您想要將該帳戶停用為 IPAM 管理員。

您可以在提出此請求--region時使用 的任何 AWS 區域。您不需要使用最初委派管理員的區域、建立 IPAM 的區域或 IPAM 操作區域。如果您停用委派的管理員帳戶，您可以隨時重新啟用該帳戶，或將新帳戶委派為 IPAM 管理員。

下列disable-ipam-organization-admin-account範例會停用您 AWS 帳戶中的委派 IPAM 管理員。

```
aws ec2 disable-ipam-organization-admin-account \
    --delegated-admin-account-id 320805250157 \
    --region ap-south-1
```

輸出：

```
{
    "Success": true
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的將 IPAM 與 AWS Organization 中的帳戶整合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableIpamOrganizationAdminAccount。

disable-serial-console-access

以下程式碼範例顯示如何使用 disable-serial-console-access。

AWS CLI

停用您帳戶的 EC2 序列主控台存取權

下列disable-serial-console-access範例會停用對序列主控台的帳戶存取。

```
aws ec2 disable-serial-console-access
```

輸出：

```
{
    "SerialConsoleAccessEnabled": false
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 EC2 序列主控台。 Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableSerialConsoleAccess。

disable-snapshot-block-public-access

以下程式碼範例顯示如何使用 disable-snapshot-block-public-access。

AWS CLI

停用封鎖快照的公開存取

下列disable-snapshot-block-public-access範例會停用快照的封鎖公開存取，以允許公開共用快照。

```
aws ec2 disable-snapshot-block-public-access
```

輸出：

```
{
    "State": "unblocked"
}
```

如需詳細資訊，請參閱《Amazon EBS 使用者指南》中的封鎖快照的公開存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableSnapshotBlockPublicAccess。

disable-transit-gateway-route-table-propagation

以下程式碼範例顯示如何使用 disable-transit-gateway-route-table-propagation。

AWS CLI

停用傳輸閘道連接，將路由傳播到指定的傳播路由表

下列disable-transit-gateway-route-table-propagation範例會停用指定的附件，將路由傳播至指定的傳播路由表。

```
aws ec2 disable-transit-gateway-route-table-propagation \
    --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \
    --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE
```

輸出：

```
{
    "Propagation": {
        "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
        "ResourceId": "vpc-4d7de228",
        "ResourceType": "vpc",
        "TransitGatewayRouteTableId": "tgw-rtb-0a823edbdeEXAMPLE",
```

```
        "State": "disabled"
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的 Transit Gateway 路由表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableTransitGatewayRouteTablePropagation。

`disable-vgw-route-propagation`

以下程式碼範例顯示如何使用 `disable-vgw-route-propagation`。

AWS CLI

停用路由傳播

此範例會停用指定的虛擬私有閘道，將靜態路由傳播到指定的路由表。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 disable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-
id vgw-9a4cacf3
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableVgwRoutePropagation。

`disable-vpc-classic-link-dns-support`

以下程式碼範例顯示如何使用 `disable-vpc-classic-link-dns-support`。

AWS CLI

停用 VPC 的 ClassicLink DNS 支援

此範例會停用 的 ClassicLink DNS 支援`vpc-88888888`。

命令：

```
aws ec2 disable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

輸出：

```
{
  "Return": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableVpcClassicLinkDnsSupport。

disable-vpc-classic-link

以下程式碼範例顯示如何使用 disable-vpc-classic-link。

AWS CLI

停用 VPC 的 ClassicLink

此範例會停用 vpc-8888888 的 ClassicLink。

命令：

```
aws ec2 disable-vpc-classic-link --vpc-id vpc-88888888
```

輸出：

```
{
  "Return": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableVpcClassicLink。

disassociate-address

以下程式碼範例顯示如何使用 disassociate-address。

AWS CLI

在 EC2-Classic 中取消彈性 IP 地址的關聯

此範例會在 EC2-Classic 中取消彈性 IP 地址與執行個體的關聯。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 disassociate-address --public-ip 198.51.100.0
```

在 EC2-VPC 中取消彈性 IP 地址的關聯

此範例會在 VPC 中取消彈性 IP 地址與執行個體的關聯。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 disassociate-address --association-id eipassoc-2bebb745
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DisassociateAddress。

disassociate-client-vpn-target-network

以下程式碼範例顯示如何使用 disassociate-client-vpn-target-network。

AWS CLI

取消網路與 Client VPN 端點的關聯

下列disassociate-client-vpn-target-network範例會取消與指定 Client VPN 端點cvpn-assoc-12312312312312312關聯 ID 相關聯的目標網路的關聯。

```
aws ec2 disassociate-client-vpn-target-network \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --association-id cvpn-assoc-12312312312312312
```

輸出：

```
{
    "AssociationId": "cvpn-assoc-12312312312312312",
    "Status": {
        "Code": "disassociating"
    }
}
```

如需詳細資訊，請參閱 AWS Client VPN 管理員指南中的目標網路。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateClientVpnTargetNetwork。

disassociate-iam-instance-profile

以下程式碼範例顯示如何使用 disassociate-iam-instance-profile。

AWS CLI

取消與 IAM 執行個體描述檔的關聯

此範例會取消 IAM 執行個體描述檔與關聯 ID 的關聯iip-assoc-05020b59952902f5f。

命令：

```
aws ec2 disassociate-iam-instance-profile --association-id iip-
assoc-05020b59952902f5f
```

輸出：

```
{
  "IamInstanceProfileAssociation": {
      "InstanceId": "i-123456789abcde123",
      "State": "disassociating",
      "AssociationId": "iip-assoc-05020b59952902f5f",
      "IamInstanceProfile": {
          "Id": "AIPAI5IVIHMFFYY2DKV5Y",
          "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"
      }
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateIamInstanceProfile。

disassociate-instance-event-window

以下程式碼範例顯示如何使用 disassociate-instance-event-window。

AWS CLI

範例 1：取消一或多個執行個體與事件時段的關聯

下列disassociate-instance-event-window範例會取消一或多個執行個體與事件時段的關聯。指定 instance-event-window-id 參數以指定事件視窗。若要解除執行個體的關聯，請指定 association-target 參數，並針對參數值指定一個或多個執行個體 ID。

```
aws ec2 disassociate-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

輸出：

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

如需事件時段限制，請參閱《Amazon EC2 使用者指南》的排程事件一節中的考量事項。

範例 2：取消執行個體標籤與事件時段的關聯

下列disassociate-instance-event-window範例會取消執行個體標籤與事件視窗的關聯。指定 instance-event-window-id 參數以指定事件視窗。若要解除執行個體標籤的關聯，請指定 association-target 參數，並針對參數值指定一個或多個標籤。

```
aws ec2 disassociate-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"
```

輸出：

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
```

```
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

如需事件時段限制，請參閱《Amazon EC2 使用者指南》的排程事件一節中的考量事項。

範例 3：取消專用主機與事件時段的關聯

下列disassociate-instance-event-window範例會取消專用主機與事件時段的關聯。指定 instance-event-window-id 參數以指定事件視窗。若要解除專用執行個體的關聯，請指定 association-target 參數，並針對參數值指定一個或多個專用執行個體 ID。

```
aws ec2 disassociate-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target DedicatedHostIds=h-029fa35a02b99801d
```

輸出：

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

如需事件時段限制，請參閱《Amazon EC2 使用者指南》的排程事件一節中的考量事項。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateInstanceEventWindow。

disassociate-ipam-resource-discovery

以下程式碼範例顯示如何使用 disassociate-ipam-resource-discovery。

AWS CLI

取消資源探索與 IPAM 的關聯

在此範例中，您是 IPAM 委派的管理員帳戶，而且您想要取消 IPAM 資源探索與 IPAM 的關聯。您執行 describe 命令，並注意到 "ResourceDiscoveryStatus": "not-found"和 想要將其與 IPAM 取消關聯，為其他關聯騰出空間。

下列disassociate-ipam-resource-discovery範例會取消您 AWS 帳戶中 IPAM 資源探索的關聯。

```
aws ec2 disassociate-ipam-resource-discovery \
    --ipam-resource-discovery-association-id ipam-res-disco-assoc-04382a6346357cf82
 \
    --region us-east-1
```

輸出：

```
{
    "IpamResourceDiscoveryAssociation": {
        "OwnerId": "320805250157",
        "IpamResourceDiscoveryAssociationId": "ipam-res-disco-
assoc-04382a6346357cf82",
        "IpamResourceDiscoveryAssociationArn":
 "arn:aws:ec2::320805250157:ipam-resource-discovery-association/ipam-res-disco-
assoc-04382a6346357cf82",
        "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
        "IpamId": "ipam-005f921c17ebd5107",
        "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
        "IpamRegion": "us-east-1",
        "IsDefault": false,
        "ResourceDiscoveryStatus": "not-found",
        "State": "disassociate-in-progress"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的將 IPAM 與組織外部的帳戶整合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateIpamResourceDiscovery。

disassociate-nat-gateway-address

以下程式碼範例顯示如何使用 disassociate-nat-gateway-address。

AWS CLI

取消彈性 IP 地址與公有 NAT 閘道的關聯

下列disassociate-nat-gateway-address範例會將指定的彈性 IP 地址與指定的公有 NAT 閘道取消關聯。

```
aws ec2 disassociate-nat-gateway-address \
    --nat-gateway-id nat-1234567890abcdef0 \
    --association-ids eipassoc-0f96bdca17EXAMPLE
```

輸出：

```
{
    "NatGatewayId": "nat-1234567890abcdef0",
    "NatGatewayAddresses": [
        {
            "AllocationId": "eipalloc-0be6ecac95EXAMPLE",
            "NetworkInterfaceId": "eni-09cc4b2558794f7f9",
            "PrivateIp": "10.0.0.74",
            "PublicIp": "3.211.231.218",
            "AssociationId": "eipassoc-0f96bdca17EXAMPLE",
            "IsPrimary": false,
            "Status": "disassociating"
        }
    ]
}
```

如需詳細資訊，請參閱 《Amazon VPC 使用者指南》中的 NAT 閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateNatGatewayAddress。

disassociate-route-table

以下程式碼範例顯示如何使用 disassociate-route-table。

AWS CLI

取消路由表的關聯

此範例會取消指定路由表與指定子網路的關聯。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 disassociate-route-table --association-id rtbassoc-781d0d1a
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateRouteTable。

disassociate-security-group-vpc

以下程式碼範例顯示如何使用 disassociate-security-group-vpc。

AWS CLI

取消安全群組與 VPC 的關聯

下列disassociate-security-group-vpc範例會將指定的安全群組與指定的 VPC 取消關聯。

```
aws ec2 disassociate-security-group-vpc \
    --group-id sg-04dbb43907d3f8a78 \
    --vpc-id vpc-0bf4c2739bc05a694
```

輸出：

```
{
    "State": "disassociating"
}
```

如需詳細資訊，請參閱《Amazon VPCs》中的將安全群組與多個 VPC 建立關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateSecurityGroupVpc。

disassociate-subnet-cidr-block

以下程式碼範例顯示如何使用 disassociate-subnet-cidr-block。

AWS CLI

取消 IPv6 CIDR 區塊與子網路的關聯

此範例使用 CIDR 區塊的關聯 ID，取消 IPv6 CIDR 區塊與子網路的關聯。

命令：

```
aws ec2 disassociate-subnet-cidr-block --association-id subnet-cidr-assoc-3aa54053
```

輸出：

```
{
  "SubnetId": "subnet-5f46ec3b",
  "Ipv6CidrBlockAssociation": {
      "Ipv6CidrBlock": "2001:db8:1234:1a00::/64",
      "AssociationId": "subnet-cidr-assoc-3aa54053",
      "Ipv6CidrBlockState": {
          "State": "disassociating"
      }
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateSubnetCidrBlock。

disassociate-transit-gateway-multicast-domain

以下程式碼範例顯示如何使用 disassociate-transit-gateway-multicast-domain。

AWS CLI

取消子網路與多點傳送網域的關聯

下列disassociate-transit-gateway-multicast-domain範例會取消子網路與指定多點傳送網域的關聯。

```
aws ec2 disassociate-transit-gateway-multicast-domain \
    --transit-gateway-attachment-id tgw-attach-070e571cd1EXAMPLE \
    --subnet-id subnet-000de86e3bEXAMPLE \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

輸出：

```
{
    "Associations": {
```

```
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef7EXAMPLE",
        "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",
        "ResourceId": "vpc-7EXAMPLE",
        "ResourceType": "vpc",
        "Subnets": [
            {
                "SubnetId": "subnet-000de86e3bEXAMPLE",
                "State": "disassociating"
            }
        ]
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的多點傳送網域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateTransitGatewayMulticastDomain。

disassociate-transit-gateway-route-table

以下程式碼範例顯示如何使用 disassociate-transit-gateway-route-table。

AWS CLI

取消傳輸閘道路由表與資源連接的關聯

下列disassociate-transit-gateway-route-table範例會將指定的附件與傳輸閘道路由表取消關聯。

```
aws ec2 disassociate-transit-gateway-route-table \
    --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE \
    --transit-gateway-attachment-id tgw-attach-08e0bc912cEXAMPLE
```

輸出：

```
{
    "Association": {
        "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
        "TransitGatewayAttachmentId": "tgw-attach-08e0bc912cEXAMPLE",
        "ResourceId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
        "ResourceType": "direct-connect-gateway",
        "State": "disassociating"
```

```
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的 Transit Gateway 路由表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateTransitGatewayRouteTable。

`disassociate-vpc-cidr-block`

以下程式碼範例顯示如何使用 `disassociate-vpc-cidr-block`。

AWS CLI

取消 IPv6 CIDR 區塊與 VPC 的關聯

此範例使用 CIDR 區塊的關聯 ID，取消 IPv6 CIDR 區塊與 VPC 的關聯。

命令：

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-eca54085
```

輸出：

```
{
  "Ipv6CidrBlockAssociation": {
      "Ipv6CidrBlock": "2001:db8:1234:1a00::/56",
      "AssociationId": "vpc-cidr-assoc-eca54085",
      "Ipv6CidrBlockState": {
          "State": "disassociating"
      }
  },
  "VpcId": "vpc-a034d6c4"
}
```

取消 IPv4 CIDR 區塊與 VPC 的關聯

此範例會取消 IPv4 CIDR 區塊與 VPC 的關聯。

命令：

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-0287ac6b
```

輸出：

```
{
  "CidrBlockAssociation": {
      "AssociationId": "vpc-cidr-assoc-0287ac6b",
      "CidrBlock": "172.18.0.0/16",
      "CidrBlockState": {
          "State": "disassociating"
      }
  },
  "VpcId": "vpc-27621243"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateVpcCidrBlock。

enable-address-transfer

以下程式碼範例顯示如何使用 enable-address-transfer。

AWS CLI

啟用彈性 IP 地址傳輸

下列enable-address-transfer範例會啟用指定彈性 IP 地址的彈性 IP 地址轉移到指定帳戶。

```
aws ec2 enable-address-transfer \
    --allocation-id eipalloc-09ad461b0d03f6aaf \
    --transfer-account-id 123456789012
```

輸出：

```
{
     "AddressTransfer": {
        "PublicIp": "100.21.184.216",
        "AllocationId": "eipalloc-09ad461b0d03f6aaf",
        "TransferAccountId": "123456789012",
        "TransferOfferExpirationTimestamp": "2023-02-22T20:51:01.000Z",
        "AddressTransferStatus": "pending"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的轉移彈性 IP 地址。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableAddressTransfer。

enable-aws-network-performance-metric-subscription

以下程式碼範例顯示如何使用 enable-aws-network-performance-metric-subscription。

AWS CLI

啟用指標訂閱

下列enable-aws-network-performance-metric-subscription範例可監控指定來源和目的地區域之間的彙總網路延遲。

```
aws ec2 enable-aws-network-performance-metric-subscription \
    --source us-east-1 \
    --destination eu-west-1 \
    --metric aggregate-latency \
    --statistic p50
```

輸出：

```
{
    "Output": true
}
```

如需詳細資訊，請參閱《 基礎設施效能使用者指南》中的管理訂閱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableAwsNetworkPerformanceMetricSubscription。

enable-ebs-encryption-by-default

以下程式碼範例顯示如何使用 enable-ebs-encryption-by-default。

AWS CLI

預設啟用 EBS 加密

下列enable-ebs-encryption-by-default範例預設會為目前區域中 AWS 的帳戶啟用 EBS 加密。

```
aws ec2 enable-ebs-encryption-by-default
```

輸出：

```
{
    "EbsEncryptionByDefault": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableEbsEncryptionByDefault。

enable-fast-launch

以下程式碼範例顯示如何使用 enable-fast-launch。

AWS CLI

開始快速啟動映像

下列enable-fast-launch範例會將指定的 AMI 設定為快速啟動，並將平行執行個體的數量上限設定為 6。用於預先佈建 AMI 的資源類型設定 snapshot，這也是預設值。

```
aws ec2 enable-fast-launch \
    --image-id ami-01234567890abcedf \
    --max-parallel-launches 6 \
    --resource-type snapshot
```

輸出：

```
{
    "ImageId": "ami-01234567890abcedf",
    "ResourceType": "snapshot",
    "SnapshotConfiguration": {
        "TargetResourceCount": 10
    },
    "LaunchTemplate": {},
    "MaxParallelLaunches": 6,
    "OwnerId": "0123456789123",
    "State": "enabling",
    "StateTransitionReason": "Client.UserInitiated",
    "StateTransitionTime": "2022-01-27T22:16:03.199000+00:00"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的設定 Windows AMI 的 EC2 Fast Launch 設定。 Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableFastLaunch。

enable-fast-snapshot-restores

以下程式碼範例顯示如何使用 enable-fast-snapshot-restores。

AWS CLI

啟用快速快照還原

下列enable-fast-snapshot-restores範例會啟用指定可用區域中指定快照的快速快照還原。

```
aws ec2 enable-fast-snapshot-restores \
    --availability-zones us-east-2a us-east-2b \
    --source-snapshot-ids snap-1234567890abcdef0
```

輸出：

```
{
    "Successful": [
        {
            "SnapshotId": "snap-1234567890abcdef0"
            "AvailabilityZone": "us-east-2a",
            "State": "enabling",
            "StateTransitionReason": "Client.UserInitiated",
            "OwnerId": "123456789012",
            "EnablingTime": "2020-01-25T23:57:49.602Z"
        },
        {
            "SnapshotId": "snap-1234567890abcdef0"
            "AvailabilityZone": "us-east-2b",
            "State": "enabling",
            "StateTransitionReason": "Client.UserInitiated",
            "OwnerId": "123456789012",
            "EnablingTime": "2020-01-25T23:57:49.596Z"
        }
    ],
    "Unsuccessful": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableFastSnapshotRestores。

enable-image-block-public-access

以下程式碼範例顯示如何使用 enable-image-block-public-access。

AWS CLI

為指定區域中AMIs 啟用封鎖公開存取

下列enable-image-block-public-access範例會在指定區域中的帳戶層級啟用 AMIs的封鎖公開存取。

```
aws ec2 enable-image-block-public-access \
    --region us-east-1 \
    --image-block-public-access-state block-new-sharing
```

輸出：

```
{
    "ImageBlockPublicAccessState": "block-new-sharing"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的封鎖對 AMIs公開存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableImageBlockPublicAccess。

enable-image-deprecation

以下程式碼範例顯示如何使用 enable-image-deprecation。

AWS CLI

棄用 AMI

下列enable-image-deprecation範例會在特定日期和時間取代 AMI。如果您指定秒數值，Amazon EC2 會將秒數四捨五入到最接近的分鐘。您必須是 AMI 擁有者才能執行此程序。

```
aws ec2 enable-image-deprecation \
    --image-id ami-1234567890abcdef0 \
```

```
    --deprecate-at '2022-10-15T13:17:12.000Z'
```

輸出：

```
{
    "RequestID": "59dbff89-35bd-4eac-99ed-be587EXAMPLE",
    "Return": "true"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的棄用 AMI。 Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableImageDeprecation。

enable-image-deregistration-protection

以下程式碼範例顯示如何使用 enable-image-deregistration-protection。

AWS CLI

啟用取消註冊保護

下列enable-image-deregistration-protection範例會啟用指定映像的取消註冊保護。

```
aws ec2 enable-image-deregistration-protection \
    --image-id ami-0b1a928a144a74ec9
```

輸出：

```
{
    "Return": "enabled-without-cooldown"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的保護 EC2 AMI 免於取消註冊。 Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableImageDeregistrationProtection。

enable-image

以下程式碼範例顯示如何使用 enable-image。

AWS CLI

啟用 AMI

下列enable-image範例會啟用指定的 AMI。

```
aws ec2 enable-image \
    --image-id ami-1234567890abcdef0
```

輸出：

```
{
    "Return": "true"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的停用 AMI。 Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableImage。

enable-ipam-organization-admin-account

以下程式碼範例顯示如何使用 enable-ipam-organization-admin-account。

AWS CLI

與 AWS Organizations 整合並將成員帳戶委派為 IPAM 帳戶

下列enable-ipam-organization-admin-account範例將 IPAM 與 AWS Organizations 整合，並將成員帳戶委派為 IPAM 帳戶。

```
aws ec2 enable-ipam-organization-admin-account \
    --delegated-admin-account-id 320805250157
```

輸出：

```
{
    "Success": true
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的將 IPAM 與 AWS Organizations 整合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableIpamOrganizationAdminAccount。

enable-reachability-analyzer-organization-sharing

以下程式碼範例顯示如何使用 enable-reachability-analyzer-organization-sharing。

AWS CLI

啟用 Reachability Analyzer 的受信任存取

下列enable-reachability-analyzer-organization-sharing範例會啟用 Reachability Analyzer 的受信任存取。

```
aws ec2 enable-reachability-analyzer-organization-sharing
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Reachability Analyzer 使用者指南中的跨帳戶分析。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableReachabilityAnalyzerOrganizationSharing。

enable-serial-console-access

以下程式碼範例顯示如何使用 enable-serial-console-access。

AWS CLI

為您的帳戶啟用對序列主控台的存取

下列enable-serial-console-access範例可讓帳戶存取序列主控台。

```
aws ec2 enable-serial-console-access
```

輸出：

```
{
    "SerialConsoleAccessEnabled": true
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 EC2 序列主控台。 Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableSerialConsoleAccess。

enable-snapshot-block-public-access

以下程式碼範例顯示如何使用 enable-snapshot-block-public-access。

AWS CLI

啟用快照的封鎖公開存取

下列enable-snapshot-block-public-access範例會封鎖快照的所有公開共用。

```
aws ec2 enable-snapshot-block-public-access \
    --state block-all-sharing
```

輸出：

```
{
    "State": "block-all-sharing"
}
```

如需詳細資訊，請參閱《Amazon EBS 使用者指南》中的封鎖快照的公開存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableSnapshotBlockPublicAccess。

enable-transit-gateway-route-table-propagation

以下程式碼範例顯示如何使用 enable-transit-gateway-route-table-propagation。

AWS CLI

若要讓傳輸閘道連接將路由傳播到指定的傳播路由表

下列enable-transit-gateway-route-table-propagation範例可讓指定的附件將路由傳播到指定的傳播路由表。

```
aws ec2 enable-transit-gateway-route-table-propagation \
    --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \
    --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE
```

輸出：

```
{
    "Propagation": {
        "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
        "ResourceId": "vpc-4d7de228",
        "ResourceType": "vpc",
        "TransitGatewayRouteTableId": "tgw-rtb-0a823edbdeEXAMPLE",
        "State": "disabled"
    }
}
```

如需詳細資訊，請參閱《傳輸閘道指南》中的傳輸閘道路由表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableTransitGatewayRouteTablePropagation。

enable-vgw-route-propagation

以下程式碼範例顯示如何使用 enable-vgw-route-propagation。

AWS CLI

啟用路由傳播

此範例可讓指定的虛擬私有閘道將靜態路由傳播至指定的路由表。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 enable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-
id vgw-9a4cacf3
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableVgwRoutePropagation。

enable-volume-io

以下程式碼範例顯示如何使用 enable-volume-io。

AWS CLI

為磁碟區啟用 I/O

此範例會在磁碟區 上啟用 I/Ovol-1234567890abcdef0。

命令：

```
aws ec2 enable-volume-io --volume-id vol-1234567890abcdef0
```

輸出：

```
{
  "Return": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableVolumeIo。

enable-vpc-classic-link-dns-support

以下程式碼範例顯示如何使用 enable-vpc-classic-link-dns-support。

AWS CLI

啟用 VPC 的 ClassicLink DNS 支援

此範例會啟用 的 ClassicLink DNS 支援vpc-88888888。

命令：

```
aws ec2 enable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

輸出：

```
{
  "Return": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableVpcClassicLinkDnsSupport。

enable-vpc-classic-link

以下程式碼範例顯示如何使用 enable-vpc-classic-link。

AWS CLI

啟用 ClassicLink 的 VPC

此範例為 ClassicLink 啟用 vpc-8888888。

命令：

```
aws ec2 enable-vpc-classic-link --vpc-id vpc-88888888
```

輸出：

```
{
  "Return": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableVpcClassicLink。

export-client-vpn-client-certificate-revocation-list

以下程式碼範例顯示如何使用 export-client-vpn-client-certificate-revocation-list。

AWS CLI

匯出用戶端憑證撤銷清單

下列export-client-vpn-client-certificate-revocation-list範例匯出指定 Client VPN 端點的用戶端憑證撤銷清單。在此範例中，輸出會以文字格式傳回，以便於讀取。

```
aws ec2 export-client-vpn-client-certificate-revocation-list \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --output text
```

輸出：

```
-----BEGIN X509 CRL-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
```

```
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END X509 CRL-----
STATUS      pending
```

如需詳細資訊，請參閱 Client VPN 管理員指南中的用戶端憑證撤銷清單。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ExportClientVpnClientCertificateRevocationList。

export-client-vpn-client-configuration

以下程式碼範例顯示如何使用 export-client-vpn-client-configuration。

AWS CLI

匯出用戶端組態

下列export-client-vpn-client-configuration範例匯出指定 Client VPN 端點的用戶端組態。在此範例中，輸出會以文字格式傳回，以便於讀取。

```
aws ec2 export-client-vpn-client-configuration \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --output text
```

輸出：

```
client
dev tun
proto udp
remote cvpn-endpoint-123456789123abcde.prod.clientvpn.ap-south-1.amazonaws.com 443
remote-random-hostname
resolv-retry infinite
nobind
persist-key
persist-tun
```

```
remote-cert-tls server
cipher AES-256-GCM
verb 3
<ca>
-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----
</ca>
reneg-sec 0
```

如需詳細資訊，請參閱 Client VPN 管理員指南中的 Client VPN 端點組態檔案匯出。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ExportClientVpnClientConfiguration。

export-image

以下程式碼範例顯示如何使用 export-image。

AWS CLI

從 AMI 匯出 VM

下列export-image範例會以指定的格式，將指定的 AMI 匯出至指定的儲存貯體。

```
aws ec2 export-image \
    --image-id ami-1234567890abcdef0 \
    --disk-image-format VMDK \
    --s3-export-location S3Bucket=my-export-bucket,S3Prefix=exports/
```

輸出：

```
{
    "DiskImageFormat": "vmdk",
    "ExportImageTaskId": "export-ami-1234567890abcdef0"
    "ImageId": "ami-1234567890abcdef0",
    "RoleName": "vmimport",
    "Progress": "0",
    "S3ExportLocation": {
        "S3Bucket": "my-export-bucket",
        "S3Prefix": "exports/"
    },
    "Status": "active",
    "StatusMessage": "validating"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ExportImage。

get-associated-ipv6-pool-cidrs

以下程式碼範例顯示如何使用 get-associated-ipv6-pool-cidrs。

AWS CLI

取得 IPv6 地址集區的關聯

下列get-associated-ipv6-pool-cidrs範例會取得指定 IPv6 地址集區的關聯。

```
aws ec2 get-associated-ipv6-pool-cidrs \
    --pool-id ipv6pool-ec2-012345abc12345abc
```

輸出：

```
{
    "Ipv6CidrAssociations": [
        {
            "Ipv6Cidr": "2001:db8:1234:1a00::/56",
            "AssociatedResource": "vpc-111111222222333ab"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAssociatedIpv6PoolCidrs。

get-aws-network-performance-data

以下程式碼範例顯示如何使用 get-aws-network-performance-data。

AWS CLI

取得網路效能資料

下列get-aws-network-performance-data範例會擷取指定期間內指定區域之間的網路效能相關資料。

```
aws ec2 get-aws-network-performance-data \
    --start-time 2022-10-26T12:00:00.000Z \
    --end-time 2022-10-26T12:30:00.000Z \
    --data-queries Id=my-query,Source=us-east-1,Destination=eu-
west-1,Metric=aggregate-latency,Statistic=p50,Period=five-minutes
```

輸出：

```
{
    "DataResponses": [
        {
            "Id": "my-query",
            "Source": "us-east-1",
            "Destination": "eu-west-1",
            "Metric": "aggregate-latency",
            "Statistic": "p50",
            "Period": "five-minutes",
            "MetricPoints": [
                {
                    "StartDate": "2022-10-26T12:00:00+00:00",
                    "EndDate": "2022-10-26T12:05:00+00:00",
                    "Value": 62.44349,
                    "Status": "OK"
                },
                {
                    "StartDate": "2022-10-26T12:05:00+00:00",
                    "EndDate": "2022-10-26T12:10:00+00:00",
                    "Value": 62.483498,
                    "Status": "OK"
                },
                {
                    "StartDate": "2022-10-26T12:10:00+00:00",
```

```
                    "EndDate": "2022-10-26T12:15:00+00:00",
                    "Value": 62.51248,
                    "Status": "OK"
                },
                {
                    "StartDate": "2022-10-26T12:15:00+00:00",
                    "EndDate": "2022-10-26T12:20:00+00:00",
                    "Value": 62.635475,
                    "Status": "OK"
                },
                {
                    "StartDate": "2022-10-26T12:20:00+00:00",
                    "EndDate": "2022-10-26T12:25:00+00:00",
                    "Value": 62.733974,
                    "Status": "OK"
                },
                {
                    "StartDate": "2022-10-26T12:25:00+00:00",
                    "EndDate": "2022-10-26T12:30:00+00:00",
                    "Value": 62.773975,
                    "Status": "OK"
                },
                {
                    "StartDate": "2022-10-26T12:30:00+00:00",
                    "EndDate": "2022-10-26T12:35:00+00:00",
                    "Value": 62.75349,
                    "Status": "OK"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《 基礎設施效能使用者指南》中的監控網路效能。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAwsNetworkPerformanceData。

get-capacity-reservation-usage

以下程式碼範例顯示如何使用 get-capacity-reservation-usage。

AWS CLI

檢視跨 AWS 帳戶的容量保留用量

下列get-capacity-reservation-usage範例顯示指定容量保留的用量資訊。

```
aws ec2 get-capacity-reservation-usage \
    --capacity-reservation-id cr-1234abcd56EXAMPLE
```

輸出：

```
{
    "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
    "InstanceUsages": [
        {
            "UsedInstanceCount": 1,
            "AccountId": "123456789012"
        }
    ],
    "AvailableInstanceCount": 4,
    "TotalInstanceCount": 5,
    "State": "active",
    "InstanceType": "t2.medium"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的共用容量保留。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCapacityReservationUsage。

get-coip-pool-usage

以下程式碼範例顯示如何使用 get-coip-pool-usage。

AWS CLI

取得客戶擁有的 IP 地址集區用量

下列get-coip-pool-usage範例會取得指定客戶擁有 IP 地址集區的用量詳細資訊。

```
aws ec2 get-coip-pool-usage \
    --pool-id ipv4pool-coip-123a45678bEXAMPLE
```

輸出：

```
{
```

```
    "CoipPoolId": "ipv4pool-coip-123a45678bEXAMPLE",
    "CoipAddressUsages": [
        {
            "CoIp": "0.0.0.0"
        },
        {
            "AllocationId": "eipalloc-123ab45c6dEXAMPLE",
            "AwsAccountId": "123456789012",
            "CoIp": "0.0.0.0"
        },
        {
            "AllocationId": "eipalloc-123ab45c6dEXAMPLE",
            "AwsAccountId": "123456789111",
            "CoIp": "0.0.0.0"
        }
    ],
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE"
}
```

如需詳細資訊，請參閱 AWS Outposts 機架使用者指南中的客戶擁有的 IP 地址。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCoipPoolUsage。

get-console-output

以下程式碼範例顯示如何使用 get-console-output。

AWS CLI

範例 1：取得主控台輸出

下列get-console-output範例會取得指定 Linux 執行個體的主控台輸出。

```
aws ec2 get-console-output \
    --instance-id i-1234567890abcdef0
```

輸出：

```
{
    "InstanceId": "i-1234567890abcdef0",
    "Timestamp": "2013-07-25T21:23:53.000Z",
    "Output": "..."
```

```
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的執行個體主控台輸出。

範例 2：取得最新的主控台輸出

下列get-console-output範例會取得指定 Linux 執行個體的最新主控台輸出。

```
aws ec2 get-console-output \
    --instance-id i-1234567890abcdef0 \
    --latest \
    --output text
```

輸出：

```
i-1234567890abcdef0 [    0.000000] Command line: root=LABEL=/ console=tty1
 console=ttyS0 selinux=0 nvme_core.io_timeout=4294967295
[    0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point
 registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
...
Cloud-init v. 0.7.6 finished at Wed, 09 May 2018 19:01:13 +0000. Datasource
 DataSourceEc2.  Up 21.50 seconds
Amazon Linux AMI release 2018.03
Kernel 4.14.26-46.32.amzn1.x
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的執行個體主控台輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetConsoleOutput。

get-console-screenshot

以下程式碼範例顯示如何使用 get-console-screenshot。

AWS CLI

擷取執行中執行個體的螢幕擷取畫面

下列get-console-screenshot範例會以 .jpg 格式擷取指定執行個體的螢幕擷取畫面。螢幕擷取畫面會以 Base64-encoded字串傳回。

```
aws ec2 get-console-screenshot \
    --instance-id i-1234567890abcdef0
```

輸出：

```
{
    "ImageData": "997987/8kgj49ikjhewkwwe0008084EXAMPLE",
    "InstanceId": "i-1234567890abcdef0"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetConsoleScreenshot。

get-default-credit-specification

以下程式碼範例顯示如何使用 get-default-credit-specification。

AWS CLI

描述預設點數選項

下列get-default-credit-specification範例說明 T2 執行個體的預設點數選項。

```
aws ec2 get-default-credit-specification \
    --instance-family t2
```

輸出：

```
{
    "InstanceFamilyCreditSpecification": {
        "InstanceFamily": "t2",
        "CpuCredits": "standard"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDefaultCreditSpecification。

get-ebs-default-kms-key-id

以下程式碼範例顯示如何使用 get-ebs-default-kms-key-id。

AWS CLI

描述 EBS 加密的預設 CMK

下列get-ebs-default-kms-key-id範例說明 AWS 帳戶 EBS 加密的預設 CMK。

```
aws ec2 get-ebs-default-kms-key-id
```

輸出會顯示 EBS 加密的預設 CMK，這是具有別名 的 AWS 受管 CMKalias/aws/ebs。

```
{
    "KmsKeyId": "alias/aws/ebs"
}
```

下列輸出顯示 EBS 加密的自訂 CMK。

```
{
    "KmsKeyId": "arn:aws:kms:us-
west-2:123456789012:key/0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetEbsDefaultKmsKeyId。

get-ebs-encryption-by-default

以下程式碼範例顯示如何使用 get-ebs-encryption-by-default。

AWS CLI

描述是否預設啟用 EBS 加密

下列get-ebs-encryption-by-default範例指出目前區域中您的帳戶是否預設啟用 EBS AWS 加密。

```
aws ec2 get-ebs-encryption-by-default
```

下列輸出表示預設停用 EBS 加密。

```
{
```

```
    "EbsEncryptionByDefault": false
}
```

下列輸出表示預設已啟用 EBS 加密。

```
{
    "EbsEncryptionByDefault": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetEbsEncryptionByDefault。

get-flow-logs-integration-template

以下程式碼範例顯示如何使用 get-flow-logs-integration-template。

AWS CLI

建立 CloudFormation 範本以自動化 VPC 流程日誌與 Amazon Athena 的整合

下列get-flow-logs-integration-template範例會建立 CloudFormation 範本，以自動化 VPC 流程日誌與 Amazon Athena 的整合。

Linux：

```
aws ec2 get-flow-logs-integration-template \
    --flow-log-id fl-1234567890abcdef0 \
    --config-delivery-s3-destination-arn arn:aws:s3:::amzn-s3-demo-bucket \
    --integrate-services
 AthenaIntegrations='[{IntegrationResultS3DestinationArn=arn:aws:s3:::amzn-s3-demo-
bucket,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
{IntegrationResultS3DestinationArn=arn:aws:s3:::amzn-s3-demo-
bucket,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
```

Windows：

```
aws ec2 get-flow-logs-integration-template ^
    --flow-log-id fl-1234567890abcdef0 ^
    --config-delivery-s3-destination-arn arn:aws:s3:::amzn-s3-demo-bucket ^
    --integrate-
services AthenaIntegrations=[{IntegrationResultS3DestinationArn=arn:aws:s3:::amzn-
s3-demo-
```

```
bucket,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
{IntegrationResultS3DestinationArn=arn:aws:s3:::amzn-s3-demo-
bucket,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
```

輸出：

```
{
    "Result": "https://amzn-s3-demo-bucket.s3.us-east-2.amazonaws.com/
VPCFlowLogsIntegrationTemplate_fl-1234567890abcdef0_Wed%20Jul
%2021%2000%3A57%3A56%20UTC%202021.yml"
}
```

如需有關使用 CloudFormation 範本的資訊，請參閱AWS CloudFormation 使用者指南》中的使用 AWS CloudFormation 範本。

如需使用 Amazon Athena 和流程日誌的資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的使用 Amazon Athena 查詢流程日誌。 Amazon Virtual Private Cloud

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFlowLogsIntegrationTemplate。

get-groups-for-capacity-reservation

以下程式碼範例顯示如何使用 get-groups-for-capacity-reservation。

AWS CLI

列出具有容量保留的資源群組

下列get-groups-for-capacity-reservation範例列出新增指定容量保留的資源群組。

```
aws ec2 get-groups-for-capacity-reservation \
    --capacity-reservation-id cr-1234abcd56EXAMPLE
```

輸出：

```
{
    "CapacityReservationsGroup": [
        {
            "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/my-
resource-group",
            "OwnerId": "123456789012"
```

```
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的容量保留群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetGroupsForCapacityReservation。

get-host-reservation-purchase-preview

以下程式碼範例顯示如何使用 get-host-reservation-purchase-preview。

AWS CLI

取得專用主機預留的購買預覽

此範例提供您帳戶中指定專用主機之指定專用主機預留的成本預覽。

命令：

```
aws ec2 get-host-reservation-purchase-preview --offering-id hro-03f707bf363b6b324 --host-id-set h-013abcd2a00cbd123
```

輸出：

```
{
  "TotalHourlyPrice": "1.499",
  "Purchase": [
      {
          "HourlyPrice": "1.499",
          "InstanceFamily": "m4",
          "PaymentOption": "NoUpfront",
          "HostIdSet": [
              "h-013abcd2a00cbd123"
          ],
          "UpfrontPrice": "0.000",
          "Duration": 31536000
      }
  ],
  "TotalUpfrontPrice": "0.000"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetHostReservationPurchasePreview。

get-image-block-public-access-state

以下程式碼範例顯示如何使用 get-image-block-public-access-state。

AWS CLI

取得指定區域中 AMIs的封鎖公有存取狀態

下列get-image-block-public-access-state範例會在指定區域中的帳戶層級取得 AMIs的封鎖公有存取狀態。

```
aws ec2 get-image-block-public-access-state \
    --region us-east-1
```

輸出：

```
{
    "ImageBlockPublicAccessState": "block-new-sharing"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的封鎖對 AMIs公開存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetImageBlockPublicAccessState。

get-instance-types-from-instance-requirements

以下程式碼範例顯示如何使用 get-instance-types-from-instance-requirements。

AWS CLI

預覽符合指定屬性的執行個體類型

下列get-instance-types-from-instance-requirements範例會先產生可使用 --generate-cli-skeleton 參數指定之所有可能屬性的清單，並將清單儲存至 JSON 檔案。然後，JSON 檔案會用來自訂要預覽相符執行個體類型的屬性。

若要產生所有可能的屬性，並將輸出直接儲存到 JSON 檔案，請使用下列命令。

```
aws ec2 get-instance-types-from-instance-requirements \
    --region us-east-1 \
    --generate-cli-skeleton input > attributes.json
```

輸出：

```
{
    "DryRun": true,
    "ArchitectureTypes": [
        "x86_64_mac"
    ],
    "VirtualizationTypes": [
        "paravirtual"
    ],
    "InstanceRequirements": {
        "VCpuCount": {
            "Min": 0,
            "Max": 0
        },
        "MemoryMiB": {
            "Min": 0,
            "Max": 0
        },
        "CpuManufacturers": [
            "intel"
        ],
        "MemoryGiBPerVCpu": {
            "Min": 0.0,
            "Max": 0.0
        },
        "ExcludedInstanceTypes": [
            ""
        ],
        "InstanceGenerations": [
            "current"
        ],
        "SpotMaxPricePercentageOverLowestPrice": 0,
        "OnDemandMaxPricePercentageOverLowestPrice": 0,
        "BareMetal": "included",
        "BurstablePerformance": "excluded",
        "RequireHibernateSupport": true,
        "NetworkInterfaceCount": {
            "Min": 0,
```

```
            "Max": 0
        },
        "LocalStorage": "required",
        "LocalStorageTypes": [
            "hdd"
        ],
        "TotalLocalStorageGB": {
            "Min": 0.0,
            "Max": 0.0
        },
        "BaselineEbsBandwidthMbps": {
            "Min": 0,
            "Max": 0
        },
        "AcceleratorTypes": [
            "inference"
        ],
        "AcceleratorCount": {
            "Min": 0,
            "Max": 0
        },
        "AcceleratorManufacturers": [
            "xilinx"
        ],
        "AcceleratorNames": [
            "t4"
        ],
        "AcceleratorTotalMemoryMiB": {
            "Min": 0,
            "Max": 0
        }
    },
    "MaxResults": 0,
    "NextToken": ""
}
```

設定 JSON 檔案。您必須提供 `ArchitectureTypes`、`VirtualizationTypes`、`VCpuCount` 和 `MemoryMiB` 的值。您可以省略其他屬性。省略時，會使用預設值。如需每個屬性及其預設值的說明，請參閱 get-instance-types-from-instance-requirements <https://docs.aws.amazon.com/cli/latest/reference/ec2/get-instance-types-from-instance-requirements.html>。

預覽具有 中指定屬性的執行個體類型`attributes.json`。使用 `--cli-input-json` 參數指定 JSON 檔案的名稱和路徑。在下列請求中，輸出會格式化為資料表。

```
aws ec2 get-instance-types-from-instance-requirements \
    --cli-input-json file://attributes.json \
    --output table
```

attributes.json 檔案的內容：

```
{

    "ArchitectureTypes": [
        "x86_64"
    ],
    "VirtualizationTypes": [
        "hvm"
    ],
    "InstanceRequirements": {
        "VCpuCount": {
            "Min": 4,
            "Max": 6
        },
        "MemoryMiB": {
            "Min": 2048
        },
        "InstanceGenerations": [
            "current"
        ]
    }
}
```

輸出：

```
-----------------------------------------
|GetInstanceTypesFromInstanceRequirements|
+---------------------------------------+
||            InstanceTypes            ||
|+-------------------------------------+|
||            InstanceType             ||
|+-------------------------------------+|
||  c4.xlarge                          ||
||  c5.xlarge                          ||
||  c5a.xlarge                         ||
||  c5ad.xlarge                        ||
||  c5d.xlarge                         ||
```

```
||  c5n.xlarge                          ||
||  d2.xlarge                           ||
...
```

如需屬性型執行個體類型選擇的詳細資訊，請參閱《Amazon EC2 使用者指南》中的屬性型執行個體類型選擇的運作方式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInstanceTypesFromInstanceRequirements。

get-instance-uefi-data

以下程式碼範例顯示如何使用 get-instance-uefi-data。

AWS CLI

從執行個體擷取 UEFI 資料

下列get-instance-uefi-data範例會從執行個體擷取 UEFI 資料。如果輸出為空，則執行個體不包含 UEFI 資料。

```
aws ec2 get-instance-uefi-data \
    --instance-id i-0123456789example
```

輸出：

```
{
    "InstanceId": "i-0123456789example",
    "UefiData": "QU1aTlVFRkkf+uLXAAAAAHj5a7fZ9+3dBzxXb/.
    <snipped>
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAD4L/J/AODshho="
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 UEFI 安全開機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInstanceUefiData。

get-ipam-address-history

以下程式碼範例顯示如何使用 get-ipam-address-history。

AWS CLI

取得 CIDR 的歷史記錄

下列get-ipam-address-history範例會取得 CIDR 的歷史記錄。

(Linux)：

```
aws ec2 get-ipam-address-history \
   --cidr 10.0.0.0/16 \
   --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 \
   --start-time 2021-12-08T01:00:00.000Z \
   --end-time 2021-12-10T01:00:00.000Z
```

(Windows)：

```
aws ec2 get-ipam-address-history ^
   --cidr 10.0.0.0/16 ^
   --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 ^
   --start-time 2021-12-08T01:00:00.000Z ^
   --end-time 2021-12-10T01:00:00.000Z
```

輸出：

```
{
    "HistoryRecords": [
        {
            "ResourceOwnerId": "123456789012",
            "ResourceRegion": "us-west-1",
            "ResourceType": "vpc",
            "ResourceId": "vpc-06cbefa9ee907e1c0",
            "ResourceCidr": "10.0.0.0/16",
            "ResourceName": "Demo",
            "ResourceComplianceStatus": "unmanaged",
            "ResourceOverlapStatus": "overlapping",
            "VpcId": "vpc-06cbefa9ee907e1c0",
            "SampledStartTime": "2021-12-08T19:54:57.675000+00:00"
        },
        {
            "ResourceOwnerId": "123456789012",
            "ResourceRegion": "us-east-2",
            "ResourceType": "vpc",
            "ResourceId": "vpc-042702f474812c9ad",
```

```
            "ResourceCidr": "10.0.0.0/16",
            "ResourceName": "test",
            "ResourceComplianceStatus": "unmanaged",
            "ResourceOverlapStatus": "overlapping",
            "VpcId": "vpc-042702f474812c9ad",
            "SampledStartTime": "2021-12-08T19:54:59.019000+00:00"
        },
        {
            "ResourceOwnerId": "123456789012",
            "ResourceRegion": "us-east-2",
            "ResourceType": "vpc",
            "ResourceId": "vpc-042b8a44f64267d67",
            "ResourceCidr": "10.0.0.0/16",
            "ResourceName": "tester",
            "ResourceComplianceStatus": "unmanaged",
            "ResourceOverlapStatus": "overlapping",
            "VpcId": "vpc-042b8a44f64267d67",
            "SampledStartTime": "2021-12-08T19:54:59.019000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的檢視 IP 地址的歷史記錄。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIpamAddressHistory。

get-ipam-discovered-accounts

以下程式碼範例顯示如何使用 get-ipam-discovered-accounts。

AWS CLI

檢視 IPAM 發現的帳戶

在此案例中，您是 IPAM 委派的管理員，想要檢視擁有 IPAM 正在探索之資源 AWS 的帳戶。

--discovery-region 是您要檢視受監控帳戶狀態的 IPAM 作業系統區域。例如，如果您有三個 IPAM 操作區域，您可能想要發出此請求三次，以檢視每個特定區域中探索的特定時間戳記。

下列get-ipam-discovered-accounts範例列出擁有 IPAM 正在探索之資源 AWS 的帳戶。

```
aws ec2 get-ipam-discovered-accounts \
   --ipam-resource-discovery-id ipam-res-disco-0365d2977fc1672fe \
```

```
    --discovery-region us-east-1
```

輸出：

```
{
    "IpamDiscoveredAccounts": [
        {
            "AccountId": "149977607591",
            "DiscoveryRegion": "us-east-1",
            "LastAttemptedDiscoveryTime": "2024-02-09T19:04:31.379000+00:00",
            "LastSuccessfulDiscoveryTime": "2024-02-09T19:04:31.379000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的將 IPAM 與組織外部的帳戶整合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIpamDiscoveredAccounts。

get-ipam-discovered-public-addresses

以下程式碼範例顯示如何使用 get-ipam-discovered-public-addresses。

AWS CLI

檢視探索到的公有 IP 地址

在此範例中，您是 IPAM 委派管理員，而且您想要檢視 IPAM 所探索資源的 IP 地址。您可以使用 describe-ipam-resource-discoveries 取得資源探索 ID。

下列get-ipam-discovered-public-addresses範例顯示資源探索的探索公有 IP 地址。

```
aws ec2 get-ipam-discovered-public-addresses \
    --ipam-resource-discovery-id ipam-res-disco-0f4ef577a9f37a162 \
    --address-region us-east-1 \
    --region us-east-1
```

輸出：

```
{
    "IpamDiscoveredPublicAddresses": [
        {
           "IpamResourceDiscoveryId": "ipam-res-disco-0f4ef577a9f37a162",
```

```
            "AddressRegion": "us-east-1",
            "Address": "54.208.155.7",
            "AddressOwnerId": "320805250157",
            "AssociationStatus": "associated",
            "AddressType": "ec2-public-ip",
            "VpcId": "vpc-073b294916198ce49",
            "SubnetId": "subnet-0b6c8a8839e9a4f15",
            "NetworkInterfaceId": "eni-081c446b5284a5e06",
            "NetworkInterfaceDescription": "",
            "InstanceId": "i-07459a6fca5b35823",
            "Tags": {},
            "NetworkBorderGroup": "us-east-1c",
            "SecurityGroups": [
                {
                    "GroupName": "launch-wizard-2",
                    "GroupId": "sg-0a489dd6a65c244ce"
                }
            ],
            "SampleTime": "2024-04-05T15:13:59.228000+00:00"
        },
        {
            "IpamResourceDiscoveryId": "ipam-res-disco-0f4ef577a9f37a162",
            "AddressRegion": "us-east-1",
            "Address": "44.201.251.218",
            "AddressOwnerId": "470889052923",
            "AssociationStatus": "associated",
            "AddressType": "ec2-public-ip",
            "VpcId": "vpc-6c31a611",
            "SubnetId": "subnet-062f47608b99834b1",
            "NetworkInterfaceId": "eni-024845359c2c3ae9b",
            "NetworkInterfaceDescription": "",
            "InstanceId": "i-04ef786d9c4e03f41",
            "Tags": {},
            "NetworkBorderGroup": "us-east-1a",
            "SecurityGroups": [
                {
                    "GroupName": "launch-wizard-32",
                    "GroupId": "sg-0ed1a426e96a68374"
                }
            ],
            "SampleTime": "2024-04-05T15:13:59.145000+00:00"
        }
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的檢視公有 IP 洞察。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIpamDiscoveredPublicAddresses。

get-ipam-discovered-resource-cidrs

以下程式碼範例顯示如何使用 get-ipam-discovered-resource-cidrs。

AWS CLI

檢視 IPAM 發現的 IP 地址 CIDRs

在此範例中，您是 IPAM 委派的管理員，想要檢視 IPAM 正在探索之資源的 IP 地址 CIDRs 的相關詳細資訊。

若要完成此請求：

您選擇的資源探索必須與 IPAM 相關聯。 --resource-region是建立資源 AWS 的區域。

下列get-ipam-discovered-resource-cidrs範例列出 IPAM 正在探索之資源的 IP 地址。

```
aws ec2 get-ipam-discovered-resource-cidrs \
   --ipam-resource-discovery-id ipam-res-disco-0365d2977fc1672fe \
   --resource-region us-east-1
```

輸出：

```
{
    {
        "IpamDiscoveredResourceCidrs": [
        {
            "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
            "ResourceRegion": "us-east-1",
            "ResourceId": "vpc-0c974c95ca7ceef4a",
            "ResourceOwnerId": "149977607591",
            "ResourceCidr": "172.31.0.0/16",
            "ResourceType": "vpc",
            "ResourceTags": [],
            "IpUsage": 0.375,
            "VpcId": "vpc-0c974c95ca7ceef4a",
            "SampleTime": "2024-02-09T19:15:16.529000+00:00"
        },
        {
```

```
            "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
            "ResourceRegion": "us-east-1",
            "ResourceId": "subnet-07fe028119082a8c1",
            "ResourceOwnerId": "149977607591",
            "ResourceCidr": "172.31.0.0/20",
            "ResourceType": "subnet",
            "ResourceTags": [],
            "IpUsage": 0.0012,
            "VpcId": "vpc-0c974c95ca7ceef4a",
            "SampleTime": "2024-02-09T19:15:16.529000+00:00"
        },
        {
            "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
            "ResourceRegion": "us-east-1",
            "ResourceId": "subnet-0a96893763984cc4e",
            "ResourceOwnerId": "149977607591",
            "ResourceCidr": "172.31.64.0/20",
            "ResourceType": "subnet",
            "ResourceTags": [],
            "IpUsage": 0.0012,
            "VpcId": "vpc-0c974c95ca7ceef4a",
            "SampleTime": "2024-02-09T19:15:16.529000+00:00"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的依資源監控 CIDR 用量。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIpamDiscoveredResourceCidrs。

get-ipam-pool-allocations

以下程式碼範例顯示如何使用 get-ipam-pool-allocations。

AWS CLI

從 IPAM 集區取得配置CIDRs

下列get-ipam-pool-allocations範例會從 IPAM 集區取得配置的 CIDRs。

(Linux)：

```
aws ec2 get-ipam-pool-allocations \
```

```
    --ipam-pool-id ipam-pool-0533048da7d823723 \
    --filters Name=ipam-pool-allocation-id,Values=ipam-pool-
alloc-0e6186d73999e47389266a5d6991e6220
```

(Windows)：

```
aws ec2 get-ipam-pool-allocations ^
    --ipam-pool-id ipam-pool-0533048da7d823723 ^
    --filters Name=ipam-pool-allocation-id,Values=ipam-pool-
alloc-0e6186d73999e47389266a5d6991e6220
```

輸出：

```
{
    "IpamPoolAllocations": [
        {
            "Cidr": "10.0.0.0/16",
            "IpamPoolAllocationId": "ipam-pool-
alloc-0e6186d73999e47389266a5d6991e6220",
            "ResourceType": "custom",
            "ResourceOwner": "123456789012"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIpamPoolAllocations。

get-ipam-pool-cidrs

以下程式碼範例顯示如何使用 get-ipam-pool-cidrs。

AWS CLI

取得佈建至 IPAM 集區的 CIDRs

下列get-ipam-pool-cidrs範例會取得佈建至 IPAM 集區的 CIDRs。

(Linux)：

```
aws ec2 get-ipam-pool-cidrs \
    --ipam-pool-id ipam-pool-0533048da7d823723 \
```

```
    --filters 'Name=cidr,Values=10.*'
```

(Windows)：

```
aws ec2 get-ipam-pool-cidrs ^
    --ipam-pool-id ipam-pool-0533048da7d823723 ^
    --filters Name=cidr,Values=10.*
```

輸出：

```
{
    "IpamPoolCidr": {
        "Cidr": "10.0.0.0/24",
        "State": "provisioned"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIpamPoolCidrs。

get-ipam-resource-cidrs

以下程式碼範例顯示如何使用 get-ipam-resource-cidrs。

AWS CLI

取得配置給資源CIDRs

下列get-ipam-resource-cidrs範例會取得配置給資源CIDRs。

(Linux)：

```
aws ec2 get-ipam-resource-cidrs \
    --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 \
    --filters Name=management-state,Values=unmanaged
```

(Windows)：

```
aws ec2 get-ipam-resource-cidrs ^
    --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 ^
    --filters Name=management-state,Values=unmanaged
```

輸出：

```
{
    "IpamResourceCidrs": [
        {
            "IpamId": "ipam-08440e7a3acde3908",
            "IpamScopeId": "ipam-scope-02fc38cd4c48e7d38",
            "ResourceRegion": "us-east-2",
            "ResourceOwnerId": "123456789012",
            "ResourceId": "vpc-621b8709",
            "ResourceName": "Default AWS VPC",
            "ResourceCidr": "172.33.0.0/16",
            "ResourceType": "vpc",
            "ResourceTags": [
                {
                    "Key": "Environment",
                    "Value": "Test"
                },
                {
                    "Key": "Name",
                    "Value": "Default AWS VPC"
                }
            ],
            "IpUsage": 0.0039,
            "ComplianceStatus": "unmanaged",
            "ManagementState": "unmanaged",
            "OverlapStatus": "nonoverlapping",
            "VpcId": "vpc-621b8709"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的依資源監控 CIDR 用量。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIpamResourceCidrs。

get-launch-template-data

以下程式碼範例顯示如何使用 get-launch-template-data。

AWS CLI

取得啟動範本的執行個體資料

此範例會取得指定執行個體的資料，並使用 --query選項傳回 中的內容LaunchTemplateData。您能以輸出做為基礎，用以建立新啟動範本或啟動範本版本。

命令：

```
aws ec2 get-launch-template-data --instance-id i-0123d646e8048babc --query
 'LaunchTemplateData'
```

輸出：

```
{
      "Monitoring": {},
      "ImageId": "ami-8c1be5f6",
      "BlockDeviceMappings": [
          {
              "DeviceName": "/dev/xvda",
              "Ebs": {
                  "DeleteOnTermination": true
              }
          }
      ],
      "EbsOptimized": false,
      "Placement": {
          "Tenancy": "default",
          "GroupName": "",
          "AvailabilityZone": "us-east-1a"
      },
      "InstanceType": "t2.micro",
      "NetworkInterfaces": [
          {
              "Description": "",
              "NetworkInterfaceId": "eni-35306abc",
              "PrivateIpAddresses": [
                  {
                      "Primary": true,
                      "PrivateIpAddress": "10.0.0.72"
                  }
              ],
              "SubnetId": "subnet-7b16de0c",
              "Groups": [
                  "sg-7c227019"
              ],
              "Ipv6Addresses": [
```

```
                {
                    "Ipv6Address": "2001:db8:1234:1a00::123"
                }
            ],
            "PrivateIpAddress": "10.0.0.72"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLaunchTemplateData。

get-managed-prefix-list-associations

以下程式碼範例顯示如何使用 get-managed-prefix-list-associations。

AWS CLI

取得字首清單關聯

下列get-managed-prefix-list-associations範例會取得與指定字首清單相關聯的資源。

```
aws ec2 get-managed-prefix-list-associations \
    --prefix-list-id pl-0123456abcabcabc1
```

輸出：

```
{
    "PrefixListAssociations": [
        {
            "ResourceId": "sg-0abc123456abc12345",
            "ResourceOwner": "123456789012"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的受管字首清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetManagedPrefixListAssociations。

get-managed-prefix-list-entries

以下程式碼範例顯示如何使用 get-managed-prefix-list-entries。

AWS CLI

取得字首清單的項目

以下內容get-managed-prefix-list-entries會取得指定字首清單的項目。

```
aws ec2 get-managed-prefix-list-entries \
    --prefix-list-id pl-0123456abcabcabc1
```

輸出：

```
{
    "Entries": [
        {
            "Cidr": "10.0.0.0/16",
            "Description": "vpc-a"
        },
        {
            "Cidr": "10.2.0.0/16",
            "Description": "vpc-b"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的受管字首清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetManagedPrefixListEntries。

get-network-insights-access-scope-analysis-findings

以下程式碼範例顯示如何使用 get-network-insights-access-scope-analysis-findings。

AWS CLI

取得 Network Insights 存取範圍分析的問題清單

下列get-network-insights-access-scope-analysis-findings範例會取得您 AWS 帳戶中選取的範圍分析問題清單。

```
aws ec2 get-network-insights-access-scope-analysis-findings \
    --region us-east-1 \
    --network-insights-access-scope-analysis-id nis \
```

```
    --nis-123456789111
```

輸出：

```
{
    "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789222",
    "AnalysisFindings": [
        {
            "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789222",
            "NetworkInsightsAccessScopeId": "nis-123456789111",
            "FindingComponents": [
                {
                    "SequenceNumber": 1,
                    "Component": {
                        "Id": "eni-02e3d42d5cceca67d",
                        "Arn": "arn:aws:ec2:us-east-1:936459623503:network-
interface/eni-02e3d32d9cceca17d"
                    },
                    "OutboundHeader": {
                        "DestinationAddresses": [
                            "0.0.0.0/5",
                            "11.0.0.0/8",
                            "12.0.0.0/6",
                            "128.0.0.0/3",
                            "16.0.0.0/4",
                            "160.0.0.0/5",
                            "168.0.0.0/6",
                            "172.0.0.0/12"
                            "8.0.0.0/7"
                        ],
                        "DestinationPortRanges": [
                            {
                                "From": 0,
                                "To": 65535
                            }
                        ],
                        "Protocol": "6",
                        "SourceAddresses": [
                            "10.0.2.253/32"
                        ],
                        "SourcePortRanges": [
                            {
                                "From": 0,
```

```
                                "To": 65535
                            }
                        ]
                    }, [etc]
                ]
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Network Access Analyzer 指南》中的使用 AWS CLI 開始使用 Network Access Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetNetworkInsightsAccessScopeAnalysisFindings。

get-network-insights-access-scope-content

以下程式碼範例顯示如何使用 get-network-insights-access-scope-content。

AWS CLI

取得 Network Insights 存取範圍內容

下列get-network-insights-access-scope-content範例會取得您 AWS 帳戶中所選範圍分析 ID 的內容。

```
aws ec2 get-network-insights-access-scope-content \
    --region us-east-1 \
    --network-insights-access-scope-id nis-123456789222
```

輸出：

```
{
    "NetworkInsightsAccessScopeContent": {
        "NetworkInsightsAccessScopeId": "nis-123456789222",
        "MatchPaths": [
            {
                "Source": {
                    "ResourceStatement": {
                        "ResourceTypes": [
                            "AWS::EC2::NetworkInterface"
```

```
                        ]
                    }
                },
                "Destination": {
                    "ResourceStatement": {
                        "ResourceTypes": [
                            "AWS::EC2::InternetGateway"
                        ]
                    }
                }
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Network Access Analyzer 指南》中的使用 AWS CLI 開始使用 Network Access Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetNetworkInsightsAccessScopeContent。

get-password-data

以下程式碼範例顯示如何使用 get-password-data。

AWS CLI

若要取得加密的密碼

此範例會取得加密的密碼。

命令：

```
aws ec2 get-password-data --instance-id i-1234567890abcdef0
```

輸出：

```
{
    "InstanceId": "i-1234567890abcdef0",
    "Timestamp": "2013-08-07T22:18:38.000Z",
    "PasswordData": "gSlJFq+VpcZXqy+iktxMF6NyxQ4qCrT4+gaOuNOenX1MmgXPTj7XEXAMPLE
UQ+YeFfb+L1U4C4AKv652Ux1iRB3CPTYP7WmU3TUnhsuBd+p6LVk7T2lKUml6OXbk6WPW1VYYm/TRPB1
```

```
e1DQ7PY4an/DgZT4mwcpRFigzhniQgDDeO1InvSDcwoUTwNs0Y1S8ouri2W4n5GNlriM3Q0AnNVelVz/
53TkDtxbNoU606M1gK9zUWSxqEgwvbV2j8c5rP0WCuaMWSFl4ziDu4bd7q+4RSyi8NUsVWnKZ4aEZffu
DPGzKrF5yLlf3etP2L4ZR6CvG7K1hx7VKOQVN32Dajw=="
}
```

若要取得解密的密碼

此範例會取得解密的密碼。

命令：

```
aws ec2 get-password-data --instance-id  i-1234567890abcdef0 --priv-launch-key C:
\Keys\MyKeyPair.pem
```

輸出：

```
{
    "InstanceId": "i-1234567890abcdef0",
    "Timestamp": "2013-08-30T23:18:05.000Z",
    "PasswordData": "&ViJ652e*u"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPasswordData。

get-reserved-instances-exchange-quote

以下程式碼範例顯示如何使用 get-reserved-instances-exchange-quote。

AWS CLI

取得交換可轉換預留執行個體的引號

此範例會取得指定可轉換預留執行個體的交換資訊。

命令：

```
aws ec2 get-reserved-instances-exchange-quote --reserved-
instance-ids 7b8750c3-397e-4da4-bbcb-a45ebexample --target-
configurations OfferingId=6fea5434-b379-434c-b07b-a7abexample
```

輸出：

```
{
  "CurrencyCode": "USD",
  "ReservedInstanceValueSet": [
      {
          "ReservedInstanceId": "7b8750c3-397e-4da4-bbcb-a45ebexample",
          "ReservationValue": {
              "RemainingUpfrontValue": "0.000000",
              "HourlyPrice": "0.027800",
              "RemainingTotalValue": "730.556200"
          }
      }
  ],
  "PaymentDue": "424.983828",
  "TargetConfigurationValueSet": [
      {
          "TargetConfiguration": {
              "InstanceCount": 5,
              "OfferingId": "6fea5434-b379-434c-b07b-a7abexample"
          },
          "ReservationValue": {
              "RemainingUpfrontValue": "424.983828",
              "HourlyPrice": "0.016000",
              "RemainingTotalValue": "845.447828"
          }
      }
  ],
  "IsValidExchange": true,
  "OutputReservedInstancesWillExpireAt": "2020-10-01T13:03:39Z",
  "ReservedInstanceValueRollup": {
      "RemainingUpfrontValue": "0.000000",
      "HourlyPrice": "0.027800",
      "RemainingTotalValue": "730.556200"
  },
  "TargetConfigurationValueRollup": {
      "RemainingUpfrontValue": "424.983828",
      "HourlyPrice": "0.016000",
      "RemainingTotalValue": "845.447828"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetReservedInstancesExchangeQuote。

get-security-groups-for-vpc

以下程式碼範例顯示如何使用 get-security-groups-for-vpc。

AWS CLI

檢視可與指定 VPC 中的網路介面相關聯的安全群組。

下列get-security-groups-for-vpc範例顯示可與 VPC 中的網路介面相關聯的安全群組。

```
aws ec2 get-security-groups-for-vpc \
    --vpc-id vpc-6c31a611 \
    --region us-east-1
```

輸出：

```
{
    "SecurityGroupForVpcs": [
        {
            "Description": "launch-wizard-36 created 2022-08-29T15:59:35.338Z",
            "GroupName": "launch-wizard-36",
            "OwnerId": "470889052923",
            "GroupId": "sg-007e0c3027ee885f5",
            "Tags": [],
            "PrimaryVpcId": "vpc-6c31a611"
        },
        {
            "Description": "launch-wizard-18 created 2024-01-19T20:22:27.527Z",
            "GroupName": "launch-wizard-18",
            "OwnerId": "470889052923",
            "GroupId": "sg-0147193bef51c9eef",
            "Tags": [],
            "PrimaryVpcId": "vpc-6c31a611"
        }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSecurityGroupsForVpc。

get-serial-console-access-status

以下程式碼範例顯示如何使用 get-serial-console-access-status。

AWS CLI

檢視 帳戶存取序列主控台的狀態

下列get-serial-console-access-status範例會判斷是否為您的 帳戶啟用序列主控台存取。

```
aws ec2 get-serial-console-access-status
```

輸出：

```
{
    "SerialConsoleAccessEnabled": true
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 EC2 序列主控台。 Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSerialConsoleAccessStatus。

get-snapshot-block-public-access-state

以下程式碼範例顯示如何使用 get-snapshot-block-public-access-state。

AWS CLI

取得快照封鎖公開存取的目前狀態

下列get-snapshot-block-public-access-state範例取得快照封鎖公開存取的目前狀態。

```
aws ec2 get-snapshot-block-public-access-state
```

輸出：

```
{
    "State": "block-all-sharing"
}
```

如需詳細資訊，請參閱《Amazon EBS 使用者指南》中的封鎖快照的公開存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSnapshotBlockPublicAccessState。

get-spot-placement-scores

以下程式碼範例顯示如何使用 get-spot-placement-scores。

AWS CLI

計算指定需求的 Spot 配置分數

下列get-spot-placement-scores範例會先產生可使用 參數為 Spot 配置分數組態指定的所有可能--generate-cli-skeleton參數清單，並將清單儲存至 JSON 檔案。然後，JSON 檔案用於設定用於計算 Spot 置放分數的要求。

若要產生可指定給 Spot 配置分數組態的所有可能參數，並將輸出直接儲存到 JSON 檔案。

```
aws ec2 get-spot-placement-scores \
    --region us-east-1 \
    --generate-cli-skeleton input > attributes.json
```

輸出：

```
{
    "InstanceTypes": [
        ""
    ],
    "TargetCapacity": 0,
    "TargetCapacityUnitType": "vcpu",
    "SingleAvailabilityZone": true,
    "RegionNames": [
        ""
    ],
    "InstanceRequirementsWithMetadata": {
        "ArchitectureTypes": [
            "x86_64_mac"
        ],
        "VirtualizationTypes": [
            "hvm"
        ],
        "InstanceRequirements": {
            "VCpuCount": {
                "Min": 0,
                "Max": 0
            },
            "MemoryMiB": {
```

```
                "Min": 0,
                "Max": 0
            },
            "CpuManufacturers": [
                "amd"
            ],
            "MemoryGiBPerVCpu": {
                "Min": 0.0,
                "Max": 0.0
            },
            "ExcludedInstanceTypes": [
                ""
            ],
            "InstanceGenerations": [
                "previous"
            ],
            "SpotMaxPricePercentageOverLowestPrice": 0,
            "OnDemandMaxPricePercentageOverLowestPrice": 0,
            "BareMetal": "excluded",
            "BurstablePerformance": "excluded",
            "RequireHibernateSupport": true,
            "NetworkInterfaceCount": {
                "Min": 0,
                "Max": 0
            },
            "LocalStorage": "included",
            "LocalStorageTypes": [
                "hdd"
            ],
            "TotalLocalStorageGB": {
                "Min": 0.0,
                "Max": 0.0
            },
            "BaselineEbsBandwidthMbps": {
                "Min": 0,
                "Max": 0
            },
            "AcceleratorTypes": [
                "fpga"
            ],
            "AcceleratorCount": {
                "Min": 0,
                "Max": 0
            },
```

```
            "AcceleratorManufacturers": [
                "amd"
            ],
            "AcceleratorNames": [
                "vu9p"
            ],
            "AcceleratorTotalMemoryMiB": {
                "Min": 0,
                "Max": 0
            }
        }
    },
    "DryRun": true,
    "MaxResults": 0,
    "NextToken": ""
}
```

設定 JSON 檔案。您必須提供 TargetCapacity 的值。如需每個參數及其預設值的說明，請參閱計算 Spot 置放分數 (AWS CLI) <https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/spot-placement-score.html#calculate-sps-cli>。

計算 中指定需求的 Spot 配置分數attributes.json。使用 --cli-input-json 參數指定 JSON 檔案的名稱和路徑。

```
aws ec2 get-spot-placement-scores \
    --region us-east-1 \
    --cli-input-json file://attributes.json
```

如果 SingleAvailabilityZone 設為 false或省略 ，則輸出 （ 如果省略，則預設為 false)。系統會傳回區域的評分清單。

```
"Recommendation": [
    {
        "Region": "us-east-1",
        "Score": 7
    },
    {
        "Region": "us-west-1",
        "Score": 5
    },
   ...
```

如果 SingleAvailabilityZone 設定為 則輸出true。傳回 SingleAvailability Zones 的評分清單。

```
"Recommendation": [
    {
        "Region": "us-east-1",
        "AvailabilityZoneId": "use1-az1"
        "Score": 8
    },
    {
        "Region": "us-east-1",
        "AvailabilityZoneId": "usw2-az3"
        "Score": 6
    },
   ...
```

如需計算 Spot 置放分數的詳細資訊，以及範例組態，請參閱《Amazon EC2 使用者指南》中的計算 Spot 置放分數。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSpotPlacementScores。

get-subnet-cidr-reservations

以下程式碼範例顯示如何使用 get-subnet-cidr-reservations。

AWS CLI

取得子網路 CIDR 保留的相關資訊

下列get-subnet-cidr-reservations範例顯示指定子網路 CIDR 保留的相關資訊。

```
aws ec2 get-subnet-cidr-reservations \
    --subnet-id subnet-03c51e2e6cEXAMPLE
```

輸出：

```
{
    "SubnetIpv4CidrReservations": [
        {
            "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
            "SubnetId": "subnet-03c51e2e6cEXAMPLE",
```

```
            "Cidr": "10.1.0.16/28",
            "ReservationType": "prefix",
            "OwnerId": "123456789012"
        }
    ],
    "SubnetIpv6CidrReservations": []
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 子網 CIDR 保留。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSubnetCidrReservations。

get-transit-gateway-attachment-propagations

以下程式碼範例顯示如何使用 get-transit-gateway-attachment-propagations。

AWS CLI

列出指定資源連接傳播路由的路由表

下列get-transit-gateway-attachment-propagations範例列出指定資源連接傳播路由的路由表。

```
aws ec2 get-transit-gateway-attachment-propagations \
    --transit-gateway-attachment-id tgw-attach-09fbd47ddfEXAMPLE
```

輸出：

```
{
    "TransitGatewayAttachmentPropagations": [
        {
            "TransitGatewayRouteTableId": "tgw-rtb-0882c61b97EXAMPLE",
            "State": "enabled"
        }
    ]
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的 Transit Gateway 路由表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTransitGatewayAttachmentPropagations。

get-transit-gateway-multicast-domain-associations

以下程式碼範例顯示如何使用 get-transit-gateway-multicast-domain-associations。

AWS CLI

檢視傳輸閘道多點傳送網域關聯的相關資訊

下列get-transit-gateway-multicast-domain-associations範例會傳回指定多點傳送網域的關聯。

```
aws ec2 get-transit-gateway-multicast-domain-associations \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

輸出：

```
{
    "MulticastDomainAssociations": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-028c1dd0f8EXAMPLE",
            "ResourceId": "vpc-01128d2c24EXAMPLE",
            "ResourceType": "vpc",
            "Subnet": {
                "SubnetId": "subnet-000de86e3bEXAMPLE",
                "State": "associated"
            }
        },
        {
            "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",
            "ResourceId": "vpc-7EXAMPLE",
            "ResourceType": "vpc",
            "Subnet": {
                "SubnetId": "subnet-4EXAMPLE",
                "State": "associated"
            }
        },
        {
            "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",
            "ResourceId": "vpc-7EXAMPLE",
            "ResourceType": "vpc",
            "Subnet": {
                "SubnetId": "subnet-5EXAMPLE",
                "State": "associated"
```

```
            }
        },
        {
            "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",
            "ResourceId": "vpc-7EXAMPLE",
            "ResourceType": "vpc",
            "Subnet": {
                "SubnetId": "subnet-aEXAMPLE",
                "State": "associated"
            }
        },
        {
            "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",
            "ResourceId": "vpc-7EXAMPLE",
            "ResourceType": "vpc",
            "Subnet": {
                "SubnetId": "subnet-fEXAMPLE",
                "State": "associated"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的多點傳送網域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTransitGatewayMulticastDomainAssociations。

get-transit-gateway-prefix-list-references

以下程式碼範例顯示如何使用 get-transit-gateway-prefix-list-references。

AWS CLI

在傳輸閘道路由表中取得字首清單參考

下列get-transit-gateway-prefix-list-references範例會取得指定傳輸閘道路由表的字首清單參考，並依特定字首清單的 ID 進行篩選。

```
aws ec2 get-transit-gateway-prefix-list-references \
    --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
    --filters Name=prefix-list-id,Values=pl-11111122222222333
```

輸出：

```
{
    "TransitGatewayPrefixListReferences": [
        {
            "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
            "PrefixListId": "pl-11111122222222333",
            "PrefixListOwnerId": "123456789012",
            "State": "available",
            "Blackhole": false,
            "TransitGatewayAttachment": {
                "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",
                "ResourceType": "vpc",
                "ResourceId": "vpc-112233445566aabbc"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的字首清單參考。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTransitGatewayPrefixListReferences。

get-transit-gateway-route-table-associations

以下程式碼範例顯示如何使用 get-transit-gateway-route-table-associations。

AWS CLI

取得指定傳輸閘道路由表之關聯的相關資訊

下列get-transit-gateway-route-table-associations範例顯示指定傳輸閘道路由表之關聯的相關資訊。

```
aws ec2 get-transit-gateway-route-table-associations \
    --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE
```

輸出：

```
{
```

```
    "Associations": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
            "ResourceId": "vpc-4d7de228",
            "ResourceType": "vpc",
            "State": "associating"
        }
    ]
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的 Transit Gateway 路由表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTransitGatewayRouteTableAssociations。

get-transit-gateway-route-table-propagations

以下程式碼範例顯示如何使用 get-transit-gateway-route-table-propagations。

AWS CLI

顯示指定傳輸閘道路由表之路由表傳播的相關資訊

下列get-transit-gateway-route-table-propagations範例會傳回指定路由表的路由表傳播。

```
aws ec2 get-transit-gateway-route-table-propagations \
    --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE
```

輸出：

```
{
    "TransitGatewayRouteTablePropagations": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-01f8100bc7EXAMPLE",
            "ResourceId": "vpc-3EXAMPLE",
            "ResourceType": "vpc",
            "State": "enabled"
        },
        {
            "TransitGatewayAttachmentId": "tgw-attach-08e0bc912cEXAMPLE",
            "ResourceId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
```

```
            "ResourceType": "direct-connect-gateway",
            "State": "enabled"
        },
        {
            "TransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
            "ResourceId": "8384da05-13ce-4a91-aada-5a1baEXAMPLE",
            "ResourceType": "direct-connect-gateway",
            "State": "enabled"
        }
    ]
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的 Transit Gateway 路由表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTransitGatewayRouteTablePropagations。

`get-verified-access-endpoint-policy`

以下程式碼範例顯示如何使用 `get-verified-access-endpoint-policy`。

AWS CLI

取得端點的 Verified Access 政策

下列`get-verified-access-endpoint-policy`範例會取得指定端點的 Verified Access 政策。

```
aws ec2 get-verified-access-endpoint-policy \
    --verified-access-endpoint-id vae-066fac616d4d546f2
```

輸出：

```
{
    "PolicyEnabled": true,
    "PolicyDocument": "permit(principal,action,resource)\nwhen
 {\n    context.identity.groups.contains(\"finance\") &&\n
 context.identity.email_verified == true\n};"
}
```

如需詳細資訊，請參閱《Verified Access 使用者指南》中的 Verified Access 政策。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetVerifiedAccessEndpointPolicy。

get-verified-access-group-policy

以下程式碼範例顯示如何使用 get-verified-access-group-policy。

AWS CLI

取得群組的 Verified Access 政策

下列get-verified-access-group-policy範例取得指定群組的 Verified Access 政策。

```
aws ec2 get-verified-access-group-policy \
    --verified-access-group-id vagr-0dbe967baf14b7235
```

輸出：

```
{
    "PolicyEnabled": true,
    "PolicyDocument": "permit(principal,action,resource)\nwhen
 {\n    context.identity.groups.contains(\"finance\") &&\n
 context.identity.email_verified == true\n};"
}
```

如需詳細資訊，請參閱《Verified Access 使用者指南》中的 Verified Access 群組。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetVerifiedAccessGroupPolicy。

get-vpn-connection-device-sample-configuration

以下程式碼範例顯示如何使用 get-vpn-connection-device-sample-configuration。

AWS CLI

下載範例組態檔案

下列get-vpn-connection-device-sample-configuration範例會下載指定的範例組態檔案。若要列出具有範例組態檔案的閘道裝置，請呼叫 get-vpn-connection-device-types命令。

```
aws ec2 get-vpn-connection-device-sample-configuration \
    --vpn-connection-id vpn-123456789abc01234 \
    --vpn-connection-device-type-id 5fb390ba
```

輸出：

```
{
    "VpnConnectionDeviceSampleConfiguration": "contents-of-the-sample-configuration-
file"
}
```

如需詳細資訊，請參閱AWS 《 Site-to-Site VPN 使用者指南》中的下載組態檔案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetVpnConnectionDeviceSampleConfiguration。

get-vpn-connection-device-types

以下程式碼範例顯示如何使用 get-vpn-connection-device-types。

AWS CLI

使用範例組態檔案列出閘道裝置

下列get-vpn-connection-device-types範例列出來自具有範例組態檔案之 Palo Alto Networks 的閘道裝置。

```
aws ec2 get-vpn-connection-device-types \
    --query "VpnConnectionDeviceTypes[?Vendor==`Palo Alto Networks`]"
```

輸出：

```
[
    {
        "VpnConnectionDeviceTypeId": "754a6372",
        "Vendor": "Palo Alto Networks",
        "Platform": "PA Series",
        "Software": "PANOS 4.1.2+"
    },
    {
        "VpnConnectionDeviceTypeId": "9612cbed",
        "Vendor": "Palo Alto Networks",
        "Platform": "PA Series",
        "Software": "PANOS 4.1.2+ (GUI)"
    },
    {
```

```
        "VpnConnectionDeviceTypeId": "5fb390ba",
        "Vendor": "Palo Alto Networks",
        "Platform": "PA Series",
        "Software": "PANOS 7.0+"
    }
]
```

如需詳細資訊，請參閱AWS 《 Site-to-Site VPN 使用者指南》中的下載組態檔案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetVpnConnectionDeviceTypes。

import-client-vpn-client-certificate-revocation-list

以下程式碼範例顯示如何使用 import-client-vpn-client-certificate-revocation-list。

AWS CLI

匯入用戶端憑證撤銷清單

下列import-client-vpn-client-certificate-revocation-list範例透過指定檔案在本機電腦上的位置，將用戶端憑證撤銷清單匯入 Client VPN 端點。

```
aws ec2 import-client-vpn-client-certificate-revocation-list \
    --certificate-revocation-list file:///path/to/crl.pem \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

輸出：

```
{
    "Return": true
}
```

如需詳細資訊，請參閱 Client VPN 管理員指南中的用戶端憑證撤銷清單。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ImportClientVpnClientCertificateRevocationList。

import-image

以下程式碼範例顯示如何使用 import-image。

AWS CLI

將 VM 映像檔案匯入為 AMI

下列import-image範例會匯入指定的 OVA。

```
aws ec2 import-image \
  --disk-containers Format=ova,UserBucket="{S3Bucket=my-import-bucket,S3Key=vms/my-server-vm.ova}"
```

輸出：

```
{
    "ImportTaskId": "import-ami-1234567890abcdef0",
    "Progress": "2",
    "SnapshotDetails": [
        {
            "DiskImageSize": 0.0,
            "Format": "ova",
            "UserBucket": {
                "S3Bucket": "my-import-bucket",
                "S3Key": "vms/my-server-vm.ova"
            }
        }
    ],
    "Status": "active",
    "StatusMessage": "pending"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ImportImage。

import-key-pair

以下程式碼範例顯示如何使用 import-key-pair。

AWS CLI

匯入公有金鑰

首先，使用您選擇的工具產生金鑰對。例如，使用此 ssh-keygen 命令：

命令：

```
ssh-keygen -t rsa -C "my-key" -f ~/.ssh/my-key
```

輸出：

```
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ec2-user/.ssh/my-key.
Your public key has been saved in /home/ec2-user/.ssh/my-key.pub.
...
```

此範例命令會匯入指定的公有金鑰。

命令：

```
aws ec2 import-key-pair --key-name "my-key" --public-key-material fileb://~/.ssh/my-key.pub
```

輸出：

```
{
  "KeyName": "my-key",
  "KeyFingerprint": "1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ImportKeyPair。

import-snapshot

以下程式碼範例顯示如何使用 import-snapshot。

AWS CLI

匯入快照

下列import-snapshot範例會將指定的磁碟匯入為快照。

```
aws ec2 import-snapshot \
    --description "My server VMDK" \
```

```
    --disk-container Format=VMDK,UserBucket={'S3Bucket=my-import-bucket,S3Key=vms/
my-server-vm.vmdk'}
```

輸出：

```
{
    "Description": "My server VMDK",
    "ImportTaskId": "import-snap-1234567890abcdef0",
    "SnapshotTaskDetail": {
        "Description": "My server VMDK",
        "DiskImageSize": "0.0",
        "Format": "VMDK",
        "Progress": "3",
        "Status": "active",
        "StatusMessage": "pending"
        "UserBucket": {
            "S3Bucket": "my-import-bucket",
            "S3Key": "vms/my-server-vm.vmdk"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ImportSnapshot。

list-images-in-recycle-bin

以下程式碼範例顯示如何使用 `list-images-in-recycle-bin`。

AWS CLI

若要列出資源回收筒中的映像

下列`list-images-in-recycle-bin`範例列出目前保留在資源回收筒中的所有映像。

```
aws ec2 list-images-in-recycle-bin
```

輸出：

```
{
    "Images": [
        {
```

```
            "RecycleBinEnterTime": "2022-03-14T15:35:08.000Z",
            "Description": "Monthly AMI One",
            "RecycleBinExitTime": "2022-03-15T15:35:08.000Z",
            "Name": "AMI_01",
            "ImageId": "ami-0111222333444abcd"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EBS 使用者指南》中的從資源回收筒復原已刪除的 AMIs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListImagesInRecycleBin。

list-snapshots-in-recycle-bin

以下程式碼範例顯示如何使用 list-snapshots-in-recycle-bin。

AWS CLI

檢視資源回收筒中的快照

下列list-snapshots-in-recycle-bin範例列出資源回收筒中快照的相關資訊，包括快照 ID、快照描述、建立快照的磁碟區 ID、刪除快照並進入資源回收筒的日期和時間，以及保留期間過期的日期和時間。

```
aws ec2 list-snapshots-in-recycle-bin \
    --snapshot-id snap-01234567890abcdef
```

輸出：

```
{
    "SnapshotRecycleBinInfo": [
        {
            "Description": "Monthly data backup snapshot",
            "RecycleBinEnterTime": "2022-12-01T13:00:00.000Z",
            "RecycleBinExitTime": "2022-12-15T13:00:00.000Z",
            "VolumeId": "vol-abcdef09876543210",
            "SnapshotId": "snap-01234567890abcdef"
        }
    ]
}
```

如需資源回收筒的詳細資訊，請參閱《Amazon EBS 使用者指南》中的從資源回收筒復原已刪除的快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSnapshotsInRecycleBin。

`lock-snapshot`

以下程式碼範例顯示如何使用 `lock-snapshot`。

AWS CLI

範例 1：在控管模式中鎖定快照

下列`lock-snapshot`範例會在控管模式中鎖定指定的快照。

```
aws ec2 lock-snapshot \
    --snapshot-id snap-0b5e733b4a8df6e0d \
    --lock-mode governance \
    --lock-duration 365
```

輸出：

```
{
    "SnapshotId": "snap-0b5e733b4a8df6e0d",
    "LockState": "governance",
    "LockDuration": 365,
    "LockCreatedOn": "2024-05-05T00:56:06.208000+00:00",
    "LockExpiresOn": "2025-05-05T00:56:06.208000+00:00",
    "LockDurationStartTime": "2024-05-05T00:56:06.208000+00:00"
}
```

如需詳細資訊，請參閱《Amazon EBS 使用者指南》中的快照鎖定。

範例 2：在合規模式下鎖定快照

下列`lock-snapshot`範例會在合規模式下鎖定指定的快照。

```
aws ec2 lock-snapshot \
    --snapshot-id snap-0163a8524c5b9901f \
    --lock-mode compliance \
    --cool-off-period 24 \
```

```
    --lock-duration 365
```

輸出：

```
{
    "SnapshotId": "snap-0b5e733b4a8df6e0d",
    "LockState": "compliance-cooloff",
    "LockDuration": 365,
    "CoolOffPeriod": 24,
    "CoolOffPeriodExpiresOn": "2024-05-06T01:02:20.527000+00:00",
    "LockCreatedOn": "2024-05-05T01:02:20.527000+00:00",
    "LockExpiresOn": "2025-05-05T01:02:20.527000+00:00",
    "LockDurationStartTime": "2024-05-05T01:02:20.527000+00:00"
}
```

如需詳細資訊，請參閱《Amazon EBS 使用者指南》中的快照鎖定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 LockSnapshot。

modify-address-attribute

以下程式碼範例顯示如何使用 modify-address-attribute。

AWS CLI

修改與彈性 IP 地址相關聯的網域名稱屬性

下列modify-address-attribute範例會修改彈性 IP 地址的網域名稱屬性。

Linux：

```
aws ec2 modify-address-attribute \
    --allocation-id eipalloc-abcdef01234567890 \
    --domain-name example.com
```

Windows：

```
aws ec2 modify-address-attribute ^
    --allocation-id eipalloc-abcdef01234567890 ^
    --domain-name example.com
```

輸出：

```
{
    "Addresses": [
        {
            "PublicIp": "192.0.2.0",
            "AllocationId": "eipalloc-abcdef01234567890",
            "PtrRecord": "example.net."
            "PtrRecordUpdate": {
                "Value": "example.com.",
                "Status": "PENDING"
        }
    ]
}
```

若要監控待定變更並檢視彈性 IP 地址的修改屬性，請參閱 AWS CLI 命令參考中的 describe-addresses-attribute。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyAddressAttribute。

modify-availability-zone-group

以下程式碼範例顯示如何使用 modify-availability-zone-group。

AWS CLI

啟用區域群組

下列modify-availability-zone-group範例會啟用指定的區域群組。

```
aws ec2 modify-availability-zone-group \
    --group-name us-west-2-lax-1 \
    --opt-in-status opted-in
```

輸出：

```
{
    "Return": true
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的區域 (Region) 和區域 (Zone)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyAvailabilityZoneGroup。

modify-capacity-reservation-fleet

以下程式碼範例顯示如何使用 modify-capacity-reservation-fleet。

AWS CLI

範例 1：修改容量保留機群的總目標容量

下列modify-capacity-reservation-fleet範例會修改指定容量保留機群的總目標容量。修改容量預留叢集的總目標容量時，機群會自動建立新的容量預留，或者修改或取消機群中的現有容量預留，以滿足新的總目標容量。您無法在機群處於 modifying 狀態時，嘗試對機群進行額外修改。

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --total-target-capacity 160
```

輸出：

```
{
    "Return": true
}
```

範例 2：修改容量保留機群的結束日期

下列modify-capacity-reservation-fleet範例會修改指定容量保留機群的結束日期。修改機群的結束日期時，會相應地更新所有個別容量預留的結束日期。您無法在機群處於 modifying 狀態時，嘗試對機群進行額外修改。

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --end-date 2022-07-04T23:59:59.000Z
```

輸出：

```
{
    "Return": true
}
```

如需容量保留機群的詳細資訊，請參閱《Amazon EC2 使用者指南》中的容量保留機群。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyCapacityReservationFleet。

modify-capacity-reservation

以下程式碼範例顯示如何使用 modify-capacity-reservation。

AWS CLI

範例 1：變更現有容量保留所保留的執行個體數量

下列modify-capacity-reservation範例會變更容量保留保留容量的執行個體數量。

```
aws ec2 modify-capacity-reservation \
    --capacity-reservation-id cr-1234abcd56EXAMPLE \
    --instance-count 5
```

輸出：

```
{
    "Return": true
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的修改容量保留。

範例 2：變更現有容量保留的結束日期和時間

下列modify-capacity-reservation範例會將現有的容量保留修改為在指定的日期和時間結束。

```
aws ec2 modify-capacity-reservation \
    --capacity-reservation-id cr-1234abcd56EXAMPLE \
    --end-date-type limited \
    --end-date 2019-08-31T23:59:59Z
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的修改容量保留。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyCapacityReservation。

modify-client-vpn-endpoint

以下程式碼範例顯示如何使用 modify-client-vpn-endpoint。

AWS CLI

修改 Client VPN 端點

下列modify-client-vpn-endpoint範例會啟用指定 Client VPN 端點的用戶端連線記錄。

```
aws ec2 modify-client-vpn-endpoint \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --connection-log-options Enabled=true,CloudwatchLogGroup=ClientVPNLogs
```

輸出：

```
{
    "Return": true
}
```

如需詳細資訊，請參閱 Client VPN 管理員指南中的 Client VPN 端點。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyClientVpnEndpoint。

modify-default-credit-specification

以下程式碼範例顯示如何使用 modify-default-credit-specification。

AWS CLI

修改預設點數選項

下列modify-default-credit-specification範例會修改 T2 執行個體的預設點數選項。

```
aws ec2 modify-default-credit-specification \
    --instance-family t2 \
    --cpu-credits unlimited
```

輸出：

```
{
    "InstanceFamilyCreditSpecification": {
        "InstanceFamily": "t2",
        "CpuCredits": "unlimited"
    }
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDefaultCreditSpecification。

modify-ebs-default-kms-key-id

以下程式碼範例顯示如何使用 modify-ebs-default-kms-key-id。

AWS CLI

設定 EBS 加密的預設 CMK

下列modify-ebs-default-kms-key-id範例會將指定的 CMK 設定為目前區域中 AWS 您帳戶 EBS 加密的預設 CMK。

```
aws ec2 modify-ebs-default-kms-key-id \
    --kms-key-id alias/my-cmk
```

輸出：

```
{
    "KmsKeyId": "arn:aws:kms:us-
west-2:123456789012:key/0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyEbsDefaultKmsKeyId。

modify-fleet

以下程式碼範例顯示如何使用 modify-fleet。

AWS CLI

擴展 EC2 機群

下列modify-fleet範例會修改指定 EC2 機群的目標容量。如果指定的值大於目前的容量，EC2 機群會啟動其他執行個體。如果指定的值小於目前的容量，EC2 機群會取消任何開啟的請求，如果終止政策為 terminate，EC2 機群會終止超過新目標容量的任何執行個體。

```
aws ec2 modify-fleet \
```

```
    --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
    --target-capacity-specification TotalTargetCapacity=5
```

輸出：

```
{
    "Return": true
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的管理 EC2 機群。 Amazon EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyFleet。

modify-fpga-image-attribute

以下程式碼範例顯示如何使用 modify-fpga-image-attribute。

AWS CLI

修改 Amazon FPGA 映像的屬性

此範例會為指定的 AFI 新增帳戶 ID 123456789012的載入許可。

命令：

```
aws ec2 modify-fpga-image-attribute --attribute loadPermission --fpga-image-
id afi-0d123e123bfc85abc --load-permission Add=[{UserId=123456789012}]
```

輸出：

```
{
  "FpgaImageAttribute": {
      "FpgaImageId": "afi-0d123e123bfc85abc",
      "LoadPermissions": [
          {
              "UserId": "123456789012"
          }
      ]
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyFpgaImageAttribute。

modify-hosts

以下程式碼範例顯示如何使用 modify-hosts。

AWS CLI

範例 1：為專用主機啟用自動置放

下列modify-hosts範例會啟用專用主機的自動置放，使其接受任何與其執行個體類型組態相符的非目標執行個體啟動。

```
aws ec2 modify-hosts \
    --host-id h-06c2f189b4EXAMPLE \
    --auto-placement on
```

輸出：

```
{
    "Successful": [
        "h-06c2f189b4EXAMPLE"
    ],
    "Unsuccessful": []
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的修改專用主機的自動置放設定。

範例 2：啟用專用主機的主機復原

下列modify-hosts範例會啟用指定專用主機的主機復原。

```
aws ec2 modify-hosts \
    --host-id h-06c2f189b4EXAMPLE \
    --host-recovery on
```

輸出：

```
{
    "Successful": [
        "h-06c2f189b4EXAMPLE"
    ],
    "Unsuccessful": []
```

```
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的修改專用主機的自動置放設定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyHosts。

modify-id-format

以下程式碼範例顯示如何使用 modify-id-format。

AWS CLI

為資源啟用較長的 ID 格式

下列modify-id-format範例會啟用instance資源類型的較長 ID 格式。

```
aws ec2 modify-id-format \
    --resource instance \
    --use-long-ids
```

停用資源的較長 ID 格式

下列modify-id-format範例會停用instance資源類型的較長 ID 格式。

```
aws ec2 modify-id-format \
    --resource instance \
    --no-use-long-ids
```

下列modify-id-format範例會針對在其選擇加入期間內的所有支援資源類型，啟用較長的 ID 格式。

```
aws ec2 modify-id-format \
    --resource all-current \
    --use-long-ids
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyIdFormat。

modify-identity-id-format

以下程式碼範例顯示如何使用 modify-identity-id-format。

AWS CLI

若要讓 IAM 角色使用較長的資源 IDs

下列modify-identity-id-format範例可讓EC2Role AWS 帳戶中的 IAM 角色針對instance資源類型使用長 ID 格式。

```
aws ec2 modify-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:role/EC2Role \
    --resource instance \
    --use-long-ids
```

若要讓 IAM 使用者使用較長的資源 IDs

下列modify-identity-id-format範例可讓AdminUser AWS 帳戶中的 IAM 使用者針對volume資源類型使用較長的 ID 格式。

```
aws ec2 modify-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:user/AdminUser \
    --resource volume \
    --use-long-ids
```

下列modify-identity-id-format範例可讓AdminUser AWS 帳戶中的 IAM 使用者針對其選擇加入期間的所有支援資源類型，使用較長的 ID 格式。

```
aws ec2 modify-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:user/AdminUser \
    --resource all-current \
    --use-long-ids
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyIdentityIdFormat。

modify-image-attribute

以下程式碼範例顯示如何使用 modify-image-attribute。

AWS CLI

範例 1：將 AMI 設為公有

下列modify-instance-attribute範例會將指定的 AMI 設為公有。

```
aws ec2 modify-image-attribute \
    --image-id ami-5731123e \
    --launch-permission "Add=[{Group=all}]"
```

此命令不會產生輸出。

範例 2：將 AMI 設為私有

下列modify-instance-attribute範例會將指定的 AMI 設為私有。

```
aws ec2 modify-image-attribute \
    --image-id ami-5731123e \
    --launch-permission "Remove=[{Group=all}]"
```

此命令不會產生輸出。

範例 3：授予 AWS 帳戶的啟動許可

下列modify-instance-attribute範例會將啟動許可授予指定的 AWS 帳戶。

```
aws ec2 modify-image-attribute \
    --image-id ami-5731123e \
    --launch-permission "Add=[{UserId=123456789012}]"
```

此命令不會產生輸出。

範例 4：從 AWS 帳戶移除啟動許可

下列modify-instance-attribute範例會從指定的 AWS 帳戶移除啟動許可。

```
aws ec2 modify-image-attribute \
    --image-id ami-5731123e \
    --launch-permission "Remove=[{UserId=123456789012}]"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyImageAttribute。

modify-instance-attribute

以下程式碼範例顯示如何使用 modify-instance-attribute。

AWS CLI

範例 1：修改執行個體類型

下列modify-instance-attribute範例會修改指定執行個體的執行個體類型。執行個體必須處於 stopped 狀態。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --instance-type "{\"Value\": \"m1.small\"}"
```

此命令不會產生輸出。

範例 2：在執行個體上啟用增強型聯網

下列modify-instance-attribute範例會啟用指定執行個體的增強型聯網。執行個體必須處於 stopped 狀態。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --sriov-net-support simple
```

此命令不會產生輸出。

範例 3：修改 sourceDestCheck 屬性

下列modify-instance-attribute範例會將指定執行個體的 sourceDestCheck 屬性設定為 true。執行個體必須位於 VPC 中。

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --source-dest-
check "{\"Value\": true}"
```

此命令不會產生輸出。

範例 4：修改根磁碟區的 deleteOnTermination 屬性

下列modify-instance-attribute範例會將指定 Amazon EBS 後端執行個體根磁碟區的 deleteOnTermination 屬性設定為 false。根據預設，此屬性true適用於根磁碟區。

命令：

```
aws ec2 modify-instance-attribute \
```

```
  --instance-id i-1234567890abcdef0 \
  --block-device-mappings "[{\"DeviceName\": \"/dev/sda1\",\"Ebs\":
{\"DeleteOnTermination\":false}}]"
```

此命令不會產生輸出。

範例 5：修改連接至執行個體的使用者資料

下列modify-instance-attribute範例會將 檔案的內容新增UserData.txt為指定執行個體的 UserData。

原始檔案 的內容UserData.txt：

```
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

檔案的內容必須經過 base64 編碼。第一個命令會將文字檔案轉換為 base64，並將其儲存為新檔案。

Linux/macOS 版本的 命令：

```
base64 UserData.txt > UserData.base64.txt
```

此命令不會產生輸出。

Windows 版本的 命令：

```
certutil -encode UserData.txt tmp.b64 && findstr /v /c:- tmp.b64 >
 UserData.base64.txt
```

輸出：

```
Input Length = 67
Output Length = 152
CertUtil: -encode command completed successfully.
```

現在您可以在以下 CLI 命令中參考該檔案：

```
aws ec2 modify-instance-attribute \
```

```
    --instance-id=i-09b5a14dbca622e76 \
    --attribute userData --value file://UserData.base64.txt
```

此命令不會產生輸出。

如需詳細資訊，請參閱 EC2 AWS 使用者指南中的使用者資料和 CLI。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyInstanceAttribute。

modify-instance-capacity-reservation-attributes

以下程式碼範例顯示如何使用 modify-instance-capacity-reservation-attributes。

AWS CLI

範例 1：修改執行個體的容量保留目標設定

下列modify-instance-capacity-reservation-attributes範例會修改已停止的執行個體，以鎖定特定容量保留。

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-EXAMPLE8765abcd4e \
    --capacity-reservation-specification
 'CapacityReservationTarget={CapacityReservationId= cr-1234abcd56EXAMPLE }'
```

輸出：

```
{
    "Return": true
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的修改執行個體的容量保留設定。

範例 2：修改執行個體的容量保留目標設定

下列modify-instance-capacity-reservation-attributes範例會修改已停止的執行個體，以指定容量保留為目標，在具有相符屬性（執行個體類型、平台、可用區域）且具有開放執行個體相符條件的任何容量保留中啟動。

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-EXAMPLE8765abcd4e \
```

```
    --capacity-reservation-specification 'CapacityReservationPreference=open'
```

輸出：

```
{
    "Return": true
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的修改執行個體的容量保留設定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyInstanceCapacityReservationAttributes。

modify-instance-credit-specification

以下程式碼範例顯示如何使用 modify-instance-credit-specification。

AWS CLI

修改執行個體 CPU 用量的點數選項

此範例會將指定區域中指定執行個體的 CPU 用量點數選項修改為「無限制」。有效點數選項為「標準」和「無限」。

命令：

```
aws ec2 modify-instance-credit-specification --instance-credit-
specification "InstanceId=i-1234567890abcdef0,CpuCredits=unlimited"
```

輸出：

```
{
  "SuccessfulInstanceCreditSpecifications": [
    {
      "InstanceId": "i-1234567890abcdef0"
    }
  ],
  "UnsuccessfulInstanceCreditSpecifications": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyInstanceCreditSpecification。

modify-instance-event-start-time

以下程式碼範例顯示如何使用 modify-instance-event-start-time。

AWS CLI

修改執行個體的事件開始時間

下列modify-instance-event-start-time命令顯示如何修改指定執行個體的事件開始時間。使用 --instance-event-id 參數指定事件 ID。使用 --not-before 參數指定新的日期和時間。

```
aws ec2 modify-instance-event-start-time --instance-id i-1234567890abcdef0
 --instance-event-id instance-event-0abcdef1234567890 --not-
before 2019-03-25T10:00:00.000
```

輸出：

```
"Event": {
    "InstanceEventId": "instance-event-0abcdef1234567890",
    "Code": "system-reboot",
    "Description": "scheduled reboot",
    "NotAfter": "2019-03-25T12:00:00.000Z",
    "NotBefore": "2019-03-25T10:00:00.000Z",
    "NotBeforeDeadline": "2019-04-22T21:00:00.000Z"
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的使用排程重新啟動的執行個體

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyInstanceEventStartTime。

modify-instance-event-window

以下程式碼範例顯示如何使用 modify-instance-event-window。

AWS CLI

範例 1：修改事件時段的時間範圍

下列modify-instance-event-window範例會修改事件時段的時間範圍。指定 time-range 參數來修改時間範圍。您不能指定 cron-expression 參數。

```
aws ec2 modify-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890
    --time-range StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8
```

輸出：

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "TimeRanges": [
            {
                "StartWeekDay": "monday",
                "StartHour": 2,
                "EndWeekDay": "wednesday",
                "EndHour": 8
            }
        ],
        "Name": "myEventWindowName",
        "AssociationTarget": {
            "InstanceIds": [
                "i-0abcdef1234567890",
                "i-0be35f9acb8ba01f0"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

如需事件時段限制，請參閱《Amazon EC2 使用者指南》的排程事件一節中的考量事項。

範例 2：修改事件時段的一組時間範圍

下列modify-instance-event-window範例會修改事件時段的時間範圍。指定 time-range 參數來修改時間範圍。您不能指定 cron-expression 參數。

```
aws ec2 modify-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --time-range '[{"StartWeekDay": "monday", "StartHour": 2, "EndWeekDay":
 "wednesday", "EndHour": 8},
        {"StartWeekDay": "thursday", "StartHour": 2, "EndWeekDay": "friday",
 "EndHour": 8}]'
```

輸出：

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "TimeRanges": [
            {
                "StartWeekDay": "monday",
                "StartHour": 2,
                "EndWeekDay": "wednesday",
                "EndHour": 8
            },
            {
                "StartWeekDay": "thursday",
                "StartHour": 2,
                "EndWeekDay": "friday",
                "EndHour": 8
            }
        ],
        "Name": "myEventWindowName",
        "AssociationTarget": {
            "InstanceIds": [
                "i-0abcdef1234567890",
                "i-0be35f9acb8ba01f0"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
```

```
    }
}
```

如需事件時段限制，請參閱《Amazon EC2 使用者指南》的排程事件一節中的考量事項。

範例 3：修改事件視窗的 Cron 表達式

下列modify-instance-event-window範例會修改事件視窗的 Cron 表達式。指定 cron-expression 參數來修改 cron 表達式。您不能指定 time-range 參數。

```
aws ec2 modify-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --cron-expression "* 21-23 * * 2,3"
```

輸出：

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [
                "i-0abcdef1234567890",
                "i-0be35f9acb8ba01f0"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

如需事件時段限制，請參閱《Amazon EC2 使用者指南》的排程事件一節中的考量事項。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyInstanceEventWindow。

modify-instance-maintenance-options

以下程式碼範例顯示如何使用 modify-instance-maintenance-options。

AWS CLI

範例 1：停用執行個體的復原行為

下列modify-instance-maintenance-options範例會停用執行中或已停止執行個體的簡化自動復原。

```
aws ec2 modify-instance-maintenance-options \
    --instance-id i-0abcdef1234567890 \
    --auto-recovery disabled
```

輸出：

```
{
    "InstanceId": "i-0abcdef1234567890",
    "AutoRecovery": "disabled"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的設定簡化的自動復原。

範例 2：將執行個體的復原行為設為預設

下列modify-instance-maintenance-options範例會將自動復原行為設定為預設值，以簡化受支援執行個體類型的自動復原。

```
aws ec2 modify-instance-maintenance-options \
    --instance-id i-0abcdef1234567890 \
    --auto-recovery default
```

輸出：

```
{
    "InstanceId": "i-0abcdef1234567890",
    "AutoRecovery": "default"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的設定簡化的自動復原。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyInstanceMaintenanceOptions。

modify-instance-metadata-options

以下程式碼範例顯示如何使用 modify-instance-metadata-options。

AWS CLI

範例 1：啟用 IMDSv2

下列modify-instance-metadata-options範例會設定在指定執行個體上使用 IMDSv2。

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567898abcdef0 \
    --http-tokens required \
    --http-endpoint enabled
```

輸出：

```
{
    "InstanceId": "i-1234567898abcdef0",
    "InstanceMetadataOptions": {
        "State": "pending",
        "HttpTokens": "required",
        "HttpPutResponseHopLimit": 1,
        "HttpEndpoint": "enabled"
    }
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的執行個體中繼資料。

範例 2：停用執行個體中繼資料

下列modify-instance-metadata-options範例會停用指定執行個體上所有版本的執行個體中繼資料。

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567898abcdef0 \
    --http-endpoint disabled
```

輸出：

```
{
```

```
    "InstanceId": "i-1234567898abcdef0",
    "InstanceMetadataOptions": {
        "State": "pending",
        "HttpTokens": "required",
        "HttpPutResponseHopLimit": 1,
        "HttpEndpoint": "disabled"
    }
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的執行個體中繼資料。

範例 3：為您的執行個體啟用執行個體中繼資料 IPv6 端點

下列modify-instance-metadata-options範例說明如何開啟執行個體中繼資料服務的 IPv6 端點。依預設，停用 IPv6 端點。即使您已將執行個體啟動至僅限 IPv6 的子網，也是如此。IMDS 的 IPv6 端點只能在建置於 Nitro 系統的執行個體上存取。

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567898abcdef0 \
    --http-protocol-ipv6 enabled \
    --http-endpoint enabled
```

輸出：

```
{
    "InstanceId": "i-1234567898abcdef0",
    "InstanceMetadataOptions": {
        "State": "pending",
        "HttpTokens": "required",
        "HttpPutResponseHopLimit": 1,
        "HttpEndpoint": "enabled",
        HttpProtocolIpv6": "enabled"
    }
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的執行個體中繼資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyInstanceMetadataOptions。

modify-instance-placement

以下程式碼範例顯示如何使用 modify-instance-placement。

AWS CLI

範例 1：移除執行個體與專用主機的親和性

下列modify-instance-placement範例會移除執行個體與專用主機的親和性，並讓它在您帳戶中支援其執行個體類型的任何可用專用主機上啟動。

```
aws ec2 modify-instance-placement \
    --instance-id i-0e6ddf6187EXAMPLE \
    --affinity default
```

輸出：

```
{
    "Return": true
}
```

範例 2：在執行個體與指定的專用主機之間建立親和性

下列modify-instance-placement範例會在執行個體與專用主機之間建立啟動關係。執行個體只能在指定的專用主機上執行。

```
aws ec2 modify-instance-placement \
    --instance-id i-0e6ddf6187EXAMPLE \
    --affinity host \
    --host-id i-0e6ddf6187EXAMPLE
```

輸出：

```
{
    "Return": true
}
```

範例 3：將執行個體移至置放群組

下列modify-instance-placement範例會將執行個體移至置放群組、停止執行個體、修改執行個體置放，然後重新啟動執行個體。

```
aws ec2 stop-instances \
    --instance-ids i-0123a456700123456
```

```
aws ec2 modify-instance-placement \
    --instance-id i-0123a456700123456 \
    --group-name MySpreadGroup

aws ec2 start-instances \
    --instance-ids i-0123a456700123456
```

範例 4：從置放群組中移除執行個體

下列modify-instance-placement範例會停止執行個體、修改執行個體置放，然後重新啟動執行個體，以從置放群組移除執行個體。下列範例會指定置放群組名稱的空字串 ("")，以表示執行個體不位於置放群組中。

停止執行個體：

```
aws ec2 stop-instances \
    --instance-ids i-0123a456700123456
```

修改置放 (Windows 命令提示）：

```
aws ec2 modify-instance-placement \
    --instance-id i-0123a456700123456 \
    --group-name ""
```

修改置放 (Windows PowerShell、Linux 和 macOS)：

```
aws ec2 modify-instance-placement `
    --instance-id i-0123a456700123456 `
    --group-name ''
```

重新啟動執行個體：

```
aws ec2 start-instances \
    --instance-ids i-0123a456700123456
```

輸出：

```
{
    "Return": true
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的修改專用主機租用和親和性。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyInstancePlacement。

modify-ipam-pool

以下程式碼範例顯示如何使用 modify-ipam-pool。

AWS CLI

修改 IPAM 集區

下列modify-ipam-pool範例會修改 IPAM 集區。

(Linux)：

```
aws ec2 modify-ipam-pool \
    --ipam-pool-id ipam-pool-0533048da7d823723 \
    --add-allocation-resource-tags "Key=Owner,Value=Build Team" \
    --clear-allocation-default-netmask-length \
    --allocation-min-netmask-length 14
```

(Windows)：

```
aws ec2 modify-ipam-pool ^
    --ipam-pool-id ipam-pool-0533048da7d823723 ^
    --add-allocation-resource-tags "Key=Owner,Value=Build Team" ^
    --clear-allocation-default-netmask-length ^
    --allocation-min-netmask-length 14
```

輸出：

```
{
    "IpamPool": {
        "OwnerId": "123456789012",
        "IpamPoolId": "ipam-pool-0533048da7d823723",
        "IpamPoolArn": "arn:aws:ec2::123456789012:ipam-pool/ipam-
pool-0533048da7d823723",
        "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-02fc38cd4c48e7d38",
        "IpamScopeType": "private",
        "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
        "IpamRegion": "us-east-1",
```

```
        "Locale": "None",
        "PoolDepth": 1,
        "State": "modify-complete",
        "AutoImport": true,
        "AddressFamily": "ipv4",
        "AllocationMinNetmaskLength": 14,
        "AllocationMaxNetmaskLength": 26,
        "AllocationResourceTags": [
            {
                "Key": "Environment",
                "Value": "Preprod"
            },
            {
                "Key": "Owner",
                "Value": "Build Team"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的編輯集區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyIpamPool。

modify-ipam-resource-cidr

以下程式碼範例顯示如何使用 modify-ipam-resource-cidr。

AWS CLI

修改配置給資源的 CIDR

下列modify-ipam-resource-cidr範例會修改資源 CIDR。

(Linux)：

```
aws ec2 modify-ipam-resource-cidr \
    --current-ipam-scope-id ipam-scope-02fc38cd4c48e7d38 \
    --destination-ipam-scope-id ipam-scope-0da34c61fd189a141 \
    --resource-id vpc-010e1791024eb0af9 \
    --resource-cidr 10.0.1.0/24 \
    --resource-region us-east-1 \
    --monitored
```

(Windows)：

```
aws ec2 modify-ipam-resource-cidr ^
    --current-ipam-scope-id ipam-scope-02fc38cd4c48e7d38 ^
    --destination-ipam-scope-id ipam-scope-0da34c61fd189a141 ^
    --resource-id vpc-010e1791024eb0af9 ^
    --resource-cidr 10.0.1.0/24 ^
    --resource-region us-east-1 ^
    --monitored
```

輸出：

```
{
    "IpamResourceCidr": {
        "IpamId": "ipam-08440e7a3acde3908",
        "IpamScopeId": "ipam-scope-0da34c61fd189a141",
        "IpamPoolId": "ipam-pool-0533048da7d823723",
        "ResourceRegion": "us-east-1",
        "ResourceOwnerId": "123456789012",
        "ResourceId": "vpc-010e1791024eb0af9",
        "ResourceCidr": "10.0.1.0/24",
        "ResourceType": "vpc",
        "ResourceTags": [
            {
                "Key": "Environment",
                "Value": "Preprod"
            },
            {
                "Key": "Owner",
                "Value": "Build Team"
            }
        ],
        "IpUsage": 0.0,
        "ComplianceStatus": "noncompliant",
        "ManagementState": "managed",
        "OverlapStatus": "overlapping",
        "VpcId": "vpc-010e1791024eb0af9"
    }
}
```

如需移動資源的詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的在範圍之間移動資源 CIDRs。

如需變更監控狀態的詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的變更資源 CIDRs 的監控狀態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyIpamResourceCidr。

`modify-ipam-resource-discovery`

以下程式碼範例顯示如何使用 `modify-ipam-resource-discovery`。

AWS CLI

修改資源探索的操作區域

在此範例中，您是 IPAM 委派的管理員，想要修改資源探索的操作區域。

若要完成此請求：

您無法修改預設資源探索，而且必須是資源探索的擁有者。您需要資源探索 ID，您可以使用 describe-ipam-resource-discoveries 取得。

下列`modify-ipam-resource-discovery`範例會修改您 AWS 帳戶中的非預設資源探索。

```
aws ec2 modify-ipam-resource-discovery \
    --ipam-resource-discovery-id ipam-res-disco-0f4ef577a9f37a162 \
    --add-operating-regions RegionName='us-west-1' \
    --remove-operating-regions RegionName='us-east-2' \
    --region us-east-1
```

輸出：

```
{
    "IpamResourceDiscovery": {
        "OwnerId": "149977607591",
        "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
        "IpamResourceDiscoveryArn": "arn:aws:ec2::149977607591:ipam-resource-
discovery/ipam-res-disco-0365d2977fc1672fe",
        "IpamResourceDiscoveryRegion": "us-east-1",
        "Description": "Example",
        "OperatingRegions": [
            {
                "RegionName": "us-east-1"
            },
            {
```

```
                "RegionName": "us-west-1"
            }
        ],
        "IsDefault": false,
        "State": "modify-in-progress"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的使用資源探索。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyIpamResourceDiscovery。

`modify-ipam-scope`

以下程式碼範例顯示如何使用 `modify-ipam-scope`。

AWS CLI

修改範圍的描述

在此案例中，您是 IPAM 委派的管理員，想要修改 IPAM 範圍的描述。

若要完成此請求，您將需要範圍 ID，您可以使用 describe-ipam-scopes 取得。

下列`modify-ipam-scope`範例會更新範圍的描述。

```
aws ec2 modify-ipam-scope \
    --ipam-scope-id ipam-scope-0d3539a30b57dcdd1 \
    --description example \
    --region us-east-1
```

輸出：

```
{
"IpamScope": {
        "OwnerId": "320805250157",
        "IpamScopeId": "ipam-scope-0d3539a30b57dcdd1",
        "IpamScopeArn": "arn:aws:ec2::320805250157:ipam-scope/ipam-
scope-0d3539a30b57dcdd1",
        "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
        "IpamRegion": "us-east-1",
        "IpamScopeType": "public",
        "IsDefault": true,
```

```
        "Description": "example",
        "PoolCount": 1,
        "State": "modify-in-progress"
    }
}
```

如需範圍的詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的 IPAM 運作方式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyIpamScope。

`modify-ipam`

以下程式碼範例顯示如何使用 `modify-ipam`。

AWS CLI

修改 IPAM

下列`modify-ipam`範例透過新增 操作區域來修改 IPAM。

(Linux)：

```
aws ec2 modify-ipam \
    --ipam-id ipam-08440e7a3acde3908 \
    --add-operating-regions RegionName=us-west-2
```

(Windows)：

```
aws ec2 modify-ipam ^
    --ipam-id ipam-08440e7a3acde3908 ^
    --add-operating-regions RegionName=us-west-2
```

輸出：

```
{
    "Ipam": {
        "OwnerId": "123456789012",
        "IpamId": "ipam-08440e7a3acde3908",
        "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
        "IpamRegion": "us-east-1",
        "PublicDefaultScopeId": "ipam-scope-0b9eed026396dbc16",
        "PrivateDefaultScopeId": "ipam-scope-02fc38cd4c48e7d38",
```

```
        "ScopeCount": 3,
        "OperatingRegions": [
            {
                "RegionName": "us-east-1"
            },
            {
                "RegionName": "us-east-2"
            },
            {
                "RegionName": "us-west-1"
            },
            {
                "RegionName": "us-west-2"
            }
        ],
        "State": "modify-in-progress"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyIpam。

modify-launch-template

以下程式碼範例顯示如何使用 modify-launch-template。

AWS CLI

變更預設啟動範本版本

此範例會將指定啟動範本的第 2 版指定為預設版本。

命令：

```
aws ec2 modify-launch-template --launch-template-id lt-0abcd290751193123 --default-version 2
```

輸出：

```
{
  "LaunchTemplate": {
      "LatestVersionNumber": 2,
      "LaunchTemplateId": "lt-0abcd290751193123",
```

```
      "LaunchTemplateName": "WebServers",
      "DefaultVersionNumber": 2,
      "CreatedBy": "arn:aws:iam::123456789012:root",
      "CreateTime": "2017-12-01T13:35:46.000Z"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyLaunchTemplate。

modify-managed-prefix-list

以下程式碼範例顯示如何使用 modify-managed-prefix-list。

AWS CLI

修改字首清單

下列modify-managed-prefix-list範例會將項目新增至指定的字首清單。

```
aws ec2 modify-managed-prefix-list \
    --prefix-list-id pl-0123456abcabcabc1 \
    --add-entries Cidr=10.1.0.0/16,Description=vpc-c \
    --current-version 1
```

輸出：

```
{
    "PrefixList": {
        "PrefixListId": "pl-0123456abcabcabc1",
        "AddressFamily": "IPv4",
        "State": "modify-in-progress",
        "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-0123456abcabcabc1",
        "PrefixListName": "vpc-cidrs",
        "MaxEntries": 10,
        "Version": 1,
        "OwnerId": "123456789012"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的受管字首清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyManagedPrefixList。

modify-network-interface-attribute

以下程式碼範例顯示如何使用 modify-network-interface-attribute。

AWS CLI

修改網路界面的附件屬性

此範例命令會修改指定網路界面的 attachment 屬性。

命令：

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --
attachment AttachmentId=eni-attach-43348162,DeleteOnTermination=false
```

修改網路界面的描述屬性

此範例命令會修改指定網路界面的 description 屬性。

命令：

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --
description "My description"
```

修改網路界面的 groupSet 屬性

此範例命令會修改指定網路界面的 groupSet 屬性。

命令：

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --
groups sg-903004f8 sg-1a2b3c4d
```

修改網路界面的 sourceDestCheck 屬性

此範例命令會修改指定網路界面的 sourceDestCheck 屬性。

命令：

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --no-
source-dest-check
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyNetworkInterfaceAttribute。

modify-private-dns-name-options

以下程式碼範例顯示如何使用 modify-private-dns-name-options。

AWS CLI

修改執行個體主機名稱的選項

下列modify-private-dns-name-options範例停用 選項，以使用 DNS A 記錄回應執行個體主機名稱的 DNS 查詢。

```
aws ec2 modify-private-dns-name-options \
    --instance-id i-1234567890abcdef0 \
    --no-enable-resource-name-dns-a-record
```

輸出：

```
{
    "Return": true
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Amazon EC2 執行個體主機名稱類型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyPrivateDnsNameOptions。

modify-reserved-instances

以下程式碼範例顯示如何使用 modify-reserved-instances。

AWS CLI

修改預留執行個體

此範例命令會將預留執行個體移至相同區域中的另一個可用區域。

命令：

```
aws ec2 modify-reserved-instances --reserved-instances-ids b847fa93-e282-4f55-b59a-1342f5bd7c02 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-Classic,InstanceCount=10
```

輸出：

```
{
  "ReservedInstancesModificationId": "rimod-d3ed4335-b1d3-4de6-ab31-0f13aaf46687"
}
```

修改預留執行個體的網路平台

此範例命令會將 EC2-Classic 預留執行個體轉換為 EC2-VPC。

命令：

```
aws ec2 modify-reserved-instances --reserved-instances-ids f127bd27-edb7-44c9-a0eb-0d7e09259af0 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-VPC,InstanceCount=5
```

輸出：

```
{
  "ReservedInstancesModificationId": "rimod-82fa9020-668f-4fb6-945d-61537009d291"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的修改預留執行個體。

修改預留執行個體的執行個體大小

此範例命令修改的預留執行個體在 us-west-1c 中具有 10 m1.small Linux/UNIX 執行個體，使得 8 m1.small 執行個體成為 2 m1.large 執行個體，其餘 2 m1.small 則成為相同可用區域中的 1 m1.medium 執行個體。命令：

```
aws ec2 modify-reserved-instances --reserved-instances-ids 1ba8e2e3-3556-4264-949e-63ee671405a9 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-Classic,InstanceCount=2,InstanceType=m1.large AvailabilityZone=us-west-1c,Platform=EC2-Classic,InstanceCount=1,InstanceType=m1.medium
```

輸出：

```
{
    "ReservedInstancesModificationId": "rimod-acc5f240-080d-4717-b3e3-1c6b11fa00b6"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的修改預留的執行個體大小。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyReservedInstances。

modify-security-group-rules

以下程式碼範例顯示如何使用 modify-security-group-rules。

AWS CLI

修改安全群組規則以更新規則描述、IP 通訊協定和 CidrIpv4 地址範圍

下列modify-security-group-rules範例會更新指定安全群組規則的描述、IP 通訊協定和 IPV4 CIDR 範圍。使用 security-group-rules 參數輸入指定安全群組規則的更新。 -1會指定所有通訊協定。

```
aws ec2 modify-security-group-rules \
    --group-id sg-1234567890abcdef0 \
    --security-group-rules SecurityGroupRuleId=sgr-
abcdef01234567890,SecurityGroupRule='{Description=test,IpProtocol=-1,CidrIpv4=0.0.0.0/0}'
```

輸出：

```
{
    "Return": true
}
```

如需安全群組規則的詳細資訊，請參閱《Amazon EC2 使用者指南》中的安全群組規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifySecurityGroupRules。

modify-snapshot-attribute

以下程式碼範例顯示如何使用 modify-snapshot-attribute。

AWS CLI

範例 1：修改快照屬性

下列modify-snapshot-attribute範例會更新指定快照的 createVolumePermission 屬性，移除指定使用者的磁碟區許可。

```
aws ec2 modify-snapshot-attribute \
```

```
    --snapshot-id snap-1234567890abcdef0 \
    --attribute createVolumePermission \
    --operation-type remove \
    --user-ids 123456789012
```

範例 2：將快照設為公有

下列modify-snapshot-attribute範例會將指定的快照設為公有。

```
aws ec2 modify-snapshot-attribute \
    --snapshot-id snap-1234567890abcdef0 \
    --attribute createVolumePermission \
    --operation-type add \
    --group-names all
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifySnapshotAttribute。

modify-snapshot-tier

以下程式碼範例顯示如何使用 modify-snapshot-tier。

AWS CLI

封存快照

下列modify-snapshot-tier範例會封存指定的快照。TieringStartTime 回應參數指出封存程序的啟動日期和時間，以 UTC 時間格式 (YYYY-MM-DDTHH:MM:SSZ) 表示。

```
aws ec2 modify-snapshot-tier \
    --snapshot-id snap-01234567890abcedf \
    --storage-tier archive
```

輸出：

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "TieringStartTime": "2021-09-15T16:44:37.574Z"
}
```

如需快照封存的詳細資訊，請參閱《Amazon EBS 使用者指南》中的封存 Amazon EBS 快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifySnapshotTier。

modify-spot-fleet-request

以下程式碼範例顯示如何使用 modify-spot-fleet-request。

AWS CLI

修改 Spot 機群請求

此範例命令會更新指定 Spot 機群請求的目標容量。

命令：

```
aws ec2 modify-spot-fleet-request --target-capacity 20 --spot-fleet-request-
id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

輸出：

```
{
    "Return": true
}
```

此範例命令會減少指定 Spot 機群請求的目標容量，而不會因此終止任何 Spot 執行個體。

命令：

```
aws ec2 modify-spot-fleet-request --target-capacity 10 --excess-capacity-
termination-policy NoTermination --spot-fleet-request-ids sfr-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE
```

輸出：

```
{
    "Return": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifySpotFleetRequest。

modify-subnet-attribute

以下程式碼範例顯示如何使用 modify-subnet-attribute。

AWS CLI

變更子網路的公有 IPv4 定址行為

此範例修改 subnet-1a2b3c4d，以指定在此子網路中啟動的所有執行個體都會指派公有 IPv4 地址。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --map-public-ip-on-launch
```

變更子網路的 IPv6 定址行為

此範例修改 subnet-1a2b3c4d，以指定在此子網路中啟動的所有執行個體都會從子網路範圍指派 IPv6 地址。

命令：

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --assign-ipv6-address-on-creation
```

如需詳細資訊，請參閱 AWS Virtual Private Cloud 使用者指南中的 VPC 中的 IP 定址。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifySubnetAttribute。

modify-traffic-mirror-filter-network-services

以下程式碼範例顯示如何使用 modify-traffic-mirror-filter-network-services。

AWS CLI

將網路服務新增至流量鏡射篩選條件

下列modify-traffic-mirror-filter-network-services範例會將 Amazon DNS 網路服務新增至指定的篩選條件。

```
aws ec2 modify-traffic-mirror-filter-network-services \
    --traffic-mirror-filter-id tmf-04812ff784EXAMPLE \
    --add-network-service amazon-dns
```

輸出：

```
{
    "TrafficMirrorFilter": {
        "Tags": [
            {
                "Key": "Name",
                "Value": "Production"
            }
        ],
        "EgressFilterRules": [],
        "NetworkServices": [
            "amazon-dns"
        ],
        "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
        "IngressFilterRules": [
            {
                "SourceCidrBlock": "0.0.0.0/0",
                "RuleNumber": 1,
                "DestinationCidrBlock": "0.0.0.0/0",
                "Description": "TCP Rule",
                "Protocol": 6,
                "TrafficDirection": "ingress",
                "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
                "RuleAction": "accept",
                "TrafficMirrorFilterRuleId": "tmf-04812ff784EXAMPLE"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《流量鏡射指南》中的修改流量鏡射篩選條件網路服務。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyTrafficMirrorFilterNetworkServices。

modify-traffic-mirror-filter-rule

以下程式碼範例顯示如何使用 modify-traffic-mirror-filter-rule。

AWS CLI

修改流量鏡像篩選條件規則

下列modify-traffic-mirror-filter-rule範例會修改指定流量鏡像篩選條件規則的描述。

```
aws ec2 modify-traffic-mirror-filter-rule \
    --traffic-mirror-filter-rule-id tmfr-0ca76e0e08EXAMPLE \
    --description "TCP Rule"
```

輸出：

```
{
    "TrafficMirrorFilterRule": {
        "TrafficMirrorFilterRuleId": "tmfr-0ca76e0e08EXAMPLE",
        "TrafficMirrorFilterId": "tmf-0293f26e86EXAMPLE",
        "TrafficDirection": "ingress",
        "RuleNumber": 100,
        "RuleAction": "accept",
        "Protocol": 6,
        "DestinationCidrBlock": "10.0.0.0/24",
        "SourceCidrBlock": "10.0.0.0/24",
        "Description": "TCP Rule"
    }
}
```

如需詳細資訊，請參閱《流量鏡射指南》中的修改流量鏡射篩選條件規則。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyTrafficMirrorFilterRule。

modify-traffic-mirror-session

以下程式碼範例顯示如何使用 modify-traffic-mirror-session。

AWS CLI

修改流量鏡像工作階段

下列modify-traffic-mirror-session範例會變更流量鏡像工作階段描述和要鏡像的封包數量。

```
aws ec2 modify-traffic-mirror-session \
    --description "Change packet length" \
    --traffic-mirror-session-id tms-08a33b1214EXAMPLE \
    --remove-fields "packet-length"
```

輸出：

```
{
    "TrafficMirrorSession": {
        "TrafficMirrorSessionId": "tms-08a33b1214EXAMPLE",
        "TrafficMirrorTargetId": "tmt-07f75d8feeEXAMPLE",
        "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
        "NetworkInterfaceId": "eni-070203f901EXAMPLE",
        "OwnerId": "111122223333",
        "SessionNumber": 1,
        "VirtualNetworkId": 7159709,
        "Description": "Change packet length",
        "Tags": []
    }
}
```

如需詳細資訊，請參閱《流量鏡像指南》中的修改流量鏡像工作階段。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyTrafficMirrorSession。

modify-transit-gateway-prefix-list-reference

以下程式碼範例顯示如何使用 modify-transit-gateway-prefix-list-reference。

AWS CLI

修改字首清單的參考

下列modify-transit-gateway-prefix-list-reference範例會變更流量路由的附件，以修改指定路由表中的字首清單參考。

```
aws ec2 modify-transit-gateway-prefix-list-reference \
    --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
    --prefix-list-id pl-11111122222222333 \
    --transit-gateway-attachment-id tgw-attach-aabbccddaabbccaab
```

輸出：

```
{
    "TransitGatewayPrefixListReference": {
        "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
        "PrefixListId": "pl-11111122222222333",
```

```
        "PrefixListOwnerId": "123456789012",
        "State": "modifying",
        "Blackhole": false,
        "TransitGatewayAttachment": {
            "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",
            "ResourceType": "vpc",
            "ResourceId": "vpc-112233445566aabbc"
        }
    }
}
```

如需詳細資訊，請參閱《傳輸閘道指南》中的字首清單參考。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyTransitGatewayPrefixListReference。

`modify-transit-gateway-vpc-attachment`

以下程式碼範例顯示如何使用 `modify-transit-gateway-vpc-attachment`。

AWS CLI

修改傳輸閘道 VPC 連接

下列`modify-transit-gateway-vpc-attachment`範例會將子網路新增至指定的傳輸閘道 VPC 連接。

```
aws ec2 modify-transit-gateway-vpc-attachment \
    --transit-gateway-attachment-id tgw-attach-09fbd47ddfEXAMPLE \
    --add-subnet-ids subnet-0e51f45802EXAMPLE
```

輸出：

```
{
    "TransitGatewayVpcAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-09fbd47ddfEXAMPLE",
        "TransitGatewayId": "tgw-0560315ccfEXAMPLE",
        "VpcId": "vpc-5eccc927",
        "VpcOwnerId": "111122223333",
        "State": "modifying",
        "SubnetIds": [
            "subnet-0e51f45802EXAMPLE",
```

```
            "subnet-1EXAMPLE"
        ],
        "CreationTime": "2019-08-08T16:47:38.000Z",
        "Options": {
            "DnsSupport": "enable",
            "Ipv6Support": "disable"
        }
    }
}
```

如需詳細資訊，請參閱《傳輸閘道指南》中的傳輸閘道連接至 VPC。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyTransitGatewayVpcAttachment。

modify-transit-gateway

以下程式碼範例顯示如何使用 modify-transit-gateway。

AWS CLI

修改傳輸閘道

下列modify-transit-gateway範例透過啟用 VPN 附件的 ECMP 支援來修改指定的傳輸閘道。

```
aws ec2 modify-transit-gateway \
    --transit-gateway-id tgw-111111222222aaaaa \
    --options VpnEcmpSupport=enable
```

輸出：

```
{
    "TransitGateway": {
        "TransitGatewayId": "tgw-111111222222aaaaa",
        "TransitGatewayArn": "64512",
        "State": "modifying",
        "OwnerId": "123456789012",
        "CreationTime": "2020-04-30T08:41:37.000Z",
        "Options": {
            "AmazonSideAsn": 64512,
            "AutoAcceptSharedAttachments": "disable",
            "DefaultRouteTableAssociation": "enable",
```

```
            "AssociationDefaultRouteTableId": "tgw-rtb-0123456789abcd123",
            "DefaultRouteTablePropagation": "enable",
            "PropagationDefaultRouteTableId": "tgw-rtb-0123456789abcd123",
            "VpnEcmpSupport": "enable",
            "DnsSupport": "enable"
        }
    }
}
```

如需詳細資訊，請參閱《傳輸閘道指南》中的傳輸閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyTransitGateway。

modify-verified-access-endpoint-policy

以下程式碼範例顯示如何使用 modify-verified-access-endpoint-policy。

AWS CLI

設定端點的已驗證存取政策

下列modify-verified-access-endpoint-policy範例會將指定的 Verified Access 政策新增至指定的 Verified Access 端點。

```
aws ec2 modify-verified-access-endpoint-policy \
    --verified-access-endpoint-id vae-066fac616d4d546f2 \
    --policy-enabled \
    --policy-document file://policy.txt
```

policy.txt 的內容：

```
permit(principal,action,resource)
when {
    context.identity.groups.contains("finance") &&
    context.identity.email.verified == true
};
```

輸出：

```
{
    "PolicyEnabled": true,
```

```
    "PolicyDocument": "permit(principal,action,resource)\nwhen
 {\n    context.identity.groups.contains(\"finance\") &&\n
 context.identity.email_verified == true\n};"
}
```

如需詳細資訊，請參閱《Verified Access 使用者指南》中的 Verified Access 政策。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVerifiedAccessEndpointPolicy。

`modify-verified-access-endpoint`

以下程式碼範例顯示如何使用 `modify-verified-access-endpoint`。

AWS CLI

修改已驗證存取端點的組態

下列`modify-verified-access-endpoint`範例會將指定的描述新增至指定的 Verified Access 端點。

```
aws ec2 modify-verified-access-endpoint \
    --verified-access-endpoint-id vae-066fac616d4d546f2 \
    --description 'Testing Verified Access'
```

輸出：

```
{
    "VerifiedAccessEndpoint": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
        "VerifiedAccessEndpointId": "vae-066fac616d4d546f2",
        "ApplicationDomain": "example.com",
        "EndpointType": "network-interface",
        "AttachmentType": "vpc",
        "DomainCertificateArn": "arn:aws:acm:us-east-2:123456789012:certificate/
eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE",
        "EndpointDomain": "my-ava-
app.edge-00c3372d53b1540bb.vai-0ce000c0b7643abea.prod.verified-access.us-
east-2.amazonaws.com",
        "SecurityGroupIds": [
            "sg-004915970c4c8f13a"
        ],
```

```
        "NetworkInterfaceOptions": {
            "NetworkInterfaceId": "eni-0aec70418c8d87a0f",
            "Protocol": "https",
            "Port": 443
        },
        "Status": {
            "Code": "updating"
        },
        "Description": "Testing Verified Access",
        "CreationTime": "2023-08-25T20:54:43",
        "LastUpdatedTime": "2023-08-25T22:46:32"
    }
}
```

如需詳細資訊，請參閱 Verified Access 使用者指南中的 Verified Access 端點。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVerifiedAccessEndpoint。

modify-verified-access-group-policy

以下程式碼範例顯示如何使用 modify-verified-access-group-policy。

AWS CLI

設定群組的已驗證存取政策

下列modify-verified-access-group-policy範例會將指定的 Verified Access 政策新增至指定的 Verified Access 群組。

```
aws ec2 modify-verified-access-group-policy \
    --verified-access-group-id vagr-0dbe967baf14b7235 \
    --policy-enabled \
    --policy-document file://policy.txt
```

policy.txt 的內容：

```
permit(principal,action,resource)
when {
    context.identity.groups.contains("finance") &&
    context.identity.email.verified == true
};
```

輸出：

```
{
    "PolicyEnabled": true,
    "PolicyDocument": "permit(principal,action,resource)\nwhen
 {\n    context.identity.groups.contains(\"finance\") &&\n
 context.identity.email_verified == true\n};"
}
```

如需詳細資訊，請參閱《Verified Access 使用者指南》中的 Verified Access 群組。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVerifiedAccessGroupPolicy。

modify-verified-access-group

以下程式碼範例顯示如何使用 modify-verified-access-group。

AWS CLI

修改 Verified Access 群組的組態

下列modify-verified-access-group範例會將指定的描述新增至指定的 Verified Access 群組。

```
aws ec2 modify-verified-access-group \
    --verified-access-group-id vagr-0dbe967baf14b7235 \
    --description "Testing Verified Access"
```

輸出：

```
{
    "VerifiedAccessGroup": {
        "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "Description": "Testing Verified Access",
        "Owner": "123456789012",
        "VerifiedAccessGroupArn": "arn:aws:ec2:us-east-2:123456789012:verified-
access-group/vagr-0dbe967baf14b7235",
        "CreationTime": "2023-08-25T19:55:19",
        "LastUpdatedTime": "2023-08-25T22:17:25"
    }
}
```

如需詳細資訊，請參閱 Verified Access 使用者指南中的 Verified Access 群組。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVerifiedAccessGroup。

modify-verified-access-instance-logging-configuration

以下程式碼範例顯示如何使用 modify-verified-access-instance-logging-configuration。

AWS CLI

啟用 Verified Access 執行個體的記錄

下列modify-verified-access-instance-logging-configuration範例會啟用指定 Verified Access 執行個體的存取記錄。日誌會交付至指定的 CloudWatch Logs 日誌群組。

```
aws ec2 modify-verified-access-instance-logging-configuration \
    --verified-access-instance-id vai-0ce000c0b7643abea \
    --access-logs CloudWatchLogs={Enabled=true,LogGroup=my-log-group}
```

輸出：

```
{
    "LoggingConfiguration": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "AccessLogs": {
            "S3": {
                "Enabled": false
            },
            "CloudWatchLogs": {
                "Enabled": true,
                "DeliveryStatus": {
                    "Code": "success"
                },
                "LogGroup": "my-log-group"
            },
            "KinesisDataFirehose": {
                "Enabled": false
            },
            "LogVersion": "ocsf-1.0.0-rc.2",
            "IncludeTrustContext": false
        }
```

```
    }
}
```

如需詳細資訊，請參閱 Verified Access 使用者指南AWS 中的 Verified Access Logs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVerifiedAccessInstanceLoggingConfiguration。

`modify-verified-access-instance`

以下程式碼範例顯示如何使用 `modify-verified-access-instance`。

AWS CLI

修改 Verified Access 執行個體的組態

下列`modify-verified-access-instance`範例會將指定的描述新增至指定的 Verified Access 執行個體。

```
aws ec2 modify-verified-access-instance \
    --verified-access-instance-id vai-0ce000c0b7643abea \
    --description "Testing Verified Access"
```

輸出：

```
{
    "VerifiedAccessInstance": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "Description": "Testing Verified Access",
        "VerifiedAccessTrustProviders": [
            {
                "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
                "TrustProviderType": "user",
                "UserTrustProviderType": "iam-identity-center"
            }
        ],
        "CreationTime": "2023-08-25T18:27:56",
        "LastUpdatedTime": "2023-08-25T22:41:04"
    }
}
```

如需詳細資訊，請參閱《Verified Access 使用者指南》中的 Verified Access 執行個體。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVerifiedAccessInstance。

modify-verified-access-trust-provider

以下程式碼範例顯示如何使用 modify-verified-access-trust-provider。

AWS CLI

修改 Verified Access 信任提供者的組態

下列modify-verified-access-trust-provider範例會將指定的描述新增至指定的 Verified Access 信任提供者。

```
aws ec2 modify-verified-access-trust-provider \
    --verified-access-trust-provider-id vatp-0bb32de759a3e19e7 \
    --description "Testing Verified Access"
```

輸出：

```
{
    "VerifiedAccessTrustProvider": {
        "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
        "Description": "Testing Verified Access",
        "TrustProviderType": "user",
        "UserTrustProviderType": "iam-identity-center",
        "PolicyReferenceName": "idc",
        "CreationTime": "2023-08-25T19:00:38",
        "LastUpdatedTime": "2023-08-25T19:18:21"
    }
}
```

如需詳細資訊，請參閱《Verified Access 使用者指南》中的 Verified Access 的信任提供者。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVerifiedAccessTrustProvider。

modify-volume-attribute

以下程式碼範例顯示如何使用 modify-volume-attribute。

AWS CLI

修改磁碟區屬性

此範例會將 ID 為 的磁碟區autoEnableIo屬性vol-1234567890abcdef0設定為 true。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 modify-volume-attribute --volume-id vol-1234567890abcdef0 --auto-enable-io
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVolumeAttribute。

modify-volume

以下程式碼範例顯示如何使用 modify-volume。

AWS CLI

範例 1：變更磁碟區大小以修改磁碟區

下列modify-volume範例會將指定磁碟區的大小變更為 150GB。

命令：

```
aws ec2 modify-volume --size 150 --volume-id vol-1234567890abcdef0
```

輸出：

```
{
    "VolumeModification": {
        "TargetSize": 150,
        "TargetVolumeType": "io1",
        "ModificationState": "modifying",
        "VolumeId": " vol-1234567890abcdef0",
        "TargetIops": 100,
        "StartTime": "2019-05-17T11:27:19.000Z",
        "Progress": 0,
        "OriginalVolumeType": "io1",
        "OriginalIops": 100,
        "OriginalSize": 100
    }
}
```

範例 2：透過變更磁碟區類型、大小和 IOPS 值來修改磁碟區

下列modify-volume範例會將磁碟區類型變更為佈建 IOPS SSD、將目標 IOPS 速率設定為 10000，並將磁碟區大小設定為 350GB。

```
aws ec2 modify-volume \
    --volume-type io1 \
    --iops 10000 \
    --size 350 \
    --volume-id vol-1234567890abcdef0
```

輸出：

```
{
    "VolumeModification": {
        "TargetSize": 350,
        "TargetVolumeType": "io1",
        "ModificationState": "modifying",
        "VolumeId": "vol-0721c1a9d08c93bf6",
        "TargetIops": 10000,
        "StartTime": "2019-05-17T11:38:57.000Z",
        "Progress": 0,
        "OriginalVolumeType": "gp2",
        "OriginalIops": 150,
        "OriginalSize": 50
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVolume。

modify-vpc-attribute

以下程式碼範例顯示如何使用 modify-vpc-attribute。

AWS CLI

修改 enableDnsSupport 屬性

此範例會修改 enableDnsSupport 屬性。此屬性指出是否已啟用 VPC 的 DNS 解析。如果此屬性為 true，Amazon DNS 伺服器會將您的執行個體的 DNS 主機名稱解析為對應的 IP 地址；否則將不會進行解析。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 modify-vpc-attribute --vpc-id vpc-a01106c2 --enable-dns-support "{\"Value
\":false}"
```

修改 enableDnsHostnames 屬性

此範例會修改 enableDnsHostnames 屬性。此屬性指出在 VPC 中啟動的執行個體是否取得 DNS 主機名稱。如果此屬性為 true，該 VPC 中的執行個體會取得 DNS 主機名稱；否則將不會取得。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 modify-vpc-attribute --vpc-id vpc-a01106c2 --enable-dns-hostnames "{\"Value
\":false}"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVpcAttribute。

modify-vpc-endpoint-connection-notification

以下程式碼範例顯示如何使用 modify-vpc-endpoint-connection-notification。

AWS CLI

修改端點連線通知

此範例會變更指定端點連線通知的 SNS 主題。

命令：

```
aws ec2 modify-vpc-endpoint-connection-notification --connection-notification-
id vpce-nfn-008776de7e03f5abc --connection-events Accept Reject --connection-
notification-arn arn:aws:sns:us-east-2:123456789012:mytopic
```

輸出：

```
{
   "ReturnValue": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVpcEndpointConnectionNotification。

modify-vpc-endpoint-service-configuration

以下程式碼範例顯示如何使用 modify-vpc-endpoint-service-configuration。

AWS CLI

修改端點服務組態

此範例會變更指定端點服務的接受要求。

命令：

```
aws ec2 modify-vpc-endpoint-service-configuration --service-id vpce-
svc-09222513e6e77dc86 --no-acceptance-required
```

輸出：

```
{
   "ReturnValue": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVpcEndpointServiceConfiguration。

modify-vpc-endpoint-service-payer-responsibility

以下程式碼範例顯示如何使用 modify-vpc-endpoint-service-payer-responsibility。

AWS CLI

修改付款人責任

下列modify-vpc-endpoint-service-payer-responsibility範例會修改指定端點服務的付款人責任。

```
aws ec2 modify-vpc-endpoint-service-payer-responsibility \
    --service-id vpce-svc-071afff70666e61e0 \
    --payer-responsibility ServiceOwner
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVpcEndpointServicePayerResponsibility。

modify-vpc-endpoint-service-permissions

以下程式碼範例顯示如何使用 modify-vpc-endpoint-service-permissions。

AWS CLI

修改端點服務許可

此範例會新增 AWS 帳戶連線至指定端點服務的許可。

命令：

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-svc-03d5ebb7d9579a2b3 --add-allowed-principals '["arn:aws:iam::123456789012:root"]'
```

輸出：

```
{
    "ReturnValue": true
}
```

此範例會新增特定 IAM 使用者 (admin) 連線至指定端點服務的許可。

命令：

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-svc-03d5ebb7d9579a2b3 --add-allowed-principals '["arn:aws:iam::123456789012:user/admin"]'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVpcEndpointServicePermissions。

modify-vpc-endpoint

以下程式碼範例顯示如何使用 modify-vpc-endpoint。

AWS CLI

修改閘道端點

此範例vpce-1a2b3c4d透過將路由表rtb-aaa222bb與端點建立關聯，並重設政策文件，來修改閘道端點。

命令：

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-1a2b3c4d --add-route-table-
ids rtb-aaa222bb --reset-policy
```

輸出：

```
{
  "Return": true
}
```

修改界面端點

此範例vpce-0fe5b17a0707d6fa5透過將子網路subnet-d6fcaa8d新增至端點來修改介面端點。

命令：

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-0fe5b17a0707d6fa5 --add-subnet-
id subnet-d6fcaa8d
```

輸出：

```
{
  "Return": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVpcEndpoint。

modify-vpc-peering-connection-options

以下程式碼範例顯示如何使用 modify-vpc-peering-connection-options。

AWS CLI

從本機 ClassicLink 連線透過 VPC 互連連線啟用通訊

在此範例中，對於對等連線 pcx-aaaabbb，請求者 VPC 的擁有者會修改 VPC 對等連線選項，讓本機 ClassicLink 連線能夠與對等 VPC 通訊。

命令：

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-
connection-id pcx-aaaabbbb --requester-peering-connection-
options AllowEgressFromLocalClassicLinkToRemoteVpc=true
```

輸出：

```
{
  "RequesterPeeringConnectionOptions": {
      "AllowEgressFromLocalClassicLinkToRemoteVpc": true
  }
}
```

啟用從本機 VPC 到遠端 ClassicLink 連線的 VPC 對等互連通訊

在此範例中，接受者 VPC 的擁有者會修改 VPC 互連連線選項，讓本機 VPC 與對等 VPC 中的 ClassicLink 連線通訊。

命令：

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-
connection-id pcx-aaaabbbb --accepter-peering-connection-
options AllowEgressFromLocalVpcToRemoteClassicLink=true
```

輸出：

```
{
  "AccepterPeeringConnectionOptions": {
    "AllowEgressFromLocalVpcToRemoteClassicLink": true
  }
}
```

啟用 VPC 對等互連的 DNS 解析支援

在此範例中，申請者 VPC 的擁有者會修改 的 VPC 互連連線選項pcx-aaaabbbb，讓本機 VPC 在從對等 VPC 中的執行個體查詢時，將公有 DNS 主機名稱解析為私有 IP 地址。

命令：

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-connection-id pcx-
aaaabbbb --requester-peering-connection-options AllowDnsResolutionFromRemoteVpc=true
```

輸出：

```
{
  "RequesterPeeringConnectionOptions": {
      "AllowDnsResolutionFromRemoteVpc": true
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVpcPeeringConnectionOptions。

modify-vpc-tenancy

以下程式碼範例顯示如何使用 modify-vpc-tenancy。

AWS CLI

修改 VPC 的租用

此範例會將 VPC 的租用修改vpc-1a2b3c4d為 default。

命令：

```
aws ec2 modify-vpc-tenancy --vpc-id vpc-1a2b3c4d --instance-tenancy default
```

輸出：

```
{
  "Return": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVpcTenancy。

modify-vpn-connection-options

以下程式碼範例顯示如何使用 modify-vpn-connection-options。

AWS CLI

修改您的 VPN 連線選項

下列modify-vpn-connection-options範例會修改指定 VPN 連接之客戶閘道端的本機 IPv4 CIDR。

```
aws ec2 modify-vpn-connection-options \
    --vpn-connection-id vpn-1122334455aabbccd \
    --local-ipv4-network-cidr 10.0.0.0/16
```

輸出：

```
{
    "VpnConnections": [
        {
            "CustomerGatewayConfiguration": "...configuration information...",
            "CustomerGatewayId": "cgw-01234567abcde1234",
            "Category": "VPN",
            "State": "modifying",
            "Type": "ipsec.1",
            "VpnConnectionId": "vpn-1122334455aabbccd",
            "TransitGatewayId": "tgw-00112233445566aab",
            "Options": {
                "EnableAcceleration": false,
                "StaticRoutesOnly": true,
                "LocalIpv4NetworkCidr": "10.0.0.0/16",
                "RemoteIpv4NetworkCidr": "0.0.0.0/0",
                "TunnelInsideIpVersion": "ipv4"
            },
            "Routes": [],
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "CanadaVPN"
                }
            ],
            "VgwTelemetry": [
                {
                    "AcceptedRouteCount": 0,
                    "LastStatusChange": "2020-07-29T10:35:11.000Z",
                    "OutsideIpAddress": "203.0.113.3",
                    "Status": "DOWN",
                    "StatusMessage": ""
                },
                {
                    "AcceptedRouteCount": 0,
```

```
                    "LastStatusChange": "2020-09-02T09:09:33.000Z",
                    "OutsideIpAddress": "203.0.113.5",
                    "Status": "UP",
                    "StatusMessage": ""
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱Site-to-Site使用者指南》中的修改站台對站台 VPN 連接選項。 AWS Site-to-Site

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVpnConnectionOptions。

modify-vpn-connection

以下程式碼範例顯示如何使用 modify-vpn-connection。

AWS CLI

修改 VPN 連線

下列modify-vpn-connection範例會變更 VPN vpn-12345678901234567連線至虛擬私有閘道 的目標閘道vgw-11223344556677889：

```
aws ec2 modify-vpn-connection \
    --vpn-connection-id vpn-12345678901234567 \
    --vpn-gateway-id vgw-11223344556677889
```

輸出：

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": "...configuration information...",
        "CustomerGatewayId": "cgw-aabbccddee1122334",
        "Category": "VPN",
        "State": "modifying",
        "Type": "ipsec.1",
        "VpnConnectionId": "vpn-12345678901234567",
        "VpnGatewayId": "vgw-11223344556677889",
        "Options": {
            "StaticRoutesOnly": false
```

```
        },
        "VgwTelemetry": [
            {
                "AcceptedRouteCount": 0,
                "LastStatusChange": "2019-07-17T07:34:00.000Z",
                "OutsideIpAddress": "18.210.3.222",
                "Status": "DOWN",
                "StatusMessage": "IPSEC IS DOWN"
            },
            {
                "AcceptedRouteCount": 0,
                "LastStatusChange": "2019-07-20T21:20:16.000Z",
                "OutsideIpAddress": "34.193.129.33",
                "Status": "DOWN",
                "StatusMessage": "IPSEC IS DOWN"
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVpnConnection。

modify-vpn-tunnel-certificate

以下程式碼範例顯示如何使用 modify-vpn-tunnel-certificate。

AWS CLI

輪換 VPN 通道憑證

下列modify-vpn-tunnel-certificate範例會輪換 VPN 連線指定通道的憑證

```
aws ec2 modify-vpn-tunnel-certificate \
    --vpn-tunnel-outside-ip-address 203.0.113.17 \
    --vpn-connection-id vpn-12345678901234567
```

輸出：

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": ...configuration information...,
        "CustomerGatewayId": "cgw-aabbccddee1122334",
```

```
        "Category": "VPN",
        "State": "modifying",
        "Type": "ipsec.1",
        "VpnConnectionId": "vpn-12345678901234567",
        "VpnGatewayId": "vgw-11223344556677889",
        "Options": {
            "StaticRoutesOnly": false
        },
        "VgwTelemetry": [
            {
                "AcceptedRouteCount": 0,
                "LastStatusChange": "2019-09-11T17:27:14.000Z",
                "OutsideIpAddress": "203.0.113.17",
                "Status": "DOWN",
                "StatusMessage": "IPSEC IS DOWN",
                "CertificateArn": "arn:aws:acm:us-east-1:123456789101:certificate/
c544d8ce-20b8-4fff-98b0-example"
            },
            {
                "AcceptedRouteCount": 0,
                "LastStatusChange": "2019-09-11T17:26:47.000Z",
                "OutsideIpAddress": "203.0.114.18",
                "Status": "DOWN",
                "StatusMessage": "IPSEC IS DOWN",
                "CertificateArn": "arn:aws:acm:us-
east-1:123456789101:certificate/5ab64566-761b-4ad3-b259-example"
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVpnTunnelCertificate。

`modify-vpn-tunnel-options`

以下程式碼範例顯示如何使用 `modify-vpn-tunnel-options`。

AWS CLI

修改 VPN 連接的通道選項

下列`modify-vpn-tunnel-options`範例會更新指定通道和 VPN 連線允許的 Diffie-Hellman 群組。

```
aws ec2 modify-vpn-tunnel-options \
    --vpn-connection-id vpn-12345678901234567 \
    --vpn-tunnel-outside-ip-address 203.0.113.17 \
    --tunnel-options Phase1DHGroupNumbers=[{Value=14},{Value=15},{Value=16},
{Value=17},{Value=18}],Phase2DHGroupNumbers=[{Value=14},{Value=15},{Value=16},
{Value=17},{Value=18}]
```

輸出：

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": "...configuration information...",
        "CustomerGatewayId": "cgw-aabbccddee1122334",
        "Category": "VPN",
        "State": "available",
        "Type": "ipsec.1",
        "VpnConnectionId": "vpn-12345678901234567",
        "VpnGatewayId": "vgw-11223344556677889",
        "Options": {
            "StaticRoutesOnly": false,
            "TunnelOptions": [
                {
                    "OutsideIpAddress": "203.0.113.17",
                    "Phase1DHGroupNumbers": [
                        {
                            "Value": 14
                        },
                        {
                            "Value": 15
                        },
                        {
                            "Value": 16
                        },
                        {
                            "Value": 17
                        },
                        {
                            "Value": 18
                        }
                    ],
                    "Phase2DHGroupNumbers": [
                        {
                            "Value": 14
```

```
                        },
                        {
                            "Value": 15
                        },
                        {
                            "Value": 16
                        },
                        {
                            "Value": 17
                        },
                        {
                            "Value": 18
                        }
                    ]
                },
                {
                    "OutsideIpAddress": "203.0.114.19"
                }
            ]
        },
        "VgwTelemetry": [
            {
                "AcceptedRouteCount": 0,
                "LastStatusChange": "2019-09-10T21:56:54.000Z",
                "OutsideIpAddress": "203.0.113.17",
                "Status": "DOWN",
                "StatusMessage": "IPSEC IS DOWN"
            },
            {
                "AcceptedRouteCount": 0,
                "LastStatusChange": "2019-09-10T21:56:43.000Z",
                "OutsideIpAddress": "203.0.114.19",
                "Status": "DOWN",
                "StatusMessage": "IPSEC IS DOWN"
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyVpnTunnelOptions。

monitor-instances

以下程式碼範例顯示如何使用 monitor-instances。

AWS CLI

啟用執行個體的詳細監控

此範例命令會啟用指定執行個體的詳細監控。

命令：

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

輸出：

```
{
  "InstanceMonitorings": [
      {
          "InstanceId": "i-1234567890abcdef0",
          "Monitoring": {
              "State": "pending"
          }
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 MonitorInstances。

move-address-to-vpc

以下程式碼範例顯示如何使用 move-address-to-vpc。

AWS CLI

將地址移至 EC2-VPC

此範例會將彈性 IP 地址 54.123.4.56 移至 EC2-VPC 平台。

命令：

```
aws ec2 move-address-to-vpc --public-ip 54.123.4.56
```

輸出：

```
{
  "Status": "MoveInProgress"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 MoveAddressToVpc。

move-byoip-cidr-to-ipam

以下程式碼範例顯示如何使用 move-byoip-cidr-to-ipam。

AWS CLI

將 BYOIP CIDR 傳輸至 IPAM

下列move-byoip-cidr-to-ipam範例會將 BYOIP CIDR 傳輸至 IPAM。

(Linux)：

```
aws ec2 move-byoip-cidr-to-ipam \
    --region us-west-2 \
    --ipam-pool-id ipam-pool-0a03d430ca3f5c035 \
    --ipam-pool-owner 111111111111 \
    --cidr 130.137.249.0/24
```

(Windows)：

```
aws ec2 move-byoip-cidr-to-ipam ^
    --region us-west-2 ^
    --ipam-pool-id ipam-pool-0a03d430ca3f5c035 ^
    --ipam-pool-owner 111111111111 ^
    --cidr 130.137.249.0/24
```

輸出：

```
{
    "ByoipCidr": {
        "Cidr": "130.137.249.0/24",
        "State": "pending-transfer"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的教學課程：將現有的 BYOIP IPv4 CIDR 轉移至 IPAM。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 MoveByoipCidrToIpam。

network-insights-access-scope

以下程式碼範例顯示如何使用 network-insights-access-scope。

AWS CLI

建立 Network Insights 存取範圍

下列create-network-insights-access-scope範例會在您的帳戶中建立網路洞見存取範圍 AWS 。

```
aws ec2 create-network-insights-access-scope \
    --cli-input-json file://access-scope-file.json
```

access-scope-file.json 的內容：

```
{
    {
        "MatchPaths": [
            {
                "Source": {
                    "ResourceStatement": {
                         "Resources": [
                            "vpc-abcd12e3"
                        ]
                    }
                }
            }
        ],
        "ExcludePaths": [
            {
                "Source": {
                    "ResourceStatement": {
                        "ResourceTypes": [
                            "AWS::EC2::InternetGateway"
                        ]
                    }
                }
```

```
            }
        ]
    }
}
```

輸出：

```
{
    "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789111"
    }{
    "NetworkInsightsAccessScope": {
        "NetworkInsightsAccessScopeId": "nis-123456789222",
        "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-1:123456789222:network-insights-access-scope/nis-123456789222",
        "CreatedDate": "2022-01-25T19:20:28.796000+00:00",
        "UpdatedDate": "2022-01-25T19:20:28.797000+00:00"
    },
    "NetworkInsightsAccessScopeContent": {
        "NetworkInsightsAccessScopeId": "nis-04c0c0fbca737c404",
        "MatchPaths": [
            {
                "Source": {
                    "ResourceStatement": {
                        "Resources": [
                            "vpc-abcd12e3"
                        ]
                    }
                }
            }
        ],
        "ExcludePaths": [
            {
                "Source": {
                    "ResourceStatement": {
                        "ResourceTypes": [
                            "AWS::EC2::InternetGateway"
                        ]
                    }
                }
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Network Access Analyzer 指南》中的使用 AWS CLI 開始使用 Network Access Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 NetworkInsightsAccessScope。

`provision-byoip-cidr`

以下程式碼範例顯示如何使用 `provision-byoip-cidr`。

AWS CLI

佈建地址範圍

下列`provision-byoip-cidr`範例佈建用於 的公有 IP 地址範圍 AWS。

```
aws ec2 provision-byoip-cidr \
    --cidr 203.0.113.25/24 \
    --cidr-authorization-context Message="$text_message",Signature="$signed_message"
```

輸出：

```
{
    "ByoipCidr": {
        "Cidr": "203.0.113.25/24",
        "State": "pending-provision"
    }
}
```

如需為授權內容建立訊息字串的詳細資訊，請參閱《Amazon EC2 使用者指南》中的使用您自己的 IP 地址。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ProvisionByoipCidr。

`provision-ipam-pool-cidr`

以下程式碼範例顯示如何使用 `provision-ipam-pool-cidr`。

AWS CLI

將 CIDR 佈建至 IPAM 集區

下列`provision-ipam-pool-cidr`範例會將 CIDR 佈建至 IPAM 集區。

(Linux)：

```
aws ec2 provision-ipam-pool-cidr \
    --ipam-pool-id ipam-pool-0533048da7d823723 \
    --cidr 10.0.0.0/24
```

(Windows)：

```
aws ec2 provision-ipam-pool-cidr ^
    --ipam-pool-id ipam-pool-0533048da7d823723 ^
    --cidr 10.0.0.0/24
```

輸出：

```
{
    "IpamPoolCidr": {
        "Cidr": "10.0.0.0/24",
        "State": "pending-provision"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的將 CIDRs 佈建至集區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ProvisionIpamPoolCidr。

purchase-host-reservation

以下程式碼範例顯示如何使用 purchase-host-reservation。

AWS CLI

購買專用主機預留

此範例會為您帳戶中指定的專用主機購買指定的專用主機保留方案。

命令：

```
aws ec2 purchase-host-reservation --offering-id hro-03f707bf363b6b324 --host-id-
set h-013abcd2a00cbd123
```

輸出：

```
{
  "TotalHourlyPrice": "1.499",
  "Purchase": [
      {
          "HourlyPrice": "1.499",
          "InstanceFamily": "m4",
          "PaymentOption": "NoUpfront",
          "HostIdSet": [
              "h-013abcd2a00cbd123"
          ],
          "HostReservationId": "hr-0d418a3a4ffc669ae",
          "UpfrontPrice": "0.000",
          "Duration": 31536000
      }
  ],
  "TotalUpfrontPrice": "0.000"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PurchaseHostReservation。

purchase-reserved-instances-offering

以下程式碼範例顯示如何使用 purchase-reserved-instances-offering。

AWS CLI

購買預留執行個體方案

此範例命令說明購買預留執行個體方案，並指定方案 ID 和執行個體計數。

命令：

```
aws ec2 purchase-reserved-instances-offering --reserved-instances-offering-
id ec06327e-dd07-46ee-9398-75b5fexample --instance-count 3
```

輸出：

```
{
  "ReservedInstancesId": "af9f760e-6f91-4559-85f7-4980eexample"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PurchaseReservedInstancesOffering。

purchase-scheduled-instances

以下程式碼範例顯示如何使用 purchase-scheduled-instances。

AWS CLI

購買排程執行個體

此範例會購買排程執行個體。

命令：

```
aws ec2 purchase-scheduled-instances --purchase-requests file://purchase-request.json
```

Purchase-request.json：

```
[
    {
        "PurchaseToken": "eyJ2IjoiMSIsInMiOjEsImMiOi...",
        "InstanceCount": 1
    }
]
```

輸出：

```
{
  "ScheduledInstanceSet": [
      {
          "AvailabilityZone": "us-west-2b",
          "ScheduledInstanceId": "sci-1234-1234-1234-1234-123456789012",
          "HourlyPrice": "0.095",
          "CreateDate": "2016-01-25T21:43:38.612Z",
          "Recurrence": {
              "OccurrenceDaySet": [
                  1
              ],
              "Interval": 1,
              "Frequency": "Weekly",
              "OccurrenceRelativeToEnd": false,
              "OccurrenceUnit": ""
          },
          "Platform": "Linux/UNIX",
```

```
            "TermEndDate": "2017-01-31T09:00:00Z",
            "InstanceCount": 1,
            "SlotDurationInHours": 32,
            "TermStartDate": "2016-01-31T09:00:00Z",
            "NetworkPlatform": "EC2-VPC",
            "TotalScheduledInstanceHours": 1696,
            "NextSlotStartTime": "2016-01-31T09:00:00Z",
            "InstanceType": "c4.large"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PurchaseScheduledInstances。

reboot-instances

以下程式碼範例顯示如何使用 reboot-instances。

AWS CLI

重新啟動 Amazon EC2 執行個體

此範例會重新啟動指定執行個體。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 reboot-instances --instance-ids i-1234567890abcdef5
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的「重新啟動您的執行個體」。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 RebootInstances。

register-image

以下程式碼範例顯示如何使用 register-image。

AWS CLI

範例 1：使用資訊清單檔案註冊 AMI

下列register-image範例使用 Amazon S3 中指定的資訊清單檔案註冊 AMI。

```
aws ec2 register-image \
    --name my-image \
    --image-location amzn-s3-demo-bucket/myimage/image.manifest.xml
```

輸出：

```
{
    "ImageId": "ami-1234567890EXAMPLE"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Amazon Machine Images (AMI)。

範例 2：使用根裝置的快照註冊 AMI

下列`register-image`範例使用 EBS 根磁碟區的指定快照，將 AMI 註冊為裝置 `/dev/xvda`。區塊型設備映射也包含空的 100 GiB EBS 磁碟區做為裝置 `/dev/xvdf`。

```
aws ec2 register-image \
    --name my-image \
    --root-device-name /dev/xvda \
    --block-device-mappings DeviceName=/dev/
xvda,Ebs={SnapshotId=snap-0db2cf683925d191f} DeviceName=/dev/
xvdf,Ebs={VolumeSize=100}
```

輸出：

```
{
    "ImageId": "ami-1a2b3c4d5eEXAMPLE"
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Amazon Machine Images (AMI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterImage。

register-instance-event-notification-attributes

以下程式碼範例顯示如何使用 `register-instance-event-notification-attributes`。

AWS CLI

範例 1：在事件通知中包含所有標籤

下列register-instance-event-notification-attributes範例包含事件通知中的所有標籤。

```
aws ec2 register-instance-event-notification-attributes \
    --instance-tag-attribute IncludeAllTagsOfInstance=true
```

輸出：

```
{
    "InstanceTagAttribute": {
        "InstanceTagKeys": [],
        "IncludeAllTagsOfInstance": true
    }
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的執行個體的排程事件。

範例 2：在事件通知中包含特定標籤

下列register-instance-event-notification-attributes範例會在事件通知中包含指定的標籤。如果 IncludeAllTagsOfInstance為 ，則無法指定標籤true。

```
aws ec2 register-instance-event-notification-attributes \
    --instance-tag-attribute InstanceTagKeys="tag-key1","tag-key2"
```

輸出：

```
{
    "InstanceTagAttribute": {
        "InstanceTagKeys": [
            "tag-key1",
            "tag-key2"
        ],
        "IncludeAllTagsOfInstance": false
    }
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的執行個體的排程事件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterInstanceEventNotificationAttributes。

register-transit-gateway-multicase-group-sources

以下程式碼範例顯示如何使用 register-transit-gateway-multicase-group-sources。

AWS CLI

向傳輸閘道多點傳送群組註冊來源。

下列register-transit-gateway-multicast-group-sources範例會向多點傳送群組註冊指定的網路介面群組來源。

```
aws ec2 register-transit-gateway-multicast-group-sources \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \
    --group-ip-address 224.0.1.0 \
    --network-interface-ids eni-07f290fc3c090cbae
```

輸出：

```
{
    "RegisteredMulticastGroupSources": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
        "RegisteredNetworkInterfaceIds": [
            "eni-07f290fc3c090cbae"
        ],
        "GroupIpAddress": "224.0.1.0"
    }
}
```

如需詳細資訊，請參閱 AWS Transit Gateways 使用者指南中的向多點傳送群組註冊來源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterTransitGatewayMulticaseGroupSources。

register-transit-gateway-multicast-group-members

以下程式碼範例顯示如何使用 register-transit-gateway-multicast-group-members。

AWS CLI

檢視傳輸閘道多點傳送網域關聯的相關資訊

下列register-transit-gateway-multicast-group-members範例會傳回指定多點傳送網域的關聯。

```
aws ec2 register-transit-gateway-multicast-group-members \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \
    --group-ip-address 224.0.1.0 \
    --network-interface-ids eni-0e246d32695012e81
```

輸出：

```
{
    "RegisteredMulticastGroupMembers": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
        "RegisteredNetworkInterfaceIds": [
            "eni-0e246d32695012e81"
        ],
        "GroupIpAddress": "224.0.1.0"
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 使用者指南中的多點傳送網域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterTransitGatewayMulticastGroupMembers。

register-transit-gateway-multicast-group-sources

以下程式碼範例顯示如何使用 register-transit-gateway-multicast-group-sources。

AWS CLI

向傳輸閘道多點傳送群組註冊來源。

下列register-transit-gateway-multicast-group-sources範例會向多點傳送群組註冊指定的網路介面群組來源。

```
aws ec2 register-transit-gateway-multicast-group-sources \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \
    --group-ip-address 224.0.1.0 \
    --network-interface-ids eni-07f290fc3c090cbae
```

輸出：

```
{
    "RegisteredMulticastGroupSources": {
```

```
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
        "RegisteredNetworkInterfaceIds": [
            "eni-07f290fc3c090cbae"
        ],
        "GroupIpAddress": "224.0.1.0"
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的多點傳送網域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterTransitGatewayMulticastGroupSources。

`reject-transit-gateway-peering-attachment`

以下程式碼範例顯示如何使用 `reject-transit-gateway-peering-attachment`。

AWS CLI

拒絕傳輸閘道對等互連附件

下列`reject-transit-gateway-peering-attachment`範例會拒絕指定的傳輸閘道對等連接請求。`--region` 參數指定接受者傳輸閘道所在的區域。

```
aws ec2 reject-transit-gateway-peering-attachment \
    --transit-gateway-attachment-id tgw-attach-4455667788aabbccd \
    --region us-east-2
```

輸出：

```
{
    "TransitGatewayPeeringAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
        "RequesterTgwInfo": {
            "TransitGatewayId": "tgw-123abc05e04123abc",
            "OwnerId": "123456789012",
            "Region": "us-west-2"
        },
        "AccepterTgwInfo": {
            "TransitGatewayId": "tgw-11223344aabbcc112",
            "OwnerId": "123456789012",
            "Region": "us-east-2"
        },
```

```
        "State": "rejecting",
        "CreationTime": "2019-12-09T11:50:31.000Z"
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的 Transit Gateway 對等連接。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RejectTransitGatewayPeeringAttachment。

`reject-transit-gateway-vpc-attachment`

以下程式碼範例顯示如何使用 `reject-transit-gateway-vpc-attachment`。

AWS CLI

拒絕傳輸閘道 VPC 連接

下列`reject-transit-gateway-vpc-attachment`範例會拒絕指定的傳輸閘道 VPC 連接。

```
aws ec2 reject-transit-gateway-vpc-attachment \
    --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE
```

輸出：

```
{
    "TransitGatewayVpcAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-0a34fe6b4fEXAMPLE",
        "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
        "VpcId": "vpc-07e8ffd50fEXAMPLE",
        "VpcOwnerId": "111122223333",
        "State": "pending",
        "SubnetIds": [
            "subnet-0752213d59EXAMPLE"
        ],
        "CreationTime": "2019-07-10T17:33:46.000Z",
        "Options": {
            "DnsSupport": "enable",
            "Ipv6Support": "disable"
        }
    }
}
```

如需詳細資訊，請參閱《傳輸閘道指南》中的傳輸閘道連接至 VPC。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RejectTransitGatewayVpcAttachment。

`reject-transit-gateway-vpc-attachments`

以下程式碼範例顯示如何使用 `reject-transit-gateway-vpc-attachments`。

AWS CLI

拒絕傳輸閘道 VPC 連接

下列`reject-transit-gateway-vpc-attachment`範例會拒絕指定的傳輸閘道 VPC 連接。

```
aws ec2 reject-transit-gateway-vpc-attachment \
    --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE
```

輸出：

```
{
    "TransitGatewayVpcAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-0a34fe6b4fEXAMPLE",
        "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
        "VpcId": "vpc-07e8ffd50fEXAMPLE",
        "VpcOwnerId": "111122223333",
        "State": "pending",
        "SubnetIds": [
            "subnet-0752213d59EXAMPLE"
        ],
        "CreationTime": "2019-07-10T17:33:46.000Z",
        "Options": {
            "DnsSupport": "enable",
            "Ipv6Support": "disable"
        }
    }
}
```

如需詳細資訊，請參閱《傳輸閘道指南》中的傳輸閘道連接至 VPC。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RejectTransitGatewayVpcAttachments。

reject-vpc-endpoint-connections

以下程式碼範例顯示如何使用 reject-vpc-endpoint-connections。

AWS CLI

拒絕介面端點連線請求

此範例會拒絕指定端點服務的指定端點連線請求。

命令：

```
aws ec2 reject-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --
vpc-endpoint-ids vpce-0c1308d7312217abc
```

輸出：

```
{
  "Unsuccessful": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RejectVpcEndpointConnections。

reject-vpc-peering-connection

以下程式碼範例顯示如何使用 reject-vpc-peering-connection。

AWS CLI

拒絕 VPC 對等互連

此範例會拒絕指定的 VPC 互連連線請求。

命令：

```
aws ec2 reject-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

輸出：

```
{
    "Return": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RejectVpcPeeringConnection。

release-address

以下程式碼範例顯示如何使用 release-address。

AWS CLI

釋出適用於 EC2-Classic 的彈性 IP 地址

此範例會釋出彈性 IP 地址，以便與 EC2-Classic 中的執行個體搭配使用。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 release-address --public-ip 198.51.100.0
```

釋出適用於 EC2-VPC 的彈性 IP 地址

此範例會釋出彈性 IP 地址，以便與 VPC 中的執行個體搭配使用。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 release-address --allocation-id eipalloc-64d5890a
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ReleaseAddress。

release-hosts

以下程式碼範例顯示如何使用 release-hosts。

AWS CLI

從您的帳戶釋出專用主機

從您的帳戶釋出專用主機。在釋出主機之前，必須停止或終止主機上的執行個體。

命令：

```
aws ec2 release-hosts --host-id=h-0029d6e3cacf1b3da
```

輸出：

```
{
    "Successful":  [
        "h-0029d6e3cacf1b3da"
         ],
  "Unsuccessful": []

 }
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReleaseHosts。

release-ipam-pool-allocation

以下程式碼範例顯示如何使用 release-ipam-pool-allocation。

AWS CLI

釋出 IPAM 集區配置

在此範例中，您是 IPAM 委派管理員，嘗試刪除 IPAM 集區，但收到在集區有配置時無法刪除集區的錯誤。您正在使用此命令來釋出集區配置。

注意下列事項：

您只能將此命令用於自訂配置。若要移除資源的配置而不刪除資源，請使用 modify-ipam-resource-cidr 將其受監控狀態設為 false。若要完成此請求，您需要 IPAM 集區 ID，您可以使用 describe-ipam-pools 取得此 ID。您還需要配置 ID，您可以使用 get-ipam-pool-allocations 取得。如果您不想逐一移除配置，您可以在刪除 IPAM 集區--cascade option時，使用 在刪除集區之前自動釋放集區中的任何配置。在執行此命令之前，有一些先決條件。如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的發行配置。--region執行此命令的 必須是配置所在 IPAM 集區的地區設定。

下列release-ipam-pool-allocation範例會發行 IPAM 集區配置。

```
aws ec2 release-ipam-pool-allocation \
    --ipam-pool-id ipam-pool-07bdd12d7c94e4693 \
    --cidr 10.0.0.0/23 \
    --ipam-pool-allocation-id ipam-pool-alloc-0e66a1f730da54791b99465b79e7d1e89 \
    --region us-west-1
```

輸出：

```
{
    "Success": true
}
```

發佈配置後，您可能想要執行 delete-ipam-pool。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReleaseIpamPoolAllocation。

replace-iam-instance-profile-association

以下程式碼範例顯示如何使用 replace-iam-instance-profile-association。

AWS CLI

取代執行個體的 IAM 執行個體設定檔

此範例會取代以與名為 AdminRole 的 IAM 執行個體設定檔之關聯 iip-assoc-060bae234aac2e7fa 所表示的 IAM 執行個體設定檔。

```
aws ec2 replace-iam-instance-profile-association \
    --iam-instance-profile Name=AdminRole \
    --association-id iip-assoc-060bae234aac2e7fa
```

輸出：

```
{
    "IamInstanceProfileAssociation": {
        "InstanceId": "i-087711ddaf98f9489",
        "State": "associating",
        "AssociationId": "iip-assoc-0b215292fab192820",
        "IamInstanceProfile": {
            "Id": "AIPAJLNLDX3AMYZNWYYAY",
            "Arn": "arn:aws:iam::123456789012:instance-profile/AdminRole"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ReplaceIamInstanceProfileAssociation。

replace-network-acl-association

以下程式碼範例顯示如何使用 replace-network-acl-association。

AWS CLI

取代與子網路相關聯的網路 ACL

此範例會將指定的網路 ACL 與指定網路 ACL 關聯的子網路建立關聯。

命令：

```
aws ec2 replace-network-acl-association --association-id aclassoc-e5b95c8c --
network-acl-id acl-5fb85d36
```

輸出：

```
{
    "NewAssociationId": "aclassoc-3999875b"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReplaceNetworkAclAssociation。

replace-network-acl-entry

以下程式碼範例顯示如何使用 replace-network-acl-entry。

AWS CLI

若要取代網路 ACL 項目

此範例會取代指定網路 ACL 的項目。新規則 100 允許將 UDP 連接埠 53 (DNS) 上 203.0.113.12/24 的流量傳入任何相關聯的子網路。

命令：

```
aws ec2 replace-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-
number 100 --protocol udp --port-range From=53,To=53 --cidr-block 203.0.113.12/24 --
rule-action allow
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReplaceNetworkAclEntry。

replace-route-table-association

以下程式碼範例顯示如何使用 replace-route-table-association。

AWS CLI

取代與子網路相關聯的路由表

此範例會將指定的路由表與指定路由表關聯的子網路建立關聯。

命令：

```
aws ec2 replace-route-table-association --association-id rtbassoc-781d0d1a --route-table-id rtb-22574640
```

輸出：

```
{
    "NewAssociationId": "rtbassoc-3a1f0f58"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReplaceRouteTableAssociation。

replace-route

以下程式碼範例顯示如何使用 replace-route。

AWS CLI

若要取代路由

此範例會取代指定路由表中指定的路由。新路由符合指定的 CIDR，並將流量傳送至指定的虛擬私有閘道。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 replace-route --route-table-id rtb-22574640 --destination-cidr-block 10.0.0.0/16 --gateway-id vgw-9a4cacf3
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReplaceRoute。

replace-transit-gateway-route

以下程式碼範例顯示如何使用 replace-transit-gateway-route。

AWS CLI

取代指定傳輸閘道路由表中的指定路由

下列replace-transit-gateway-route範例會取代指定傳輸閘道路由表中的路由。

```
aws ec2 replace-transit-gateway-route \
    --destination-cidr-block 10.0.2.0/24 \
    --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE \
    --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE
```

輸出：

```
{
    "Route": {
        "DestinationCidrBlock": "10.0.2.0/24",
        "TransitGatewayAttachments": [
            {
                "ResourceId": "vpc-4EXAMPLE",
                "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
                "ResourceType": "vpc"
            }
        ],
        "Type": "static",
        "State": "active"
    }
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的 Transit Gateway 路由表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReplaceTransitGatewayRoute。

report-instance-status

以下程式碼範例顯示如何使用 report-instance-status。

AWS CLI

報告執行個體的狀態意見回饋

此範例命令會報告指定執行個體的狀態意見回饋。

命令：

```
aws ec2 report-instance-status --instances i-1234567890abcdef0 --status impaired --
reason-codes unresponsive
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReportInstanceStatus。

request-spot-fleet

以下程式碼範例顯示如何使用 request-spot-fleet。

AWS CLI

以最低價格請求子網路中的 Spot 機群

此範例命令會建立具有兩個啟動規格的 Spot 機群請求，這些啟動規格只會因子網路而有所不同。Spot 機群會以最低價格啟動指定子網路中的執行個體。如果執行個體是在預設 VPC 中啟動，它們預設會收到公有 IP 地址。如果執行個體是在非預設的 VPC 中啟動，則其預設不會接收公有 IP 地址。

請注意，您無法在 Spot 機群請求中指定與相同可用區域不同的子網路。

命令：

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

Config.json：

```
{
  "SpotPrice": "0.04",
  "TargetCapacity": 2,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "KeyName": "my-key-pair",
          "SecurityGroups": [
              {
```

```
                "GroupId": "sg-1a2b3c4d"
            }
        ],
        "InstanceType": "m3.medium",
        "SubnetId": "subnet-1a2b3c4d, subnet-3c4d5e6f",
        "IamInstanceProfile": {
            "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
        }
    }
  ]
}
```

輸出：

```
{
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

以最低價格請求可用區域中的 Spot 機群

此範例命令會建立具有兩個啟動規格的 Spot 機群請求，這些啟動規格只會因可用區域而有所不同。Spot 機群會以最低價格在指定的可用區域中啟動執行個體。如果您的帳戶僅支援 EC2-VPC，Amazon EC2 會在可用區域的預設子網路中啟動 Spot 執行個體。如果您的帳戶支援 EC2-Classic，Amazon EC2 會在可用區域中啟動 EC2-Classic 中的執行個體。

命令：

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

Config.json：

```
{
  "SpotPrice": "0.04",
  "TargetCapacity": 2,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "KeyName": "my-key-pair",
          "SecurityGroups": [
              {
```

```
                "GroupId": "sg-1a2b3c4d"
            }
        ],
        "InstanceType": "m3.medium",
        "Placement": {
            "AvailabilityZone": "us-west-2a, us-west-2b"
        },
        "IamInstanceProfile": {
            "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
        }
    }
  ]
}
```

在子網路中啟動 Spot 執行個體並為其指派公有 IP 地址

此範例命令會將公有地址指派給在非預設 VPC 中啟動的執行個體。請注意，當您指定網路界面時，您必須使用網路界面包含子網路 ID 和安全群組 ID。

命令：

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

Config.json：

```
{
  "SpotPrice": "0.04",
  "TargetCapacity": 2,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
        "ImageId": "ami-1a2b3c4d",
        "KeyName": "my-key-pair",
        "InstanceType": "m3.medium",
        "NetworkInterfaces": [
            {
                "DeviceIndex": 0,
                "SubnetId": "subnet-1a2b3c4d",
                "Groups": [ "sg-1a2b3c4d" ],
                "AssociatePublicIpAddress": true
            }
        ],
        "IamInstanceProfile": {
```

```
            "Arn": "arn:aws:iam::880185128111:instance-profile/my-iam-role"
        }
    }
  ]
}
```

使用多樣化配置策略請求 Spot 機群

此範例命令會建立 Spot 機群請求，使用多樣化的配置策略啟動 30 個執行個體。啟動規格因執行個體類型而異。Spot 機群會將執行個體分散到啟動規格，因此每種類型都有 10 個執行個體。

命令：

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

Config.json：

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 30,
  "AllocationStrategy": "diversified",
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c4.2xlarge",
          "SubnetId": "subnet-1a2b3c4d"
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "m3.2xlarge",
          "SubnetId": "subnet-1a2b3c4d"
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "r3.2xlarge",
          "SubnetId": "subnet-1a2b3c4d"
      }
  ]
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的 Spot Fleet 請求。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RequestSpotFleet。

`request-spot-instances`

以下程式碼範例顯示如何使用 `request-spot-instances`。

AWS CLI

請求 Spot 執行個體

此範例命令會為指定可用區域中的五個執行個體建立一次性 Spot 執行個體請求。如果您的帳戶僅支援 EC2-VPC，Amazon EC2 會在指定可用區域的預設子網路中啟動執行個體。如果您的帳戶支援 EC2-Classic，Amazon EC2 會在指定的可用區域中啟動 EC2-Classic 中的執行個體。

命令：

```
aws ec2 request-spot-instances --spot-price "0.03" --instance-count 5 --type "one-time" --launch-specification file://specification.json
```

Specification.json：

```
{
  "ImageId": "ami-1a2b3c4d",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "Placement": {
    "AvailabilityZone": "us-west-2a"
  },
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

輸出：

```
{
  "SpotInstanceRequests": [
      {
          "Status": {
              "UpdateTime": "2014-03-25T20:54:21.000Z",
              "Code": "pending-evaluation",
```

```
            "Message": "Your Spot request has been submitted for review, and is
 pending evaluation."
         },
         "ProductDescription": "Linux/UNIX",
         "SpotInstanceRequestId": "sir-df6f405d",
         "State": "open",
         "LaunchSpecification": {
             "Placement": {
                 "AvailabilityZone": "us-west-2a"
             },
             "ImageId": "ami-1a2b3c4d",
             "KeyName": "my-key-pair",
             "SecurityGroups": [
                 {
                     "GroupName": "my-security-group",
                     "GroupId": "sg-1a2b3c4d"
                 }
             ],
             "Monitoring": {
                 "Enabled": false
             },
             "IamInstanceProfile": {
                 "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
             },
             "InstanceType": "m3.medium"
         },
         "Type": "one-time",
         "CreateTime": "2014-03-25T20:54:20.000Z",
         "SpotPrice": "0.050000"
      },
      ...
  ]
}
```

此範例命令會為指定子網路中的五個執行個體建立一次性 Spot 執行個體請求。Amazon EC2 會在指定子網中啟動執行個體。如果 VPC 是非預設 VPC，執行個體預設不會收到公有 IP 地址。

命令：

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 5 --type "one-
time" --launch-specification file://specification.json
```

Specification.json：

```
{
  "ImageId": "ami-1a2b3c4d",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "SubnetId": "subnet-1a2b3c4d",
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

輸出：

```
{
  "SpotInstanceRequests": [
      {
          "Status": {
             "UpdateTime": "2014-03-25T22:21:58.000Z",
             "Code": "pending-evaluation",
             "Message": "Your Spot request has been submitted for review, and is
 pending evaluation."
          },
          "ProductDescription": "Linux/UNIX",
          "SpotInstanceRequestId": "sir-df6f405d",
          "State": "open",
          "LaunchSpecification": {
             "Placement": {
                 "AvailabilityZone": "us-west-2a"
             }
             "ImageId": "ami-1a2b3c4d"
             "SecurityGroups": [
                 {
                     "GroupName": "my-security-group",
                     "GroupID": "sg-1a2b3c4d"
                 }
             ]
             "SubnetId": "subnet-1a2b3c4d",
             "Monitoring": {
                 "Enabled": false
             },
             "IamInstanceProfile": {
                 "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
             },
             "InstanceType": "m3.medium",
```

```
            },
            "Type": "one-time",
            "CreateTime": "2014-03-25T22:21:58.000Z",
            "SpotPrice": "0.050000"
        },
        ...
    ]
}
```

此範例會將公有 IP 地址指派給您在非預設 VPC 中啟動的 Spot 執行個體。請注意，當您指定網路界面時，您必須使用網路界面包含子網路 ID 和安全群組 ID。

命令：

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 1 --type "one-time" --launch-specification file://specification.json
```

Specification.json：

```
{
  "ImageId": "ami-1a2b3c4d",
  "KeyName": "my-key-pair",
  "InstanceType": "m3.medium",
  "NetworkInterfaces": [
    {
      "DeviceIndex": 0,
      "SubnetId": "subnet-1a2b3c4d",
      "Groups": [ "sg-1a2b3c4d" ],
      "AssociatePublicIpAddress": true
    }
  ],
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RequestSpotInstances。

reset-address-attribute

以下程式碼範例顯示如何使用 reset-address-attribute。

AWS CLI

重設與彈性 IP 地址相關聯的網域名稱屬性

下列`reset-address-attribute`範例會重設彈性 IP 地址的網域名稱屬性。

Linux：

```
aws ec2 reset-address-attribute \
    --allocation-id eipalloc-abcdef01234567890 \
    --attribute domain-name
```

Windows：

```
aws ec2 reset-address-attribute ^
    --allocation-id eipalloc-abcdef01234567890 ^
    --attribute domain-name
```

輸出：

```
{
    "Addresses": [
        {
            "PublicIp": "192.0.2.0",
            "AllocationId": "eipalloc-abcdef01234567890",
            "PtrRecord": "example.com."
            "PtrRecordUpdate": {
                "Value": "example.net.",
                "Status": "PENDING"
        }
    ]
}
```

若要監控待定變更，請參閱 AWS CLI 命令參考中的 describe-addresses-attribute。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetAddressAttribute。

`reset-ebs-default-kms-key-id`

以下程式碼範例顯示如何使用 `reset-ebs-default-kms-key-id`。

AWS CLI

重設 EBS 加密的預設 CMK

下列reset-ebs-default-kms-key-id範例會重設目前區域中 AWS 您帳戶 EBS 加密的預設 CMK。

```
aws ec2 reset-ebs-default-kms-key-id
```

輸出：

```
{
   "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/8c5b2c63-b9bc-45a3-
a87a-5513eEXAMPLE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetEbsDefaultKmsKeyId。

reset-fpga-image-attribute

以下程式碼範例顯示如何使用 reset-fpga-image-attribute。

AWS CLI

重設 Amazon FPGA 映像的屬性

此範例會重設指定 AFI 的載入許可。

命令：

```
aws ec2 reset-fpga-image-attribute --fpga-image-id afi-0d123e123bfc85abc --
attribute loadPermission
```

輸出：

```
{
  "Return": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetFpgaImageAttribute。

reset-image-attribute

以下程式碼範例顯示如何使用 reset-image-attribute。

AWS CLI

重設 launchPermission 屬性

此範例會將指定 AMI 的 launchPermission 屬性重設為其預設值。根據預設，AMIs是私有的。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 reset-image-attribute --image-id ami-5731123e --attribute launchPermission
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetImageAttribute。

reset-instance-attribute

以下程式碼範例顯示如何使用 reset-instance-attribute。

AWS CLI

重設 sourceDestCheck 屬性

此範例會重設指定執行個體的 sourceDestCheck 屬性。執行個體必須位於 VPC 中。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --
attribute sourceDestCheck
```

重設核心屬性

此範例會重設指定執行個體的 kernel 屬性。執行個體必須處於 stopped 狀態。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --
attribute kernel
```

重設 ramdisk 屬性

此範例會重設指定執行個體的 ramdisk 屬性。執行個體必須處於 stopped 狀態。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --
attribute ramdisk
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetInstanceAttribute。

reset-network-interface-attribute

以下程式碼範例顯示如何使用 reset-network-interface-attribute。

AWS CLI

重設網路介面屬性

下列reset-network-interface-attribute範例會將來源/目的地檢查屬性的值重設為 true。

```
aws ec2 reset-network-interface-attribute \
    --network-interface-id eni-686ea200 \
    --source-dest-check
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetNetworkInterfaceAttribute。

reset-snapshot-attribute

以下程式碼範例顯示如何使用 reset-snapshot-attribute。

AWS CLI

重設快照屬性

此範例會重設快照 的建立磁碟區許可snap-1234567890abcdef0。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 reset-snapshot-attribute --snapshot-id snap-1234567890abcdef0 --
attribute createVolumePermission
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetSnapshotAttribute。

restore-address-to-classic

以下程式碼範例顯示如何使用 restore-address-to-classic。

AWS CLI

將地址還原至 EC2-Classic

此範例會將彈性 IP 地址 198.51.100.0 還原至 EC2-Classic 平台。

命令：

```
aws ec2 restore-address-to-classic --public-ip 198.51.100.0
```

輸出：

```
{
  "Status": "MoveInProgress",
  "PublicIp": "198.51.100.0"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreAddressToClassic。

restore-image-from-recycle-bin

以下程式碼範例顯示如何使用 restore-image-from-recycle-bin。

AWS CLI

從資源回收筒還原映像

下列restore-image-from-recycle-bin範例會從資源回收筒還原 AMI ami-0111222333444abcd。

```
aws ec2 restore-image-from-recycle-bin \
```

```
    --image-id ami-0111222333444abcd
```

輸出：

```
{
    "Return": true
}
```

如需詳細資訊，請參閱《Amazon EBS 使用者指南》中的從資源回收筒復原已刪除的 AMIs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreImageFromRecycleBin。

restore-managed-prefix-list-version

以下程式碼範例顯示如何使用 restore-managed-prefix-list-version。

AWS CLI

us-west-2**還原字首清單版本**

以下內容會從指定字首清單的第 1 版restore-managed-prefix-list-version還原項目。

```
aws ec2 restore-managed-prefix-list-version \
    --prefix-list-id pl-0123456abcabcabc1 \
    --current-version 2 \
    --previous-version 1
```

輸出：

```
{
    "PrefixList": {
        "PrefixListId": "pl-0123456abcabcabc1",
        "AddressFamily": "IPv4",
        "State": "restore-in-progress",
        "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-0123456abcabcabc1",
        "PrefixListName": "vpc-cidrs",
        "MaxEntries": 10,
        "Version": 2,
        "OwnerId": "123456789012"
    }
}
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的受管字首清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreManagedPrefixListVersion。

`restore-snapshot-from-recycle-bin`

以下程式碼範例顯示如何使用 `restore-snapshot-from-recycle-bin`。

AWS CLI

從資源回收筒還原快照

下列`restore-snapshot-from-recycle-bin`範例會從資源回收筒還原快照。當您從資源回收筒還原快照時，該快照會立即可供使用，而且會從資源回收筒中移除。您可以採取您在帳戶中使用任何其他快照的同一方式來使用還原的快照。

```
aws ec2 restore-snapshot-from-recycle-bin \
    --snapshot-id snap-01234567890abcdef
```

此命令不會產生輸出。

如需資源回收筒的詳細資訊，請參閱《Amazon EBS 使用者指南》中的從資源回收筒復原已刪除的快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreSnapshotFromRecycleBin。

`restore-snapshot-tier`

以下程式碼範例顯示如何使用 `restore-snapshot-tier`。

AWS CLI

範例 1：永久還原封存的快照

下列`restore-snapshot-tier`範例會永久還原指定的快照。指定 `--snapshot-id`並包含`permanent-restore`選項。

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-01234567890abcedf \
    --permanent-restore
```

輸出：

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "IsPermanentRestore": true
}
```

如需快照封存的詳細資訊，請參閱《Amazon EBS 使用者指南》中的封存 Amazon EBS 快照。

範例 2：暫時還原封存的快照

下列restore-snapshot-tier範例會暫時還原指定的快照。省略 --permanent-restore 選項。指定 ，--snapshot-id並針對 temporary-restore-days指定要還原快照的天數。temporary-restore-days 必須以天數指定。允許的範圍是 1到 180。如果您未指定一值，其會預設為 1 天 。

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-01234567890abcedf \
    --temporary-restore-days 5
```

輸出：

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "RestoreDuration": 5,
    "IsPermanentRestore": false
}
```

如需快照封存的詳細資訊，請參閱《Amazon EBS 使用者指南》中的封存 Amazon EBS 快照。

範例 3：修改還原期間

下列restore-snapshot-tier範例會將指定快照的還原期間變更為 10 天。

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-01234567890abcedf
    --temporary-restore-days 10
```

輸出：

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "RestoreDuration": 10,
```

```
    "IsPermanentRestore": false
}
```

如需快照封存的詳細資訊，請參閱《Amazon EBS 使用者指南》中的封存 Amazon EBS 快照。

範例 4：修改還原類型

下列restore-snapshot-tier範例會將指定快照的還原類型從暫時變更為永久。

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-01234567890abcedf
    --permanent-restore
```

輸出：

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "IsPermanentRestore": true
}
```

如需快照封存的詳細資訊，請參閱《Amazon EBS 使用者指南》中的封存 Amazon EBS 快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreSnapshotTier。

revoke-client-vpn-ingress

以下程式碼範例顯示如何使用 revoke-client-vpn-ingress。

AWS CLI

撤銷 Client VPN 端點的授權規則

下列revoke-client-vpn-ingress範例會撤銷所有群組的網際網路存取 (0.0.0.0/0) 規則。

```
aws ec2 revoke-client-vpn-ingress \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --target-network-cidr 0.0.0.0/0 --revoke-all-groups
```

輸出：

```
{
    "Status": {
```

```
        "Code": "revoking"
    }
}
```

如需詳細資訊，請參閱 AWS Client VPN 管理員指南中的授權規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RevokeClientVpnIngress。

`revoke-security-group-egress`

以下程式碼範例顯示如何使用 `revoke-security-group-egress`。

AWS CLI

範例 1：移除允許傳出流量至特定地址範圍的規則

下列`revoke-security-group-egress`範例命令會移除規則，授予 TCP 連接埠 80 上指定地址範圍的存取權。

```
aws ec2 revoke-security-group-egress \
    --group-id sg-026c12253ce15eff7 \
    --ip-
permissions [{IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges=[{CidrIp=10.0.0.0/16}]
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的安全群組。

範例 2：移除允許傳出流量至特定安全群組的規則

下列`revoke-security-group-egress`範例命令會移除規則，將存取權授予 TCP 連接埠 80 上指定的安全群組。

```
aws ec2 revoke-security-group-egress \
    --group-id sg-026c12253ce15eff7 \
    --ip-permissions '[{"IpProtocol": "tcp", "FromPort": 443, "ToPort":
 443,"UserIdGroupPairs": [{"GroupId": "sg-06df23a01ff2df86d"}]}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的安全群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RevokeSecurityGroupEgress。

revoke-security-group-ingress

以下程式碼範例顯示如何使用 revoke-security-group-ingress。

AWS CLI

範例 1：從安全群組移除規則

下列revoke-security-group-ingress範例會從預設 VPC 的指定安全群組中移除203.0.113.0/24地址範圍的 TCP 連接埠 22 存取。

```
aws ec2 revoke-security-group-ingress \
    --group-name mySecurityGroup
    --protocol tcp \
    --port 22 \
    --cidr 203.0.113.0/24
```

如果成功，此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的安全群組。

範例 2：使用 IP 許可集移除規則

下列revoke-security-group-ingress範例使用 ip-permissions 參數來移除允許 ICMP 訊息的傳入規則 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set (類型 3，代碼 4)。

```
aws ec2 revoke-security-group-ingress \
    --group-id sg-026c12253ce15eff7 \
    --ip-
permissions IpProtocol=icmp,FromPort=3,ToPort=4,IpRanges=[{CidrIp=0.0.0.0/0}]
```

如果成功，此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的安全群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RevokeSecurityGroupIngress。

run-instances

以下程式碼範例顯示如何使用 run-instances。

AWS CLI

範例 1：在預設子網路中啟動執行個體

下列 run-instances 範例會在目前區域的預設子網路中啟動 t2.micro 類型的單一執行個體，並將其與該區域中預設 VPC 的預設子網路建立關聯。如果您不打算使用 SSH (Linux) 或 RDP (Windows) 連接至執行個體，則金鑰對為選用項目。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
    --key-name MyKeyPair
```

輸出：

```
{
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-0abcdef1234567890",
            "InstanceId": "i-1231231230abcdef0",
            "InstanceType": "t2.micro",
            "KeyName": "MyKeyPair",
            "LaunchTime": "2018-05-10T08:05:20.000Z",
            "Monitoring": {
                "State": "disabled"
            },
            "Placement": {
                "AvailabilityZone": "us-east-2a",
                "GroupName": "",
                "Tenancy": "default"
            },
            "PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
            "PrivateIpAddress": "10.0.0.157",
            "ProductCodes": [],
            "PublicDnsName": "",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "StateTransitionReason": "",
            "SubnetId": "subnet-04a636d18e83cfacb",
            "VpcId": "vpc-1234567890abcdef0",
```

```
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "",
            "EbsOptimized": false,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2018-05-10T08:05:20.000Z",
                        "AttachmentId": "eni-attach-0e325c07e928a0405",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching"
                    },
                    "Description": "",
                    "Groups": [
                        {
                            "GroupName": "MySecurityGroup",
                            "GroupId": "sg-0598c7d356eba48d7"
                        }
                    ],
                    "Ipv6Addresses": [],
                    "MacAddress": "0a:ab:58:e0:67:e2",
                    "NetworkInterfaceId": "eni-0c0a29997760baee7",
                    "OwnerId": "123456789012",
                    "PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
                    "PrivateIpAddress": "10.0.0.157",
                    "PrivateIpAddresses": [
                        {
                            "Primary": true,
                            "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
                            "PrivateIpAddress": "10.0.0.157"
                        }
                    ],
                    "SourceDestCheck": true,
                    "Status": "in-use",
                    "SubnetId": "subnet-04a636d18e83cfacb",
                    "VpcId": "vpc-1234567890abcdef0",
                    "InterfaceType": "interface"
                }
            ],
            "RootDeviceName": "/dev/xvda",
            "RootDeviceType": "ebs",
```

```
            "SecurityGroups": [
                {
                    "GroupName": "MySecurityGroup",
                    "GroupId": "sg-0598c7d356eba48d7"
                }
            ],
            "SourceDestCheck": true,
            "StateReason": {
                "Code": "pending",
                "Message": "pending"
            },
            "Tags": [],
            "VirtualizationType": "hvm",
            "CpuOptions": {
                "CoreCount": 1,
                "ThreadsPerCore": 1
            },
            "CapacityReservationSpecification": {
                "CapacityReservationPreference": "open"
            },
            "MetadataOptions": {
                "State": "pending",
                "HttpTokens": "optional",
                "HttpPutResponseHopLimit": 1,
                "HttpEndpoint": "enabled"
            }
        }
    ],
    "OwnerId": "123456789012",
    "ReservationId": "r-02a3f596d91211712"
}
```

範例 2：在非預設的子網路中啟動執行個體，並新增公有 IP 地址

下列 `run-instances` 範例會針對您要在非預設的子網路中啟動的執行個體請求公有 IP 地址。執行個體已與指定的安全群組建立關聯。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
    --subnet-id subnet-08fc749671b2d077c \
    --security-group-ids sg-0b0384b66d7d692f9 \
    --associate-public-ip-address \
```

```
    --key-name MyKeyPair
```

如需 run-instances 的輸出範例，請參閱範例 1。

範例 3：啟動具有額外磁碟區的執行個體

下列 run-instances 範例會使用 mapping.json 中指定的區塊型裝置映射，在啟動時連接額外的磁碟區。區塊型裝置映射可以指定 EBS 磁碟區、執行個體儲存體磁碟區，或同時指定 EBS 磁碟區和執行個體儲存體磁碟區。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
    --subnet-id subnet-08fc749671b2d077c \
    --security-group-ids sg-0b0384b66d7d692f9 \
    --key-name MyKeyPair \
    --block-device-mappings file://mapping.json
```

mapping.json 的內容。此範例會為 /dev/sdh 新增空的 EBS 磁碟區 (大小為 100 GiB)。

```
[
    {
        "DeviceName": "/dev/sdh",
        "Ebs": {
            "VolumeSize": 100
        }
    }
]
```

mapping.json 的內容。此範例會將 ephemeral1 新增為執行個體儲存體磁碟區。

```
[
    {
        "DeviceName": "/dev/sdc",
        "VirtualName": "ephemeral1"
    }
]
```

如需 run-instances 的輸出範例，請參閱範例 1。

如需區塊型裝置映射的詳細資訊，請參閱中的《Amazon EC2 使用者指南》的區塊型裝置映射。

範例 4：啟動執行個體並在建立時新增標籤

下列 run-instances 範例會將索引鍵為 webserver、值為 production 的標籤新增至執行個體。命令也會為任何建立的 EBS 磁碟區 (此範例中為根磁碟區) 套用鍵為 cost-center，值為 cc123 的標籤。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
    --count 1 \
    --subnet-id subnet-08fc749671b2d077c \
    --key-name MyKeyPair \
    --security-group-ids sg-0b0384b66d7d692f9 \
    --tag-specifications
 'ResourceType=instance,Tags=[{Key=webserver,Value=production}]'
 'ResourceType=volume,Tags=[{Key=cost-center,Value=cc123}]'
```

如需 run-instances 的輸出範例，請參閱範例 1。

範例 5：使用使用者資料啟動執行個體

下列 run-instances 範例會將使用者資料傳遞至名為 my_script.txt 的檔案中，且該檔案包含您執行個體的組態指令碼。指令碼會在啟動時執行。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
    --count 1 \
    --subnet-id subnet-08fc749671b2d077c \
    --key-name MyKeyPair \
    --security-group-ids sg-0b0384b66d7d692f9 \
    --user-data file://my_script.txt
```

如需 run-instances 的輸出範例，請參閱範例 1。

如需執行個體使用者資料的詳細資訊，請參閱《Amazon EC2 使用者指南》中的使用執行個體使用者資料。

範例 6：啟動爆量效能執行個體

下列 run-instances 範例會啟動具有 unlimited 積分選項的 t2.micro 執行個體。啟動 T2 執行個體時，如果您未指定 --credit-specification，預設值為 standard 積分選項。啟動 T3 執行個體時，預設值為 unlimited 積分選項。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
    --count 1 \
    --subnet-id subnet-08fc749671b2d077c \
    --key-name MyKeyPair \
    --security-group-ids sg-0b0384b66d7d692f9 \
    --credit-specification CpuCredits=unlimited
```

如需 run-instances 的輸出範例，請參閱範例 1。

如需爆量效能執行個體的詳細資訊，請參閱《Amazon EC2 使用者指南》中的爆量效能執行個體。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 RunInstances。

run-scheduled-instances

以下程式碼範例顯示如何使用 run-scheduled-instances。

AWS CLI

啟動排程執行個體

此範例會在 VPC 中啟動指定的排程執行個體。

命令：

```
aws ec2 run-scheduled-instances --scheduled-instance-
id sci-1234-1234-1234-1234-123456789012 --instance-count 1 --launch-
specification file://launch-specification.json
```

Launch-specification.json：

```
{
  "ImageId": "ami-12345678",
  "KeyName": "my-key-pair",
  "InstanceType": "c4.large",
  "NetworkInterfaces": [
```

```
    {
        "DeviceIndex": 0,
        "SubnetId": "subnet-12345678",
        "AssociatePublicIpAddress": true,
        "Groups": ["sg-12345678"]
    }
  ],
  "IamInstanceProfile": {
      "Name": "my-iam-role"
  }
}
```

輸出：

```
{
  "InstanceIdSet": [
      "i-1234567890abcdef0"
  ]
}
```

此範例會在 EC2-Classic 中啟動指定的排程執行個體。

命令：

```
aws ec2 run-scheduled-instances --scheduled-instance-
id sci-1234-1234-1234-1234-123456789012 --instance-count 1 --launch-
specification file://launch-specification.json
```

Launch-specification.json：

```
{
  "ImageId": "ami-12345678",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": ["sg-12345678"],
  "InstanceType": "c4.large",
  "Placement": {
    "AvailabilityZone": "us-west-2b"
  }
  "IamInstanceProfile": {
      "Name": "my-iam-role"
  }
}
```

輸出：

```
{
  "InstanceIdSet": [
      "i-1234567890abcdef0"
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RunScheduledInstances。

search-local-gateway-routes

以下程式碼範例顯示如何使用 search-local-gateway-routes。

AWS CLI

在本機閘道路由表中搜尋路由

下列search-local-gateway-routes範例會在指定的本機閘道路由表中搜尋靜態路由。

```
aws ec2 search-local-gateway-routes \
    --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \
    --filters "Name=type,Values=static"
```

輸出：

```
{
    "Route": {
        "DestinationCidrBlock": "0.0.0.0/0",
        "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
        "Type": "static",
        "State": "deleted",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7EXAMPLE"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchLocalGatewayRoutes。

search-transit-gateway-multicast-groups

以下程式碼範例顯示如何使用 search-transit-gateway-multicast-groups。

AWS CLI

搜尋一或多個傳輸閘道多點傳送群組並傳回群組成員資格資訊

下列search-transit-gateway-multicast-groups範例會傳回指定多點傳送群組的群組成員資格。

```
aws ec2 search-transit-gateway-multicast-groups \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-000fb24d04EXAMPLE
```

輸出：

```
{
    "MulticastGroups": [
        {
            "GroupIpAddress": "224.0.1.0",
            "TransitGatewayAttachmentId": "tgw-attach-0372e72386EXAMPLE",
            "SubnetId": "subnet-0187aff814EXAMPLE",
            "ResourceId": "vpc-0065acced4EXAMPLE",
            "ResourceType": "vpc",
            "NetworkInterfaceId": "eni-03847706f6EXAMPLE",
            "GroupMember": false,
            "GroupSource": true,
            "SourceType": "static"
        }
    ]
}
```

如需詳細資訊，請參閱 Transit Gateways 指南中的傳輸閘道上的多點傳送。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchTransitGatewayMulticastGroups。

search-transit-gateway-routes

以下程式碼範例顯示如何使用 search-transit-gateway-routes。

AWS CLI

在指定的傳輸閘道路由表中搜尋路由

下列search-transit-gateway-routes範例會傳回static指定路由表中類型 的所有路由。

```
aws ec2 search-transit-gateway-routes \
    --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \
    --filters "Name=type,Values=static"
```

輸出：

```
{
    "Routes": [
        {
            "DestinationCidrBlock": "10.0.2.0/24",
            "TransitGatewayAttachments": [
                {
                    "ResourceId": "vpc-4EXAMPLE",
                    "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
                    "ResourceType": "vpc"
                }
            ],
            "Type": "static",
            "State": "active"
        },
        {
            "DestinationCidrBlock": "10.1.0.0/24",
            "TransitGatewayAttachments": [
                {
                    "ResourceId": "vpc-4EXAMPLE",
                    "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
                    "ResourceType": "vpc"
                }
            ],
            "Type": "static",
            "State": "active"
        }
    ],
    "AdditionalRoutesAvailable": false
}
```

如需詳細資訊，請參閱《傳輸閘道指南》中的傳輸閘道路由表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchTransitGatewayRoutes。

send-diagnostic-interrupt

以下程式碼範例顯示如何使用 send-diagnostic-interrupt。

AWS CLI

傳送診斷中斷

下列send-diagnostic-interrupt範例會將診斷中斷傳送至指定的執行個體。

```
aws ec2 send-diagnostic-interrupt \
    --instance-id i-1234567890abcdef0
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SendDiagnosticInterrupt。

start-instances

以下程式碼範例顯示如何使用 start-instances。

AWS CLI

啟動 Amazon EC2 執行個體

此範例會啟動指定且受 Amazon EBS 支援的執行個體。

命令：

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

輸出：

```
{
    "StartingInstances": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "PreviousState": {
```

```
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的「停止和啟動執行個體」。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 StartInstances。

`start-network-insights-access-scope-analysis`

以下程式碼範例顯示如何使用 `start-network-insights-access-scope-analysis`。

AWS CLI

啟動 Network Insights 存取範圍分析

下列`start-network-insights-access-scope-analysis`範例會在您的帳戶中開始範圍分析 AWS 。

```
aws ec2 start-network-insights-access-scope-analysis \
    --region us-east-1 \
    --network-insights-access-scope-id nis-123456789111
```

輸出：

```
{
    "NetworkInsightsAccessScopeAnalysis": {
        "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789222",
        "NetworkInsightsAccessScopeAnalysisArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope-analysis/nisa-123456789222",
        "NetworkInsightsAccessScopeId": "nis-123456789111",
        "Status": "running",
        "StartDate": "2022-01-26T00:47:06.814000+00:00"
    }
}
```

如需詳細資訊，請參閱《Network Access Analyzer 指南》中的使用 AWS CLI 開始使用 Network Access Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartNetworkInsightsAccessScopeAnalysis。

start-network-insights-analysis

以下程式碼範例顯示如何使用 start-network-insights-analysis。

AWS CLI

分析路徑

下列start-network-insights-analysis範例會分析來源和目的地之間的路徑。若要檢視路徑分析的結果，請使用 describe-network-insights-analyses命令。

```
aws ec2 start-network-insights-analysis \
    --network-insights-path-id nip-0b26f224f1d131fa8
```

輸出：

```
{
    "NetworkInsightsAnalysis": {
        "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a",
        "NetworkInsightsAnalysisArn": "arn:aws:ec2:us-east-1:123456789012:network-
insights-analysis/nia-02207aa13eb480c7a",
        "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",
        "StartDate": "2021-01-20T22:58:37.495Z",
        "Status": "running"
    }
}
```

如需詳細資訊，請參閱 Reachability Analyzer 指南中的開始使用 AWS CLI。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartNetworkInsightsAnalysis。

start-vpc-endpoint-service-private-dns-verification

以下程式碼範例顯示如何使用 start-vpc-endpoint-service-private-dns-verification。

AWS CLI

啟動 DNS 驗證程序

下列start-vpc-endpoint-service-private-dns-verification範例會啟動指定端點服務的 DNS 驗證程序。

```
aws ec2 start-vpc-endpoint-service-private-dns-verification \
    --service-id vpce-svc-071afff70666e61e0
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS PrivateLink 使用者指南中的管理 DNS 名稱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartVpcEndpointServicePrivateDnsVerification。

stop-instances

以下程式碼範例顯示如何使用 stop-instances。

AWS CLI

範例 1：停止 Amazon EC2 執行個體

下列 stop-instances 範例會停止指定且受 Amazon EBS 支援的執行個體。

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0
```

輸出：

```
{
    "StoppingInstances": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
```

```
    ]
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的停止和啟動執行個體。

範例 2：讓 Amazon EC2 執行個體休眠

下列 `stop-instances` 範例會在 Amazon EBS 已啟用休眠並符合休眠必要條件時，讓該執行個體進入休眠。執行個體進入休眠狀態之後，即停止運作。

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --hibernate
```

輸出：

```
{
    "StoppingInstances": [
        {
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "InstanceId": "i-1234567890abcdef0",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的讓您的隨需 Linux 執行個體休眠。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 StopInstances。

terminate-client-vpn-connections

以下程式碼範例顯示如何使用 `terminate-client-vpn-connections`。

AWS CLI

終止與 Client VPN 端點的連線

下列terminate-client-vpn-connections範例會終止與 Client VPN 端點的指定連線。

```
aws ec2 terminate-client-vpn-connections \
    --client-vpn-endpoint-id vpn-endpoint-123456789123abcde \
    --connection-id cvpn-connection-04edd76f5201e0cb8
```

輸出：

```
{
    "ClientVpnEndpointId": "vpn-endpoint-123456789123abcde",
    "ConnectionStatuses": [
        {
            "ConnectionId": "cvpn-connection-04edd76f5201e0cb8",
            "PreviousStatus": {
                "Code": "active"
            },
            "CurrentStatus": {
                "Code": "terminating"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 Client VPN 管理員指南中的用戶端連線。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TerminateClientVpnConnections。

terminate-instances

以下程式碼範例顯示如何使用 terminate-instances。

AWS CLI

若要終止 Amazon EC2 執行個體

此範例會終止指定執行個體。

命令：

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

輸出：

```
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《AWS 命令行介面使用者指南》中的「Amazon EC2 執行個體」。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TerminateInstances。

unassign-ipv6-addresses

以下程式碼範例顯示如何使用 unassign-ipv6-addresses。

AWS CLI

從網路界面取消指派 IPv6 地址

此範例會從指定的網路界面取消指派指定的 IPv6 地址。

命令：

```
aws ec2 unassign-ipv6-addresses --ipv6-
addresses 2001:db8:1234:1a00:3304:8879:34cf:4071 --network-interface-id eni-23c49b68
```

輸出：

```
{
  "NetworkInterfaceId": "eni-23c49b68",
  "UnassignedIpv6Addresses": [
      "2001:db8:1234:1a00:3304:8879:34cf:4071"
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UnassignIpv6Addresses。

unassign-private-ip-addresses

以下程式碼範例顯示如何使用 unassign-private-ip-addresses。

AWS CLI

從網路界面取消指派次要私有 IP 地址

此範例會從指定的網路界面取消指派指定的私有 IP 地址。如果命令成功，則不會傳回任何輸出。

命令：

```
aws ec2 unassign-private-ip-addresses --network-interface-id eni-e5aa89a3 --private-ip-addresses 10.0.0.82
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UnassignPrivateIpAddresses。

unassign-private-nat-gateway-address

以下程式碼範例顯示如何使用 unassign-private-nat-gateway-address。

AWS CLI

從私有 NAT 閘道取消指派私有 IP 地址

下列unassign-private-nat-gateway-address範例會從指定的私有 NAT 閘道取消指派指定的 IP 地址。

```
aws ec2 unassign-private-nat-gateway-address \
    --nat-gateway-id nat-1234567890abcdef0 \
    --private-ip-addresses 10.0.20.197
```

輸出：

```
{
    "NatGatewayId": "nat-0ee3edd182361f662",
    "NatGatewayAddresses": [
        {
            "NetworkInterfaceId": "eni-0065a61b324d1897a",
            "PrivateIp": "10.0.20.197",
            "IsPrimary": false,
            "Status": "unassigning"
        }
    ]
}
```

如需詳細資訊，請參閱 《Amazon VPC 使用者指南》中的 NAT 閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UnassignPrivateNatGatewayAddress。

unlock-snapshot

以下程式碼範例顯示如何使用 unlock-snapshot。

AWS CLI

解除鎖定快照

下列unlock-snapshot範例會解除鎖定指定的快照。

```
aws ec2 unlock-snapshot \
    --snapshot-id snap-0b5e733b4a8df6e0d
```

輸出：

```
{
    "SnapshotId": "snap-0b5e733b4a8df6e0d"
}
```

如需詳細資訊，請參閱《Amazon EBS 使用者指南》中的快照鎖定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UnlockSnapshot。

unmonitor-instances

以下程式碼範例顯示如何使用 unmonitor-instances。

AWS CLI

停用執行個體的詳細監控

這個範例命令會停用指定執行個體的詳細監控。

命令：

```
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

輸出：

```
{
  "InstanceMonitorings": [
      {
          "InstanceId": "i-1234567890abcdef0",
          "Monitoring": {
              "State": "disabling"
          }
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UnmonitorInstances。

update-security-group-rule-descriptions-egress

以下程式碼範例顯示如何使用 update-security-group-rule-descriptions-egress。

AWS CLI

更新傳出安全群組規則的描述

下列update-security-group-rule-descriptions-egress範例會更新指定連接埠和 IPv4 地址範圍之安全群組規則的描述。描述 'Outbound HTTP access to server 2' 會取代規則的任何現有描述。

```
aws ec2 update-security-group-rule-descriptions-egress \
```

```
    --group-id sg-02f0d35a850ba727f \
    --ip-permissions
 IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges=[{CidrIp=203.0.113.0/24,Description="Outbound
 HTTP access to server 2"}]
```

輸出：

```
{
    "Return": true
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的安全群組規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSecurityGroupRuleDescriptionsEgress。

update-security-group-rule-descriptions-ingress

以下程式碼範例顯示如何使用 update-security-group-rule-descriptions-ingress。

AWS CLI

範例 1：使用 CIDR 來源更新傳入安全群組規則的描述

下列update-security-group-rule-descriptions-ingress範例會更新指定連接埠和 IPv4 地址範圍之安全群組規則的描述。描述 'SSH access from ABC office' 會取代規則的任何現有描述。

```
aws ec2 update-security-group-rule-descriptions-ingress \
    --group-id sg-02f0d35a850ba727f \
    --ip-permissions
 IpProtocol=tcp,FromPort=22,ToPort=22,IpRanges='[{CidrIp=203.0.113.0/16,Description="SSH
 access from corpnet"}]'
```

輸出：

```
{
    "Return": true
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的安全群組規則。

範例 2：使用字首清單來源更新傳入安全群組規則的描述

下列update-security-group-rule-descriptions-ingress範例會更新指定連接埠和字首清單之安全群組規則的描述。描述 'SSH access from ABC office' 會取代規則的任何現有描述。

```
aws ec2 update-security-group-rule-descriptions-ingress \
    --group-id sg-02f0d35a850ba727f \
    --ip-permissions
 IpProtocol=tcp,FromPort=22,ToPort=22,PrefixListIds='[{PrefixListId=pl-12345678,Description=
 access from corpnet"}]'
```

輸出：

```
{
    "Return": true
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的安全群組規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSecurityGroupRuleDescriptionsIngress。

withdraw-byoip-cidr

以下程式碼範例顯示如何使用 withdraw-byoip-cidr。

AWS CLI

停止公告地址範圍

下列withdraw-byoip-cidr範例會停止公告指定的地址範圍。

```
aws ec2 withdraw-byoip-cidr
    --cidr 203.0.113.25/24
```

輸出：

```
{
    "ByoipCidr": {
        "Cidr": "203.0.113.25/24",
        "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
```

```
        "State": "advertised"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 WithdrawByoipCidr。

使用 的 Amazon EC2 Instance Connect 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon EC2 Instance Connect 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`send-ssh-public-key`

以下程式碼範例顯示如何使用 `send-ssh-public-key`。

AWS CLI

將 SSH 公有金鑰傳送至執行個體

下列`send-ssh-public-key`範例會將指定的 SSH 公有金鑰傳送至指定的執行個體。金鑰用於驗證指定的使用者。

```
aws ec2-instance-connect send-ssh-public-key \
    --instance-id i-1234567890abcdef0 \
    --instance-os-user ec2-user \
    --availability-zone us-east-2b \
    --ssh-public-key file://path/my-rsa-key.pub
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SendSshPublicKey。

使用 的 Amazon ECR 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon ECR 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`batch-check-layer-availability`

以下程式碼範例顯示如何使用 `batch-check-layer-availability`。

AWS CLI

檢查 layer 的可用性

下列`batch-check-layer-availability`範例會檢查儲存`cluster-autoscaler`庫`sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb`具有摘要之 layer 的可用性。

```
aws ecr batch-check-layer-availability \
    --repository-name cluster-autoscaler \
    --layer-
digests sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed
```

輸出：

```
{
    "layers": [
        {
            "layerDigest":
 "sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed",
            "layerAvailability": "AVAILABLE",
            "layerSize": 2777,
            "mediaType": "application/vnd.docker.container.image.v1+json"
```

```
        }
    ],
    "failures": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchCheckLayerAvailability。

batch-delete-image

以下程式碼範例顯示如何使用 batch-delete-image。

AWS CLI

範例 1：刪除映像

下列batch-delete-image範例會刪除 帳戶預設登錄檔中指定儲存庫precise中具有 標籤的映像。

```
aws ecr batch-delete-image \
    --repository-name ubuntu \
    --image-ids imageTag=precise
```

輸出：

```
{
    "failures": [],
    "imageIds": [
        {
            "imageTag": "precise",
            "imageDigest":
 "sha256:19665f1e6d1e504117a1743c0a3d3753086354a38375961f2e665416ef4b1b2f"
        }
    ]
}
```

範例 2：刪除多個映像

下列batch-delete-image範例會刪除指定儲存庫team1中以 prod和 標記的所有映像。

```
aws ecr batch-delete-image \
    --repository-name MyRepository \
    --image-ids imageTag=prod imageTag=team1
```

輸出：

```
{
    "imageIds": [
        {
            "imageDigest": "sha256:123456789012",
            "imageTag": "prod"
        },
        {
            "imageDigest": "sha256:567890121234",
            "imageTag": "team1"
        }
    ],
    "failures": []
}
```

如需詳細資訊，請參閱《Amazon ECR 使用者指南》中的刪除映像。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDeleteImage。

batch-get-image

以下程式碼範例顯示如何使用 batch-get-image。

AWS CLI

範例 1：取得映像

下列batch-get-image範例會取得 帳戶中預設登錄cluster-autoscaler檔中名為 之儲存庫v1.13.6中具有 標籤的映像。

```
aws ecr batch-get-image \
    --repository-name cluster-autoscaler \
    --image-ids imageTag=v1.13.6
```

輸出：

```
{
    "images": [
        {
            "registryId": "012345678910",
            "repositoryName": "cluster-autoscaler",
```

```
            "imageId": {
                "imageDigest":
 "sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",
                "imageTag": "v1.13.6"
            },
            "imageManifest": "{\n   \"schemaVersion\": 2,\n
 \"mediaType\": \"application/vnd.docker.distribution.manifest.v2+json
\",\n   \"config\": {\n      \"mediaType\": \"application/
vnd.docker.container.image.v1+json\",\n      \"size\": 2777,\n      \"digest
\": \"sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed
\"\n   },\n   \"layers\": [\n      {\n         \"mediaType
\": \"application/vnd.docker.image.rootfs.diff.tar.gzip
\",\n         \"size\": 17743696,\n         \"digest\":
 \"sha256:39fafc05754f195f134ca11ecdb1c9a691ab0848c697fffeb5a85f900caaf6e1\"\n
      },\n      {\n         \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n         \"size\": 2565026,\n
 \"digest\":
 \"sha256:8c8a779d3a537b767ae1091fe6e00c2590afd16767aa6096d1b318d75494819f
\"\n      },\n      {\n         \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n         \"size\": 28005981,\n
 \"digest\":
 \"sha256:c44ba47496991c9982ee493b47fd25c252caabf2b4ae7dd679c9a27b6a3c8fb7\"\n
      },\n      {\n         \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n         \"size\": 775,\n         \"digest
\": \"sha256:e2c388b44226544363ca007be7b896bcce1baebea04da23cbd165eac30be650f\"\n
   }\n   ]\n}"
        }
    ],
    "failures": []
}
```

範例 2：取得多個映像

下列batch-get-image範例顯示指定儲存庫team1中以 prod和 標記的所有影像的詳細資訊。

```
aws ecr batch-get-image \
    --repository-name MyRepository \
    --image-ids imageTag=prod imageTag=team1
```

輸出：

```
{
    "images": [
```

```
        {
            "registryId": "123456789012",
            "repositoryName": "MyRepository",
            "imageId": {
                "imageDigest": "sha256:123456789012",
                "imageTag": "prod"
            },
            "imageManifest": "manifestExample1"
        },
        {
            "registryId": "567890121234",
            "repositoryName": "MyRepository",
            "imageId": {
                "imageDigest": "sha256:123456789012",
                "imageTag": "team1"
            },
            "imageManifest": "manifestExample2"
        }
    ],
    "failures": []
}
```

如需詳細資訊，請參閱《Amazon ECR 使用者指南》中的映像。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetImage。

complete-layer-upload

以下程式碼範例顯示如何使用 complete-layer-upload。

AWS CLI

若要完成映像層上傳

下列complete-layer-upload範例會完成映像層上傳至layer-test儲存庫。

```
aws ecr complete-layer-upload \
    --repository-name layer-test \
    --upload-id 6cb64b8a-9378-0e33-2ab1-b780fab8a9e9 \
    --layer-digests 6cb64b8a-9378-0e33-2ab1-
b780fab8a9e9:48074e6d3a68b39aad8ccc002cdad912d4148c0f92b3729323e
```

輸出：

```
{
    "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9",
    "layerDigest":
 "sha256:9a77f85878aa1906f2020a0ecdf7a7e962d57e882250acd773383224b3fe9a02",
    "repositoryName": "layer-test",
    "registryId": "130757420319"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CompleteLayerUpload。

create-repository

以下程式碼範例顯示如何使用 create-repository。

AWS CLI

範例 1：建立儲存庫

下列create-repository範例會在 帳戶的預設登錄檔中指定的命名空間內建立儲存庫。

```
aws ecr create-repository \
    --repository-name project-a/sample-repo
```

輸出：

```
{
    "repository": {
        "registryId": "123456789012",
        "repositoryName": "project-a/sample-repo",
        "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/
sample-repo"
    }
}
```

如需詳細資訊，請參閱《Amazon ECR 使用者指南》中的建立儲存庫。

範例 2：建立以影像標籤不可變性設定的儲存庫

下列create-repository範例會在 帳戶的預設登錄檔中建立針對標籤不可變性設定的儲存庫。

```
aws ecr create-repository \
```

```
    --repository-name project-a/sample-repo \
    --image-tag-mutability IMMUTABLE
```

輸出：

```
{
    "repository": {
        "registryId": "123456789012",
        "repositoryName": "project-a/sample-repo",
        "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/
sample-repo",
        "imageTagMutability": "IMMUTABLE"
    }
}
```

如需詳細資訊，請參閱《Amazon ECR 使用者指南》中的映像標籤可變更性。

範例 3：建立使用掃描組態設定的儲存庫

下列create-repository範例會建立儲存庫，其設定為在 帳戶的預設登錄檔中的映像推送時執行漏洞掃描。

```
aws ecr create-repository \
    --repository-name project-a/sample-repo \
    --image-scanning-configuration scanOnPush=true
```

輸出：

```
{
    "repository": {
        "registryId": "123456789012",
        "repositoryName": "project-a/sample-repo",
        "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/
sample-repo",
        "imageScanningConfiguration": {
            "scanOnPush": true
        }
    }
}
```

如需詳細資訊，請參閱《Amazon ECR 使用者指南》中的映像掃描。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRepository。

delete-lifecycle-policy

以下程式碼範例顯示如何使用 delete-lifecycle-policy。

AWS CLI

刪除儲存庫的生命週期政策

下列delete-lifecycle-policy範例會刪除hello-world儲存庫的生命週期政策。

```
aws ecr delete-lifecycle-policy \
    --repository-name hello-world
```

輸出：

```
{
    "registryId": "012345678910",
    "repositoryName": "hello-world",
    "lifecyclePolicyText": "{\"rules\":[{\"rulePriority\":1,\"description\":\"Remove
 untagged images.\",\"selection\":{\"tagStatus\":\"untagged\",\"countType\":
\"sinceImagePushed\",\"countUnit\":\"days\",\"countNumber\":10},\"action\":{\"type
\":\"expire\"}}]}",
    "lastEvaluatedAt": 0.0
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLifecyclePolicy。

delete-repository-policy

以下程式碼範例顯示如何使用 delete-repository-policy。

AWS CLI

刪除儲存庫的儲存庫政策

下列delete-repository-policy範例會刪除儲存庫的cluster-autoscaler儲存庫政策。

```
aws ecr delete-repository-policy \
    --repository-name cluster-autoscaler
```

輸出：

```
{
    "registryId": "012345678910",
    "repositoryName": "cluster-autoscaler",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
 \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" :
 \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage
\", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRepositoryPolicy。

delete-repository

以下程式碼範例顯示如何使用 delete-repository。

AWS CLI

刪除儲存庫

下列delete-repository範例命令強制刪除 帳戶預設登錄檔中指定的儲存庫。如果儲存庫包含映像，則需要 --force旗標。

```
aws ecr delete-repository \
    --repository-name ubuntu \
    --force
```

輸出：

```
{
    "repository": {
        "registryId": "123456789012",
        "repositoryName": "ubuntu",
        "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/ubuntu"
    }
}
```

如需詳細資訊，請參閱《Amazon ECR 使用者指南》中的刪除儲存庫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRepository。

describe-image-scan-findings

以下程式碼範例顯示如何使用 describe-image-scan-findings。

AWS CLI

描述影像的掃描問題清單

下列describe-image-scan-findings範例會使用 帳戶預設登錄檔中指定儲存庫中的映像摘要，傳回映像的映像掃描問題清單。

```
aws ecr describe-image-scan-findings \
    --repository-name sample-repo \
    --image-
id imageDigest=sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6
```

輸出：

```
{
    "imageScanFindings": {
      "findings": [
          {
              "name": "CVE-2019-5188",
              "description": "A code execution vulnerability exists in the directory
 rehashing functionality of E2fsprogs e2fsck 1.45.4. A specially crafted ext4
 directory can cause an out-of-bounds write on the stack, resulting in code
 execution. An attacker can corrupt a partition to trigger this vulnerability.",
              "uri": "http://people.ubuntu.com/~ubuntu-security/cve/CVE-2019-5188",
              "severity": "MEDIUM",
              "attributes": [
                  {
                      "key": "package_version",
                      "value": "1.44.1-1ubuntu1.1"
                  },
                  {
                      "key": "package_name",
                      "value": "e2fsprogs"
                  },
                  {
                      "key": "CVSS2_VECTOR",
                      "value": "AV:L/AC:L/Au:N/C:P/I:P/A:P"
                  },
                  {
```

```
                    "key": "CVSS2_SCORE",
                    "value": "4.6"
                }
            ]
        }
    ],
    "imageScanCompletedAt": 1579839105.0,
    "vulnerabilitySourceUpdatedAt": 1579811117.0,
    "findingSeverityCounts": {
        "MEDIUM": 1
    }
  },
  "registryId": "123456789012",
  "repositoryName": "sample-repo",
  "imageId": {
      "imageDigest":
 "sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6"
  },
  "imageScanStatus": {
      "status": "COMPLETE",
      "description": "The scan was completed successfully."
  }
}
```

如需詳細資訊，請參閱《Amazon ECR 使用者指南》中的映像掃描。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeImageScanFindings。

describe-images

以下程式碼範例顯示如何使用 describe-images。

AWS CLI

描述儲存庫中的映像

以下describe-images範例顯示具有標籤 之cluster-autoscaler儲存庫中影像的詳細資訊v1.13.6。

```
aws ecr describe-images \
    --repository-name cluster-autoscaler \
    --image-ids imageTag=v1.13.6
```

輸出：

```
{
    "imageDetails": [
        {
            "registryId": "012345678910",
            "repositoryName": "cluster-autoscaler",
            "imageDigest":
 "sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",
            "imageTags": [
                "v1.13.6"
            ],
            "imageSizeInBytes": 48318255,
            "imagePushedAt": 1565128275.0
        }
    ]
}
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeImages。

describe-repositories

以下程式碼範例顯示如何使用 describe-repositories。

AWS CLI

描述登錄檔中的儲存庫

此範例說明 帳戶預設登錄檔中的儲存庫。

命令：

```
aws ecr describe-repositories
```

輸出：

```
{
    "repositories": [
        {
            "registryId": "012345678910",
            "repositoryName": "ubuntu",
```

```
            "repositoryArn": "arn:aws:ecr:us-west-2:012345678910:repository/ubuntu"
        },
        {
            "registryId": "012345678910",
            "repositoryName": "test",
            "repositoryArn": "arn:aws:ecr:us-west-2:012345678910:repository/test"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRepositories。

get-authorization-token

以下程式碼範例顯示如何使用 get-authorization-token。

AWS CLI

取得預設登錄檔的授權字符

下列get-authorization-token範例命令會取得預設登錄檔的授權字符。

```
aws ecr get-authorization-token
```

輸出：

```
{
    "authorizationData": [
        {
            "authorizationToken": "QVdTOkN...",
            "expiresAt": 1448875853.241,
            "proxyEndpoint": "https://123456789012.dkr.ecr.us-west-2.amazonaws.com"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAuthorizationToken。

get-download-url-for-layer

以下程式碼範例顯示如何使用 get-download-url-for-layer。

AWS CLI

取得 layer 的下載 URL

下列get-download-url-for-layer範例顯示儲存cluster-autoscaler庫sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb具有摘要之 layer 的下載 URL。

```
aws ecr get-download-url-for-layer \
    --repository-name cluster-autoscaler  \
    --layer-
digest sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed
```

輸出：

```
{
    "downloadUrl": "https://prod-us-west-2-starport-layer-bucket.s3.us-
west-2.amazonaws.com/e501-012345678910-9cb60dc0-7284-5643-3987-
da6dac0465f0/04620aac-66a5-4167-8232-55ee7ef6d565?X-Amz-Algorithm=AWS4-HMAC-
SHA256&X-Amz-Date=20190814T220617Z&X-Amz-SignedHeaders=host&X-Amz-Expires=3600&X-
Amz-Credential=AKIA32P3D2JDNMVAJLGF%2F20190814%2Fus-west-2%2Fs3%2Faws4_request&X-
Amz-Signature=9161345894947a1672467a0da7a1550f2f7157318312fe4941b59976239c3337",
    "layerDigest":
 "sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDownloadUrlForLayer。

get-lifecycle-policy-preview

以下程式碼範例顯示如何使用 get-lifecycle-policy-preview。

AWS CLI

擷取生命週期政策預覽的詳細資訊

下列get-lifecycle-policy-preview範例會擷取帳戶預設登錄檔中指定儲存庫的生命週期政策預覽結果。

命令：

```
aws ecr get-lifecycle-policy-preview \
```

```
    --repository-name "project-a/amazon-ecs-sample"
```

輸出：

```
{
    "registryId": "012345678910",
    "repositoryName": "project-a/amazon-ecs-sample",
    "lifecyclePolicyText": "{\n    \"rules\": [\n        {\n
 \"rulePriority\": 1,\n            \"description\": \"Expire images older than 14
 days\",\n            \"selection\": {\n                \"tagStatus\": \"untagged\",
\n                \"countType\": \"sinceImagePushed\",\n                \"countUnit
\": \"days\",\n                \"countNumber\": 14\n            },\n
 \"action\": {\n                \"type\": \"expire\"\n            }\n        }\n
  ]\n}\n",
    "status": "COMPLETE",
    "previewResults": [],
    "summary": {
        "expiringImageTotalCount": 0
    }
}
```

如需詳細資訊，請參閱《Amazon ECR 使用者指南》中的生命週期政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLifecyclePolicyPreview。

get-lifecycle-policy

以下程式碼範例顯示如何使用 get-lifecycle-policy。

AWS CLI

擷取生命週期政策

下列get-lifecycle-policy範例會在帳戶的預設登錄檔中顯示指定儲存庫的生命週期政策詳細資訊。

```
aws ecr get-lifecycle-policy \
    --repository-name "project-a/amazon-ecs-sample"
```

輸出：

```
{
```

```
    "registryId": "123456789012",
    "repositoryName": "project-a/amazon-ecs-sample",
    "lifecyclePolicyText": "{\"rules\":[{\"rulePriority\":1,\"description\":
\"Expire images older than 14 days\",\"selection\":{\"tagStatus\":\"untagged\",
\"countType\":\"sinceImagePushed\",\"countUnit\":\"days\",\"countNumber\":14},
\"action\":{\"type\":\"expire\"}}]}",
    "lastEvaluatedAt": 1504295007.0
}
```

如需詳細資訊，請參閱《Amazon ECR 使用者指南》中的生命週期政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLifecyclePolicy。

get-login-password

以下程式碼範例顯示如何使用 get-login-password。

AWS CLI

擷取要向登錄檔進行身分驗證的密碼

以下get-login-password顯示密碼，您可以與您選擇的容器用戶端搭配使用，以驗證您的 IAM 主體可存取的任何 Amazon ECR 登錄檔。

```
aws ecr get-login-password
```

輸出：

```
<password>
```

若要搭配 Docker CLI 使用 ，請將 get-login-password命令的輸出輸送至 docker login命令。擷取密碼時，請確定您指定 Amazon ECR 登錄檔所在的相同區域。

```
aws ecr get-login-password \
    --region <region>
 \
| docker login \
    --username AWS \
    --password-stdin <aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

如需詳細資訊，請參閱《Amazon ECR 使用者指南》中的登錄驗證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLoginPassword。

get-login

以下程式碼範例顯示如何使用 get-login。

AWS CLI

將 Docker 登入命令擷取到您的預設登錄檔

此範例會列印命令，讓您用來登入預設的 Amazon ECR 登錄檔。

命令：

```
aws ecr get-login
```

輸出：

```
docker login -u AWS -p <password> -e none https://
<aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

登入另一個帳戶的登錄檔

此範例會列印一或多個命令，您可用來登入與其他 帳戶相關聯的 Amazon ECR 登錄檔。

命令：

```
aws ecr get-login --registry-ids 012345678910 023456789012
```

輸出：

```
docker login -u <username> -p <token-1> -e none <endpoint-1>
docker login -u <username> -p <token-2> -e none <endpoint-2>
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLogin。

get-repository-policy

以下程式碼範例顯示如何使用 get-repository-policy。

AWS CLI

擷取儲存庫的儲存庫政策

下列get-repository-policy範例顯示cluster-autoscaler儲存庫政策的詳細資訊。

```
aws ecr get-repository-policy \
    --repository-name cluster-autoscaler
```

輸出：

```
{
    "registryId": "012345678910",
    "repositoryName": "cluster-autoscaler",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
 \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" :
 \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage
\", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRepositoryPolicy。

initiate-layer-upload

以下程式碼範例顯示如何使用 initiate-layer-upload。

AWS CLI

啟動映像層上傳

下列initiate-layer-upload範例會啟動映像層上傳到layer-test儲存庫。

```
aws ecr initiate-layer-upload \
    --repository-name layer-test
```

輸出：

```
{
    "partSize": 10485760,
    "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 InitiateLayerUpload。

list-images

以下程式碼範例顯示如何使用 list-images。

AWS CLI

列出儲存庫中的映像

下列list-images範例顯示cluster-autoscaler儲存庫中的映像清單。

```
aws ecr list-images \
    --repository-name cluster-autoscaler
```

輸出：

```
{
    "imageIds": [
        {
            "imageDigest":
 "sha256:99c6fb4377e9a420a1eb3b410a951c9f464eff3b7dbc76c65e434e39b94b6570",
            "imageTag": "v1.13.8"
        },
        {
            "imageDigest":
 "sha256:99c6fb4377e9a420a1eb3b410a951c9f464eff3b7dbc76c65e434e39b94b6570",
            "imageTag": "v1.13.7"
        },
        {
            "imageDigest":
 "sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",
            "imageTag": "v1.13.6"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListImages。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出儲存庫的標籤

下列list-tags-for-resource範例顯示與hello-world儲存庫相關聯的標籤清單。

```
aws ecr list-tags-for-resource \
    --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world
```

輸出：

```
{
    "tags": [
        {
            "Key": "Stage",
            "Value": "Integ"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

put-image-scanning-configuration

以下程式碼範例顯示如何使用 put-image-scanning-configuration。

AWS CLI

更新儲存庫的映像掃描組態

下列put-image-scanning-configuration範例會更新指定儲存庫的映像掃描組態。

```
aws ecr put-image-scanning-configuration \
    --repository-name sample-repo \
    --image-scanning-configuration scanOnPush=true
```

輸出：

```
{
   "registryId": "012345678910",
   "repositoryName": "sample-repo",
   "imageScanningConfiguration": {
```

```
    "scanOnPush": true
  }
}
```

如需詳細資訊，請參閱《Amazon ECR 使用者指南》中的映像掃描。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutImageScanningConfiguration。

put-image-tag-mutability

以下程式碼範例顯示如何使用 put-image-tag-mutability。

AWS CLI

更新儲存庫的映像標籤可變性設定

下列put-image-tag-mutability範例會為標籤不可變性設定指定的儲存庫。這可防止覆寫儲存庫中的所有映像標籤。

```
aws ecr put-image-tag-mutability \
    --repository-name hello-repository \
    --image-tag-mutability IMMUTABLE
```

輸出：

```
{
  "registryId": "012345678910",
  "repositoryName": "sample-repo",
  "imageTagMutability": "IMMUTABLE"
}
```

如需詳細資訊，請參閱《Amazon ECR 使用者指南》中的映像標籤可變更性。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutImageTagMutability。

put-image

以下程式碼範例顯示如何使用 put-image。

AWS CLI

使用其資訊清單重新標記映像

下列put-image範例會在具有現有映像資訊清單的hello-world儲存庫中建立新的標籤。

```
aws ecr put-image \
    --repository-name hello-world \
    --image-tag 2019.08 \
    --image-manifest file://hello-world.manifest.json
```

hello-world.manifest.json 的內容：

```
{
    "schemaVersion": 2,
    "mediaType": "application/vnd.docker.distribution.manifest.v2+json",
    "config": {
        "mediaType": "application/vnd.docker.container.image.v1+json",
        "size": 5695,
        "digest":
 "sha256:cea5fe7701b7db3dd1c372f3cea6f43cdda444fcc488f530829145e426d8b980"
    },
    "layers": [
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 39096921,
            "digest":
 "sha256:d8868e50ac4c7104d2200d42f432b661b2da8c1e417ccfae217e6a1e04bb9295"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 57938,
            "digest":
 "sha256:83251ac64627fc331584f6c498b3aba5badc01574e2c70b2499af3af16630eed"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 423,
            "digest":
 "sha256:589bba2f1b36ae56f0152c246e2541c5aa604b058febfcf2be32e9a304fec610"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 680,
            "digest":
 "sha256:d62ecaceda3964b735cdd2af613d6bb136a52c1da0838b2ff4b4dab4212bcb1c"
        },
```

```
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 162,
            "digest":
"sha256:6d93b41cfc6bf0d2522b7cf61588de4cd045065b36c52bd3aec2ba0622b2b22b"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 28268840,
            "digest":
"sha256:6986b4d4c07932c680b3587f2eac8b0e013568c003cc23b04044628a5c5e599f"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 35369152,
            "digest":
"sha256:8c5ec60f10102dc8da0649d866c7c2f706e459d0bdc25c83ad2de86f4996c276"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 155,
            "digest":
"sha256:cde50b1c594539c5f67cbede9aef95c9ae321ccfb857f7b251b45b84198adc85"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 28737,
            "digest":
"sha256:2e102807ab72a73fc9abf53e8c50e421bdc337a0a8afcb242176edeec65977e4"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 190,
            "digest":
"sha256:fc379bbd5ed37808772bef016553a297356c59b8f134659e6ee4ecb563c2f5a7"
        },
        {
            "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
            "size": 28748,
            "digest":
"sha256:021db240dfccf5a1aff19507d17c0177e5888e518acf295b52204b1825e8b7ee"
        }
    ]
```

```
}
```

輸出：

```
{
    "image": {
        "registryId": "130757420319",
        "repositoryName": "hello-world",
        "imageId": {
            "imageDigest":
 "sha256:8ece96b74f87652876199d83bd107d0435a196133af383ac54cb82b6cc5283ae",
            "imageTag": "2019.08"
        },
        "imageManifest": "{\n  \"schemaVersion\": 2,\n  \"mediaType
\": \"application/vnd.docker.distribution.manifest.v2+json
\",\n  \"config\": {\n    \"mediaType\": \"application/
vnd.docker.container.image.v1+json\",\n    \"size\": 5695,\n    \"digest\":
 \"sha256:cea5fe7701b7db3dd1c372f3cea6f43cdda444fcc488f530829145e426d8b980\"\n
  },\n  \"layers\": [\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 39096921,\n      \"digest
\": \"sha256:d8868e50ac4c7104d2200d42f432b661b2da8c1e417ccfae217e6a1e04bb9295\"\n
  },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 57938,\n      \"digest
\": \"sha256:83251ac64627fc331584f6c498b3aba5badc01574e2c70b2499af3af16630eed
\"\n    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 423,\n      \"digest\":
 \"sha256:589bba2f1b36ae56f0152c246e2541c5aa604b058febfcf2be32e9a304fec610\"\n    },
\n    {\n      \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",
\n      \"size\": 680,\n      \"digest\":
 \"sha256:d62ecaceda3964b735cdd2af613d6bb136a52c1da0838b2ff4b4dab4212bcb1c
\"\n    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 162,\n      \"digest
\": \"sha256:6d93b41cfc6bf0d2522b7cf61588de4cd045065b36c52bd3aec2ba0622b2b22b
\"\n    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 28268840,\n      \"digest
\": \"sha256:6986b4d4c07932c680b3587f2eac8b0e013568c003cc23b04044628a5c5e599f
\"\n    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 35369152,\n      \"digest
\": \"sha256:8c5ec60f10102dc8da0649d866c7c2f706e459d0bdc25c83ad2de86f4996c276\"\n
  },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 155,\n      \"digest\":
 \"sha256:cde50b1c594539c5f67cbede9aef95c9ae321ccfb857f7b251b45b84198adc85\"\n    },
\n    {\n      \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",
```

```
\n      \"size\": 28737,\n      \"digest\":
 \"sha256:2e102807ab72a73fc9abf53e8c50e421bdc337a0a8afcb242176edeec65977e4\"\n    },
\n    {\n      \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",
\n      \"size\": 190,\n      \"digest\":
 \"sha256:fc379bbd5ed37808772bef016553a297356c59b8f134659e6ee4ecb563c2f5a7\"\n    },
\n    {\n      \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",
\n      \"size\": 28748,\n      \"digest\":
 \"sha256:021db240dfccf5a1aff19507d17c0177e5888e518acf295b52204b1825e8b7ee\"\n
  }\n  ]\n}\n"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutImage。

put-lifecycle-policy

以下程式碼範例顯示如何使用 put-lifecycle-policy。

AWS CLI

建立生命週期政策

下列put-lifecycle-policy範例會在 帳戶的預設登錄檔中為指定的儲存庫建立生命週期政策。

```
aws ecr put-lifecycle-policy \
    --repository-name "project-a/amazon-ecs-sample" \
    --lifecycle-policy-text "file://policy.json"
```

policy.json 的內容：

```
{
   "rules": [
       {
           "rulePriority": 1,
           "description": "Expire images older than 14 days",
           "selection": {
               "tagStatus": "untagged",
               "countType": "sinceImagePushed",
               "countUnit": "days",
               "countNumber": 14
           },
           "action": {
```

```
                "type": "expire"
            }
        }
    ]
}
```

輸出：

```
{
    "registryId": "<aws_account_id>",
    "repositoryName": "project-a/amazon-ecs-sample",
    "lifecyclePolicyText": "{\"rules\":[{\"rulePriority\":1,\"description\":\"Expire
 images older than 14 days\",\"selection\":{\"tagStatus\":\"untagged\",\"countType
\":\"sinceImagePushed\",\"countUnit\":\"days\",\"countNumber\":14},\"action\":
{\"type\":\"expire\"}}]}"
}
```

如需詳細資訊，請參閱《Amazon ECR 使用者指南》中的生命週期政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutLifeCyclePolicy。

set-repository-policy

以下程式碼範例顯示如何使用 set-repository-policy。

AWS CLI

設定儲存庫的儲存庫政策

下列set-repository-policy範例會將檔案中包含的儲存庫政策連接至cluster-autoscaler儲存庫。

```
aws ecr set-repository-policy \
    --repository-name cluster-autoscaler \
    --policy-text file://my-policy.json
```

my-policy.json 的內容：

```
{
    "Version" : "2008-10-17",
    "Statement" : [
```

```
        {
            "Sid" : "allow public pull",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : [
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ]
        }
    ]
}
```

輸出：

```
{
    "registryId": "012345678910",
    "repositoryName": "cluster-autoscaler",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
 \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" :
 \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage
\", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetRepositoryPolicy。

start-image-scan

以下程式碼範例顯示如何使用 start-image-scan。

AWS CLI

啟動映像漏洞掃描

下列start-image-scan範例會啟動 的映像掃描，並由指定儲存庫中的映像摘要指定。

```
aws ecr start-image-scan \
    --repository-name sample-repo \
    --image-
id imageDigest=sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6
```

輸出：

```
{
   "registryId": "012345678910",
   "repositoryName": "sample-repo",
   "imageId": {
       "imageDigest":
 "sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6"
   },
   "imageScanStatus": {
       "status": "IN_PROGRESS"
   }
}
```

如需詳細資訊，請參閱《Amazon ECR 使用者指南》中的映像掃描。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartImageScan。

start-lifecycle-policy-preview

以下程式碼範例顯示如何使用 start-lifecycle-policy-preview。

AWS CLI

建立生命週期政策預覽

下列start-lifecycle-policy-preview範例會為指定的儲存庫建立由 JSON 檔案定義的生命週期政策預覽。

```
aws ecr start-lifecycle-policy-preview \
    --repository-name "project-a/amazon-ecs-sample" \
    --lifecycle-policy-text "file://policy.json"
```

policy.json 的內容：

```
{
   "rules": [
       {
           "rulePriority": 1,
           "description": "Expire images older than 14 days",
           "selection": {
               "tagStatus": "untagged",
               "countType": "sinceImagePushed",
               "countUnit": "days",
```

```
                "countNumber": 14
            },
            "action": {
                "type": "expire"
            }
        }
    ]
}
```

輸出：

```
{
    "registryId": "012345678910",
    "repositoryName": "project-a/amazon-ecs-sample",
    "lifecyclePolicyText": "{\n    \"rules\": [\n        {\n
 \"rulePriority\": 1,\n            \"description\": \"Expire images older than 14
 days\",\n            \"selection\": {\n                \"tagStatus\": \"untagged\",
\n                \"countType\": \"sinceImagePushed\",\n                \"countUnit
\": \"days\",\n                \"countNumber\": 14\n            },\n
 \"action\": {\n                \"type\": \"expire\"\n            }\n        }\n
  ]\n}\n",
    "status": "IN_PROGRESS"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartLifecyclePolicyPreview。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

標記儲存庫

下列tag-resource範例會設定儲存hello-world庫Integ上具有索引鍵Stage和值的標籤。

```
aws ecr tag-resource \
    --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world \
    --tags Key=Stage,Value=Integ
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

取消標記儲存庫

下列untag-resource範例Stage會從hello-world儲存庫移除具有 金鑰的標籤。

```
aws ecr untag-resource \
    --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world \
    --tag-keys Stage
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

upload-layer-part

以下程式碼範例顯示如何使用 upload-layer-part。

AWS CLI

上傳圖層部分

以下會將映像層部分upload-layer-part上傳至layer-test儲存庫。

```
aws ecr upload-layer-part \
    --repository-name layer-test \
    --upload-id 6cb64b8a-9378-0e33-2ab1-b780fab8a9e9 \
    --part-first-byte 0 \
    --part-last-byte 8323314 \
    --layer-part-blob file:///var/lib/docker/image/overlay2/layerdb/sha256/
ff986b10a018b48074e6d3a68b39aad8ccc002cdad912d4148c0f92b3729323e/layer.b64
```

輸出：

```
{
    "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9",
```

```
    "registryId": "012345678910",
    "lastByteReceived": 8323314,
    "repositoryName": "layer-test"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UploadLayerPart。

使用 的 Amazon ECR 公有範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon ECR Public 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`batch-delete-image`

以下程式碼範例顯示如何使用 `batch-delete-image`。

AWS CLI

範例 1：若要使用映像摘要 ID 刪除映像，映像及其所有標籤都會在公有登錄檔的儲存庫內刪除

下列`batch-delete-image`範例會透過指定映像摘要來刪除映像：

```
aws ecr-public batch-delete-image \
    --repository-name project-a/nginx-web-app \
    --image-
ids imageDigest=sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2
```

輸出：

```
{
```

```
"imageIds": [
    {
        "imageDigest":
 "sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2",
        "imageTag": "latest"
    }
],
"failures": []
}
```

如需詳細資訊，請參閱《Amazon ECR 公有使用者指南》中的刪除公有儲存庫中的映像。

範例 2：指定與您要從儲存庫刪除之映像相關聯的標籤，以刪除任何映像。

下列batch-delete-image範例透過指定與公有登錄檔project-a/nginx-web-app中名為 的映像儲存庫相關聯的標籤來刪除映像。如果您只有一個標籤並執行此命令，則會移除映像。否則，如果您有相同映像的多個標籤，請指定一個標籤，而且只會從儲存庫移除該標籤，而不會移除映像。

```
aws ecr-public batch-delete-image \
    --repository-name project-a/nginx-web-app \
    --image-ids imageTag=_temp
```

輸出：

```
{
    "imageIds": [
        {
            "imageDigest":
 "sha256:f7a86a0760e2f8d7eff07e515fc87bf4bac45c35376c06f9a280f15ecad6d7e0",
            "imageTag": "_temp"
        }
    ],
    "failures": []
}
```

如需詳細資訊，請參閱《Amazon ECR 公有使用者指南》中的刪除公有儲存庫中的映像。

範例 3：若要刪除多個映像，您可以在公有登錄檔的儲存庫請求中指定多個映像標籤或映像摘要。

下列batch-delete-image範例透過在請求中指定多個映像標籤或映像摘要，從名為 project-a/nginx-web-app 的儲存庫中刪除多個映像。

```
aws ecr-public batch-delete-image \
    --repository-name project-a/nginx-web-app \
    --image-ids imageTag=temp2.0
  imageDigest=sha256:47ba980bc055353d9c0af89b1894f68faa43ca93856917b8406316be86f01278
```

輸出：

```
{
     "imageIds": [
         {
             "imageDigest":
 "sha256:47ba980bc055353d9c0af89b1894f68faa43ca93856917b8406316be86f01278"
         },
         {
             "imageDigest":
 "sha256:f7a86a0760e2f8d7eff07e515fc87bf4bac45c35376c06f9a280f15ecad6d7e0",
             "imageTag": "temp2.0"
         }
     ],
     "failures": []
 }
```

如需詳細資訊，請參閱《Amazon ECR 公有使用者指南》中的刪除公有儲存庫中的映像。

範例 4：若要使用登錄 ID 和映像摘要 ID 刪除跨 AWS 帳戶中的映像，會在公有登錄檔的儲存庫內刪除映像及其所有標籤

下列batch-delete-image範例透過在跨 AWS 帳戶中指定映像摘要來刪除映像：

```
aws ecr-public batch-delete-image \
    --registry-id 123456789098 \
    --repository-name project-a/nginx-web-app \
    --image-
ids imageDigest=sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2
 \
    --region us-east-1
```

輸出：

```
{
    "imageIds": [
```

```
        {
            "imageDigest":
 "sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2",
            "imageTag": "temp2.0"
        }
    ],
    "failures": []
}
```

如需詳細資訊，請參閱《Amazon ECR 公有使用者指南》中的刪除公有儲存庫中的映像。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDeleteImage。

create-repository

以下程式碼範例顯示如何使用 create-repository。

AWS CLI

範例 1：在公有登錄檔中建立儲存庫

下列create-repository範例會在公有登錄檔中建立名為 project-a/nginx-web-app 的儲存庫。

```
aws ecr-public create-repository \
    --repository-name project-a/nginx-web-app
```

輸出：

```
{
    "repository": {
        "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/
nginx-web-app",
        "registryId": "123456789012",
        "repositoryName": "project-a/nginx-web-app",
        "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/
nginx-web-app",
        "createdAt": "2024-07-01T21:08:55.131000+00:00"
    },
    "catalogData": {}
}
```

如需詳細資訊，請參閱《Amazon ECR 公有使用者指南》中的建立公有儲存庫。

範例 2：在公有登錄檔中建立儲存庫，並簡短說明儲存庫中映像相容的儲存庫內容、系統和操作架構

下列create-repository範例會在公有登錄檔中建立名為 project-a/nginx-web-app 的儲存庫，其中簡短說明儲存庫中映像相容的儲存庫、系統和操作架構內容。

```
aws ecr-public create-repository \
    --repository-name project-a/nginx-web-app \
    --catalog-data 'description=My project-a ECR Public
 Repository,architectures=ARM,ARM 64,x86,x86-64,operatingSystems=Linux'
```

輸出：

```
{
    "repository": {
        "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/
nginx-web-app",
        "registryId": "123456789012",
        "repositoryName": "project-a/nginx-web-app",
        "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/
nginx-web-app",
        "createdAt": "2024-07-01T21:23:20.455000+00:00"
    },
    "catalogData": {
        "description": "My project-a ECR Public Repository",
        "architectures": [
            "ARM",
            "ARM 64",
            "x86",
            "x86-64"
        ],
        "operatingSystems": [
            "Linux"
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon ECR 公有使用者指南》中的建立公有儲存庫。

範例 3：在公有登錄檔中建立儲存庫，以及 logoImageBlob、 aboutText、 usageText 和標籤資訊

下列create-repository範例會在公有登錄檔中建立名為 project-a/nginx-web-app 的儲存庫，以及 logoImageBlob、 aboutText、 usageText 和標籤資訊。

```
aws ecr-public create-repository \
    --cli-input-json file://myfile.json
```

myfile.json 的內容：

```
{
    "repositoryName": "project-a/nginx-web-app",
    "catalogData": {
        "description": "My project-a ECR Public Repository",
        "architectures": [
            "ARM",
            "ARM 64",
            "x86",
            "x86-64"
        ],
        "operatingSystems": [
            "Linux"
        ],
        "logoImageBlob": "iVBORw0KGgoA<<truncated-for-better-reading>>ErkJggg==",
        "aboutText": "## Quick reference\n\nMaintained by: [the Amazon Linux Team]
(https://github.com/aws/amazon-linux-docker-images)\n\nWhere to get help: [the
 Docker Community Forums](https://forums.docker.com/), [the Docker Community Slack]
(https://dockr.ly/slack), or [Stack Overflow](https://stackoverflow.com/search?
tab=newest&q=docker)\n\n## Supported tags and respective `dockerfile` links\n\n*
 [`2.0.20200722.0`, `2`, `latest`](https://github.com/amazonlinux/container-images/
blob/03d54f8c4d522bf712cffd6c8f9aafba0a875e78/Dockerfile)\n* [`2.0.20200722.0-
with-sources`, `2-with-sources`, `with-sources`](https://github.com/
amazonlinux/container-images/blob/1e7349845e029a2e6afe6dc473ef17d052e3546f/
Dockerfile)\n* [`2018.03.0.20200602.1`, `2018.03`, `1`](https://github.com/
amazonlinux/container-images/blob/f10932e08c75457eeb372bf1cc47ea2a4b8e98c8/
Dockerfile)\n* [`2018.03.0.20200602.1-with-sources`, `2018.03-with-sources`,
 `1-with-sources`](https://github.com/amazonlinux/container-images/
blob/8c9ee491689d901aa72719be0ec12087a5fa8faf/Dockerfile)\n\n## What is Amazon
 Linux?\n\nAmazon Linux is provided by Amazon Web Services (AWS). It is designed
 to provide a stable, secure, and high-performance execution environment for
 applications running on Amazon EC2. The full distribution includes packages that
 enable easy integration with AWS, including launch configuration tools and many
 popular AWS libraries and tools. AWS provides ongoing security and maintenance
 updates to all instances running Amazon Linux.\n\nThe Amazon Linux container image
 contains a minimal set of packages. To install additional packages, [use `yum`]
(https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/managing-software.html).\n\nAWS
 provides two versions of Amazon Linux: [Amazon Linux 2](https://aws.amazon.com/
amazon-linux-2/) and [Amazon Linux AMI](https://aws.amazon.com/amazon-linux-ami/).
```

\n\nFor information on security updates for Amazon Linux, please refer to [Amazon Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html) and [Amazon Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note that Docker Hub's vulnerability scanning for Amazon Linux is currently based on RPM versions, which does not reflect the state of backported patches for vulnerabilities.\n\n## Where can I run Amazon Linux container images?\n\nYou can run Amazon Linux container images in any Docker based environment. Examples include, your laptop, in Amazon EC2 instances, and Amazon ECS clusters.\n\n## License\n\nAmazon Linux is available under the [GNU General Public License, version 2.0](https://github.com/aws/amazon-linux-docker-images/blob/master/LICENSE). Individual software packages are available under their own licenses; run `rpm -qi [package name]` or check `/usr/share/doc/[package name]-*` and `/usr/share/licenses/[package name]-*` for details.\n\nAs with all Docker images, these likely also contain other software which may be under other licenses (such as Bash, etc from the base distribution, along with any direct or indirect dependencies of the primary software being contained).\n\nSome additional license information which was able to be auto-detected might be found in [the `repo-info` repository's `amazonlinux/` directory](https://github.com/docker-library/repo-info/tree/master/repos/amazonlinux).\n\n## Security\n\nFor information on security updates for Amazon Linux, please refer to [Amazon Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html) and [Amazon Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note that Docker Hub's vulnerability scanning for Amazon Linux is currently based on RPM versions, which does not reflect the state of backported patches for vulnerabilities.",
       "usageText": "## Supported architectures\n\namd64, arm64v8\n\n## Where can I run Amazon Linux container images?\n\nYou can run Amazon Linux container images in any Docker based environment. Examples include, your laptop, in Amazon EC2 instances, and ECS clusters.\n\n## How do I install a software package from Extras repository in Amazon Linux 2?\n\nAvailable packages can be listed with the `amazon-linux-extras` command. Packages can be installed with the `amazon-linux-extras install <package>` command. Example: `amazon-linux-extras install rust1`\n\n## Will updates be available for Amazon Linux containers?\n\nSimilar to the Amazon Linux images for Amazon EC2 and on-premises use, Amazon Linux container images will get ongoing updates from Amazon in the form of security updates, bug fix updates, and other enhancements. Security bulletins for Amazon Linux are available at https://alas.aws.amazon.com/\n\n## Will AWS Support the current version of Amazon Linux going forward?\n\nYes; in order to avoid any disruption to your existing applications and to facilitate migration to Amazon Linux 2, AWS will provide regular security updates for Amazon Linux 2018.03 AMI and container image for 2 years after the final LTS build is announced. You can also use all your existing support channels such as AWS Support and Amazon Linux Discussion Forum to continue to submit support requests."
    },
    "tags": [

```
        {
            "Key": "Name",
            "Value": "project-a/nginx-web-app"
        },
        {
            "Key": "Environment",
            "Value": "Prod"
        }
    ]
}
```

輸出：

```
{
    "repository": {
        "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/
nginx-web-app",
        "registryId": "123456789012",
        "repositoryName": "project-a/nginx-web-app",
        "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/
nginx-web-app",
        "createdAt": "2024-07-01T21:53:05.749000+00:00"
    },
    "catalogData": {
        "description": "My project-a ECR Public Repository",
        "architectures": [
            "ARM",
            "ARM 64",
            "x86",
            "x86-64"
        ],
        "operatingSystems": [
            "Linux"
        ],
        "logoUrl": "https://d3g9o9u8re44ak.cloudfront.net/
logo/23861450-4b9b-403c-9a4c-7aa0ef140bb8/2f9bf5a7-a32f-45b4-b5cd-c5770a35e6d7.png",
        "aboutText": "## Quick reference\n\nMaintained by: [the Amazon Linux Team]
(https://github.com/aws/amazon-linux-docker-images)\n\nWhere to get help: [the
 Docker Community Forums](https://forums.docker.com/), [the Docker Community Slack]
(https://dockr.ly/slack), or [Stack Overflow](https://stackoverflow.com/search?
tab=newest&q=docker)\n\n## Supported tags and respective `dockerfile` links\n\n*
 [`2.0.20200722.0`, `2`, `latest`](https://github.com/amazonlinux/container-images/
blob/03d54f8c4d522bf712cffd6c8f9aafba0a875e78/Dockerfile)\n* [`2.0.20200722.0-
```

```
with-sources`, `2-with-sources`, `with-sources`](https://github.com/
amazonlinux/container-images/blob/1e7349845e029a2e6afe6dc473ef17d052e3546f/
Dockerfile)\n* [`2018.03.0.20200602.1`, `2018.03`, `1`](https://github.com/
amazonlinux/container-images/blob/f10932e08c75457eeb372bf1cc47ea2a4b8e98c8/
Dockerfile)\n* [`2018.03.0.20200602.1-with-sources`, `2018.03-with-sources`,
 `1-with-sources`](https://github.com/amazonlinux/container-images/
blob/8c9ee491689d901aa72719be0ec12087a5fa8faf/Dockerfile)\n\n## What is Amazon
 Linux?\n\nAmazon Linux is provided by Amazon Web Services (AWS). It is designed
 to provide a stable, secure, and high-performance execution environment for
 applications running on Amazon EC2. The full distribution includes packages that
 enable easy integration with AWS, including launch configuration tools and many
 popular AWS libraries and tools. AWS provides ongoing security and maintenance
 updates to all instances running Amazon Linux.\n\nThe Amazon Linux container image
 contains a minimal set of packages. To install additional packages, [use `yum`]
(https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/managing-software.html).\n\nAWS
 provides two versions of Amazon Linux: [Amazon Linux 2](https://aws.amazon.com/
amazon-linux-2/) and [Amazon Linux AMI](https://aws.amazon.com/amazon-linux-ami/).
\n\nFor information on security updates for Amazon Linux, please refer to [Amazon
 Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html) and [Amazon
 Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note that Docker
 Hub's vulnerability scanning for Amazon Linux is currently based on RPM versions,
 which does not reflect the state of backported patches for vulnerabilities.\n
\n## Where can I run Amazon Linux container images?\n\nYou can run Amazon Linux
 container images in any Docker based environment. Examples include, your laptop,
 in Amazon EC2 instances, and Amazon ECS clusters.\n\n## License\n\nAmazon Linux is
 available under the [GNU General Public License, version 2.0](https://github.com/
aws/amazon-linux-docker-images/blob/master/LICENSE). Individual software packages
 are available under their own licenses; run `rpm -qi [package name]` or check
 `/usr/share/doc/[package name]-*` and `/usr/share/licenses/[package name]-*` for
 details.\n\nAs with all Docker images, these likely also contain other software
 which may be under other licenses (such as Bash, etc from the base distribution,
 along with any direct or indirect dependencies of the primary software being
 contained).\n\nSome additional license information which was able to be auto-
detected might be found in [the `repo-info` repository's `amazonlinux/` directory]
(https://github.com/docker-library/repo-info/tree/master/repos/amazonlinux).\n\n##
 Security\n\nFor information on security updates for Amazon Linux, please refer
 to [Amazon Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html)
 and [Amazon Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note
 that Docker Hub's vulnerability scanning for Amazon Linux is currently based
 on RPM versions, which does not reflect the state of backported patches for
 vulnerabilities.",
        "usageText": "## Supported architectures\n\namd64, arm64v8\n\n## Where
 can I run Amazon Linux container images?\n\nYou can run Amazon Linux container
 images in any Docker based environment. Examples include, your laptop, in Amazon
```

```
 EC2 instances, and ECS clusters.\n\n## How do I install a software package from
 Extras repository in Amazon Linux 2?\n\nAvailable packages can be listed with the
 `amazon-linux-extras` command. Packages can be installed with the `amazon-linux-
extras install <package>` command. Example: `amazon-linux-extras install rust1`\n
\n## Will updates be available for Amazon Linux containers?\n\nSimilar to the Amazon
 Linux images for Amazon EC2 and on-premises use, Amazon Linux container images will
 get ongoing updates from Amazon in the form of security updates, bug fix updates,
 and other enhancements. Security bulletins for Amazon Linux are available at
 https://alas.aws.amazon.com/\n\n## Will AWS Support the current version of Amazon
 Linux going forward?\n\nYes; in order to avoid any disruption to your existing
 applications and to facilitate migration to Amazon Linux 2, AWS will provide
 regular security updates for Amazon Linux 2018.03 AMI and container image for 2
 years after the final LTS build is announced. You can also use all your existing
 support channels such as AWS Support and Amazon Linux Discussion Forum to continue
 to submit support requests."
    }
}
```

如需詳細資訊，請參閱《Amazon ECR 公有使用者指南》中的建立公有儲存庫和《Amazon ECR 公有使用者指南》中的儲存庫目錄資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRepository。

delete-repository-policy

以下程式碼範例顯示如何使用 delete-repository-policy。

AWS CLI

在公有登錄檔中刪除儲存庫政策

下列delete-repository-policy範例會刪除您 AWS 帳戶中 ECR Public 儲存庫的儲存庫政策。

```
aws ecr-public delete-repository-policy \
     --repository-name project-a/nginx-web-app \
     --region us-east-1
```

輸出：

```
{
    "registryId": "123456789012",
    "repositoryName": "project-a/nginx-web-app",
```

```
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
    \"Sid\" : \"AllowPush\",\n    \"Effect\" : \"Allow\",\n    \"Principal
\" : {\n      \"AWS\" : [ \"arn:aws:iam:"123456789012":user/eksuser1\",
 \"arn:aws:iam:"123456789012":user/admin\" ]\n    },\n    \"Action\" :
 [ \"ecr-public:BatchCheckLayerAvailability\", \"ecr-public:PutImage\",
 \"ecr-public:InitiateLayerUpload\", \"ecr-public:UploadLayerPart\", \"ecr-
public:CompleteLayerUpload\" ]\n  } ]\n}"
}
```

如需詳細資訊，請參閱《Amazon ECR 公有使用者指南》中的刪除公有儲存庫政策陳述式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRepositoryPolicy。

delete-repository

以下程式碼範例顯示如何使用 delete-repository。

AWS CLI

刪除公有登錄檔中的儲存庫

下列delete-repository範例project-a/nginx-web-app會從公有登錄檔中刪除名為 的儲存庫。

```
aws ecr-public delete-repository \
    --repository-name project-a/nginx-web-app
```

輸出：

```
{
    "repository": {
        "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/
nginx-web-app",
        "registryId": "123456789012",
        "repositoryName": "project-a/nginx-web-app",
        "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/
nginx-web-app",
        "createdAt": "2024-07-01T22:14:50.103000+00:00"
    }
}
```

如需詳細資訊，請參閱 Amazon ECR Public 中的刪除公有儲存庫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRepository。

describe-image-tags

以下程式碼範例顯示如何使用 describe-image-tags。

AWS CLI

範例 1：描述公有儲存庫中的映像標籤詳細資訊

下列describe-image-tags範例說明project-a/nginx-web-app範例儲存庫中的映像標籤。

```
aws ecr-public describe-image-tags \
    --repository-name project-a/nginx-web-app \
    --region us-east-1
```

輸出：

```
{
    "imageTagDetails": [
        {
            "imageTag": "latest",
            "createdAt": "2024-07-10T22:29:00-05:00",
            "imageDetail": {
                "imageDigest":
 "sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2",
                "imageSizeInBytes": 121956548,
                "imagePushedAt": "2024-07-10T22:29:00-05:00",
                "imageManifestMediaType": "application/
vnd.docker.distribution.manifest.v2+json",
                "artifactMediaType": "application/
vnd.docker.container.image.v1+json"
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeImageTags。

describe-images

以下程式碼範例顯示如何使用 describe-images。

AWS CLI

範例 1：描述公有登錄檔儲存庫中的映像

下列describe-images範例說明公有登錄檔中名為 的儲存庫project-a/nginx-web-app中的 imagesDetails。

```
aws ecr-public describe-images \
    --repository-name project-a/nginx-web-app \
    --region us-east-1
```

輸出：

```
{
    "imageDetails": [
        {
            "registryId": "123456789012",
            "repositoryName": "project-a/nginx-web-app",
            "imageDigest":
 "sha256:0d8c93e72e82fa070d49565c00af32abbe8ddfd7f75e39f4306771ae0628c7e8",
            "imageTags": [
                "temp1.0"
            ],
            "imageSizeInBytes": 123184716,
            "imagePushedAt": "2024-07-23T11:32:49-05:00",
            "imageManifestMediaType": "application/
vnd.docker.distribution.manifest.v2+json",
            "artifactMediaType": "application/vnd.docker.container.image.v1+json"
        },
        {
            "registryId": "123456789012",
            "repositoryName": "project-a/nginx-web-app",
            "imageDigest":
 "sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2",
            "imageTags": [
                "temp2.0"
            ],
            "imageSizeInBytes": 121956548,
            "imagePushedAt": "2024-07-23T11:39:38-05:00",
            "imageManifestMediaType": "application/
vnd.docker.distribution.manifest.v2+json",
            "artifactMediaType": "application/vnd.docker.container.image.v1+json"
        },
```

```
        {
            "registryId": "123456789012",
            "repositoryName": "project-a/nginx-web-app",
            "imageDigest":
 "sha256:f7a86a0760e2f8d7eff07e515fc87bf4bac45c35376c06f9a280f15ecad6d7e0",
            "imageTags": [
                "temp3.0",
                "latest"
            ],
            "imageSizeInBytes": 232108879,
            "imagePushedAt": "2024-07-22T00:54:34-05:00",
            "imageManifestMediaType": "application/
vnd.docker.distribution.manifest.v2+json",
            "artifactMediaType": "application/vnd.docker.container.image.v1+json"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon ECR Public》中的描述公有儲存庫中的映像。

範例 2：依排序 imageTags & imagePushedAt 描述儲存庫中的映像

下列describe-images範例說明公有登錄檔中名為 project-a/nginx-web-app 的儲存庫中的映像。

```
aws ecr-public describe-images \
    --repository-name project-a/nginx-web-app \
    --query 'sort_by(imageDetails,& imagePushedAt)[*].imageTags[*]' \
    --output text
```

輸出：

```
temp3.0 latest
temp1.0
temp2.0
```

範例 3：描述來自儲存庫的映像，以產生在儲存庫中推送的最後 2 個映像標籤

下列describe-images範例會從公有登錄檔project-a/nginx-web-app中名為 的儲存庫取得映像標籤詳細資訊，並查詢結果以僅顯示前兩個記錄。

```
aws ecr-public describe-images \
```

```
    --repository-name project-a/nginx-web-app  \
    --query 'sort_by(imageDetails,& imagePushedAt)[*].imageTags[*] | [0:2]' \
    --output text
```

輸出：

```
temp3.0 latest
temp1.0
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeImages。

describe-registries

以下程式碼範例顯示如何使用 describe-registries。

AWS CLI

描述公有登錄檔中的所有登錄檔

下列describe-registries範例說明 帳戶中的所有登錄檔。

```
aws ecr-public describe-registries
```

輸出：

```
{
 "registries": [
     {
         "registryId": "123456789012",
         "registryArn": "arn:aws:ecr-public::123456789012:registry/123456789012",
         "registryUri": "public.ecr.aws/publicregistrycustomalias",
         "verified": false,
         "aliases": [
             {
                 "name": "publicregistrycustomalias",
                 "status": "ACTIVE",
                 "primaryRegistryAlias": true,
                 "defaultRegistryAlias": true
             }
         ]
     }
```

```
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRegistries。

describe-repository

以下程式碼範例顯示如何使用 describe-repository。

AWS CLI

範例 1：描述公有登錄檔中的儲存庫

下列describe-repositories範例說明公有登錄檔project-a/nginx-web-app中名為 的儲存庫。

```
aws ecr-public describe-repositories \
    --repository-name project-a/nginx-web-app
```

輸出：

```
{
    "repositories": [
        {
            "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/
nginx-web-app",
            "registryId": "123456789012",
            "repositoryName": "project-a/nginx-web-app",
            "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/
nginx-web-app",
            "createdAt": "2024-07-07T00:07:56.526000-05:00"
        }
    ]
}
```

範例 2：描述資料表中公有登錄檔中的所有儲存庫

下列describe-repositories範例說明公有登錄檔中的所有儲存庫，然後將儲存庫名稱輸出為資料表格式。

```
aws ecr-public describe-repositories \
```

```
    --region us-east-1 \
    --output table \
    --query "repositories[*].repositoryName"
```

輸出：

```
----------------------------
|   DescribeRepositories    |
+---------------------------+
|  project-a/nginx-web-app  |
|  nginx                    |
|  myfirstrepo1             |
|  helm-test-chart          |
|  test-ecr-public          |
|  nginx-web-app            |
|  sample-repo              |
+---------------------------+
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRepository。

get-authorization-token

以下程式碼範例顯示如何使用 get-authorization-token。

AWS CLI

範例 1：擷取 IAM 主體可存取之任何 Amazon ECR 公有登錄檔的授權字符

下列get-authorization-token範例會使用 CLI AWS 取得授權字符，並將其設定為 環境變數。

```
aws ecr-public get-authorization-token \
    --region us-east-1
```

輸出：

```
{
    "authorizationData": {
        "authorizationToken":
 "QVdTOmV5SndZWGxzYjJKJFHDSFKJHERWUY65IOU36TRYEGFNSDLRIUOTUYTHJKLDFGOcmFUQk9OSFV2UVV4a0x6Sm1
        "expiresAt": "2024-07-25T21:37:26.301000-04:00"
```

```
    }
}
```

如需詳細資訊，請參閱 Amazon ECR Public 中的 Amazon ECR 公有登錄檔。

範例 2：擷取 IAM 主體可存取之任何 Amazon ECR 公有登錄檔的授權字符

下列get-authorization-token範例會使用 CLI AWS 取得授權字符，並將其設定為 環境變數。

```
aws ecr-public get-authorization-token \
    --region us-east-1 \
    --output=text \
    --query 'authorizationData.authorizationToken'
```

輸出：

```
QVdTOmV5SndZWGxzYjJKJFHDSFKJHERWUY65IOU36TRYEGFNSDLRIUOTUYTHJKLDFGOcmFUQk9OSFV2UVV4a0x6Sm1ZV
```

如需詳細資訊，請參閱 Amazon ECR Public 中的 Amazon ECR 公有登錄檔。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAuthorizationToken。

get-login-password

以下程式碼範例顯示如何使用 get-login-password。

AWS CLI

範例 1：向 Amazon ECR 公有登錄檔驗證 Docker

下列get-login-password範例使用 GetAuthorizationToken API 擷取和顯示身分驗證字符，您可以使用該 API 向 Amazon ECR 公有登錄檔進行身分驗證。

```
aws ecr-public get-login-password \
    --region us-
east-1
| docker login \
    --username AWS \
    --password-stdin public.ecr.aws
```

此命令不會在終端機中產生輸出，而是將輸出輸送至 Docker。

如需詳細資訊，請參閱 Amazon ECR Public 中的驗證公有登錄檔。

範例 2：向您自己的自訂 AmazonECR 公有登錄檔驗證 Docker

下列get-login-password範例使用 GetAuthorizationToken API 擷取並顯示身分驗證字符，您可以使用該 API 向您自己的自訂 Amazon ECR 公有登錄檔進行身分驗證。

```
 aws ecr-public get-login-password \
    --region us-east-1 \
| docker login \
    --username AWS \
    --password-stdin public.ecr.aws/<your-public-registry-custom-alias>
```

此命令不會在終端機中產生輸出，而是將輸出輸送到 Docker。

如需詳細資訊，請參閱《Amazon ECR Public》中的驗證您自己的 Amazon ECR Public。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLoginPassword。

get-registry-catalog-data

以下程式碼範例顯示如何使用 get-registry-catalog-data。

AWS CLI

擷取公有 ECR 登錄檔的目錄中繼資料

以下內容會get-registry-catalog-data擷取 ECR 公有登錄檔的目錄中繼資料。

```
aws ecr-public get-registry-catalog-data \
    --region us-east-1
```

輸出：

```
{
    "registryCatalogData": {
        "displayName": "YourCustomPublicRepositoryalias"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRegistryCatalogData。

get-repository-catalog-data

以下程式碼範例顯示如何使用 get-repository-catalog-data。

AWS CLI

在公有登錄檔中擷取儲存庫的目錄中繼資料

下列get-repository-catalog-data範例列出公有登錄project-a/nginx-web-app檔中儲存庫的目錄中繼資料。

```
aws ecr-public get-repository-catalog-data \
    --repository-name project-a/nginx-web-app \
    --region us-east-1
```

輸出：

```
{
    "catalogData": {
        "description": "My project-a ECR Public Repository",
        "architectures": [
            "ARM",
            "ARM 64",
            "x86",
            "x86-64"
        ],
        "operatingSystems": [
            "Linux"
        ],
        "logoUrl": "https://d3g9o9u8re44ak.cloudfront.net/logo/491d3846-8f33-4d8b-
a10c-c2ce271e6c0d/4f09d87c-2569-4916-a932-5c296bf6f88a.png",
        "aboutText": "## Quick reference\n\nMaintained <truncated>",
        "usageText": "## Supported architectures\n\namd64, arm64v8\n\n##
 <truncated>"
    }
}
```

如需詳細資訊，請參閱 Amazon ECR Public 中的儲存庫目錄資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRepositoryCatalogData。

get-repository-policy

以下程式碼範例顯示如何使用 get-repository-policy。

AWS CLI

取得與儲存庫相關聯的儲存庫政策

下列get-repository-policy範例會取得與儲存庫相關聯的儲存庫政策。

```
aws ecr-public get-repository-policy \
    --repository-name project-a/nginx-web-app \
    --region us-east-1
```

輸出：

```
{
    "registryId": "123456789012",
    "repositoryName": "project-a/nginx-web-app",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
    \"Sid\" : \"AllowPush\",\n    \"Effect\" : \"Allow\",\n    \"Principal
\" : {\n      \"AWS\" : [ \"arn:aws:iam::123456789012:user/eksuser1\",
 \"arn:aws:iam::123456789012:user/admin\" ]\n    },\n    \"Action\" :
 [ \"ecr-public:BatchCheckLayerAvailability\", \"ecr-public:PutImage\",
 \"ecr-public:InitiateLayerUpload\", \"ecr-public:UploadLayerPart\", \"ecr-
public:CompleteLayerUpload\" ]\n  } ]\n}"
}
```

如需詳細資訊，請參閱《Amazon ECR 公有使用者指南》中的搭配 AWS SDK 或 CLI 使用 GetRepositoryPolicy。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRepositoryPolicy。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出公有登錄檔中公有儲存庫的標籤

下列list-tags-for-resource範例列出公有登錄檔project-a/nginx-web-app中名為 之資源的標籤。

```
aws ecr-public list-tags-for-resource \
    --resource-arn arn:aws:ecr-public::123456789012:repository/project-a/nginx-web-app \
    --region us-east-1
```

輸出：

```
{
    "tags": [
        {
            "Key": "Environment",
            "Value": "Prod"
        },
        {
            "Key": "stack",
            "Value": "dev1"
        },
        {
            "Key": "Name",
            "Value": "project-a/nginx-web-app"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon ECR Public》中的列出公有儲存庫的標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

put-registry-catalog-data

以下程式碼範例顯示如何使用 put-registry-catalog-data。

AWS CLI

建立或更新公有 ECR 登錄檔的目錄中繼資料

以下內容put-registry-catalog-data會建立或更新 ECR 公有登錄檔的目錄中繼資料。只有具有已驗證帳戶徽章的帳戶才能具有登錄檔顯示名稱。

```
aws ecr-public put-registry-catalog-data \
    --region us-east-1 \
    --display-name <YourCustomPublicRepositoryalias>
```

輸出：

```
{
    "registryCatalogData": {
        "displayName": "YourCustomPublicRepositoryalias"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutRegistryCatalogData。

put-repository-catalog-data

以下程式碼範例顯示如何使用 put-repository-catalog-data。

AWS CLI

在公有登錄檔中建立或更新儲存庫的目錄資料

下列put-repository-catalog-data範例會在公有登錄檔中建立或更新名為 project-a/nginx-web-app 的儲存庫目錄資料，以及 logoImageBlob、 aboutText、 usageText 和標籤資訊。

```
aws ecr-public put-repository-catalog-data \
    --repository-name project-a/nginx-web-app \
    --cli-input-json file://repository-catalog-data.json \
    --region us-east-1
```

repository-catalog-data.json 的內容：

```
{
    "repositoryName": "project-a/nginx-web-app",
    "catalogData": {
        "description": "My project-a ECR Public Repository",
        "architectures": [
            "ARM",
            "ARM 64",
            "x86",
            "x86-64"
        ],
        "operatingSystems": [
            "Linux"
        ],
        "logoImageBlob": "iVBORw0KGgoA<<truncated-for-better-reading>>ErkJggg==",
```

```
        "aboutText": "## Quick reference.",
        "usageText": "## Supported architectures are as follows."
    }
}
```

輸出：

```
{
    "catalogData": {
        "description": "My project-a ECR Public Repository",
        "architectures": [
            "ARM",
            "ARM 64",
            "x86",
            "x86-64"
        ],
        "operatingSystems": [
            "Linux"
        ],
        "logoUrl": "https://d3g9o9u8re44ak.cloudfront.net/logo/df86cf58-ee60-4061-
b804-0be24d97ccb1/4a9ed9b2-69e4-4ede-b924-461462d20ef0.png",
        "aboutText": "## Quick reference.",
        "usageText": "## Supported architectures are as follows."
    }
}
```

如需詳細資訊，請參閱 Amazon ECR Public 中的儲存庫目錄資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutRepositoryCatalogData。

set-repository-policy

以下程式碼範例顯示如何使用 set-repository-policy。

AWS CLI

範例 1：設定儲存庫政策以允許提取儲存庫

下列set-repository-policy範例會將 ECR 公有儲存庫政策套用至指定的儲存庫，以控制存取許可。

```
aws ecr-public set-repository-policy \
    --repository-name project-a/nginx-web-app \
```

```
    --policy-text file://my-repository-policy.json
```

my-repository-policy.json 的內容：

```
{
    "Version" : "2008-10-17",
    "Statement" : [
        {
            "Sid" : "allow public pull",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : [
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ]
        }
    ]
}
```

輸出：

```
{
    "registryId": "12345678901",
    "repositoryName": "project-a/nginx-web-app",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
 \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" :
 \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage
\", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
}
```

如需詳細資訊，請參閱《Amazon ECR 公有使用者指南》中的設定儲存庫政策陳述式。

範例 2：設定儲存庫政策，以允許帳戶中的 IAM 使用者推送映像

下列set-repository-policy範例允許您帳戶中的 IAM 使用者使用 ，使用file://my-repository-policy.json名為政策文字的輸入檔案，將映像推送至您 AWS 帳戶中的 ECR 儲存庫。

```
aws ecr-public set-repository-policy \
    --repository-name project-a/nginx-web-app \
    --policy-text file://my-repository-policy.json
```

my-repository-policy.json 的內容：

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "AllowPush",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::account-id:user/push-pull-user-1",
                    "arn:aws:iam::account-id:user/push-pull-user-2"
                ]
            },
            "Action": [
                "ecr-public:BatchCheckLayerAvailability",
                "ecr-public:PutImage",
                "ecr-public:InitiateLayerUpload",
                "ecr-public:UploadLayerPart",
                "ecr-public:CompleteLayerUpload"
            ]
        }
    ]
}
```

輸出：

```
{
    "registryId": "12345678901",
    "repositoryName": "project-a/nginx-web-app",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" :
 [ {\n    \"Sid\" : \"AllowPush\",\n    \"Effect\" : \"Allow\",\n
 \"Principal\" : {\n      \"AWS\" : [ \"arn:aws:iam::12345678901:user/admin
\", \"arn:aws:iam::12345678901:user/eksuser1\" ]\n    },\n    \"Action\" :
 [ \"ecr-public:BatchCheckLayerAvailability\", \"ecr-public:PutImage\",
 \"ecr-public:InitiateLayerUpload\", \"ecr-public:UploadLayerPart\", \"ecr-
public:CompleteLayerUpload\" ]\n  } ]\n}"
}
```

如需詳細資訊，請參閱《Amazon ECR 公有使用者指南》中的設定儲存庫政策陳述式。

範例 3：設定儲存庫政策以允許來自不同帳戶的 IAM 使用者推送映像

下列set-repository-policy範例允許特定帳戶在 AWS 您的帳戶中使用 cli 輸入檔案：//my-repository-policy.json 推送映像。

```
aws ecr-public set-repository-policy \
    --repository-name project-a/nginx-web-app \
    --policy-text file://my-repository-policy.json
```

my-repository-policy.json 的內容：

```
 {
     "Version": "2008-10-17",
     "Statement": [
         {
             "Sid": "AllowCrossAccountPush",
             "Effect": "Allow",
             "Principal": {
                 "AWS": "arn:aws:iam::other-or-same-account-id:role/RoleName"
             },
             "Action": [
                 "ecr-public:BatchCheckLayerAvailability",
                 "ecr-public:PutImage",
                 "ecr-public:InitiateLayerUpload",
                 "ecr-public:UploadLayerPart",
                 "ecr-public:CompleteLayerUpload"
             ]
         }
     ]
}
```

輸出：

```
{
    "registryId": "12345678901",
    "repositoryName": "project-a/nginx-web-app",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
 \"Sid\" : \"AllowCrossAccountPush\",\n    \"Effect\" : \"Allow\",\n    \"Principal
\" : {\n      \"AWS\" : \"arn:aws:iam::12345678901:role/RoleName\"\n    },\n
 \"Action\" : [ \"ecr-public:BatchCheckLayerAvailability\", \"ecr-public:PutImage
\", \"ecr-public:InitiateLayerUpload\", \"ecr-public:UploadLayerPart\", \"ecr-
public:CompleteLayerUpload\" ]\n  } ]\n}"
}
```

如需詳細資訊，請參閱《Amazon ECR 公有使用者指南》中的公有儲存庫政策範例。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetRepositoryPolicy。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

範例 1：在公有登錄檔中標記現有的公有儲存庫

下列tag-resource範例會在公有登錄檔project-a/nginx-web-app中標記名為 的儲存庫。

```
aws ecr-public tag-resource \
    --resource-arn arn:aws:ecr-public::123456789012:repository/project-a/nginx-web-app \
    --tags Key=stack,Value=dev \
    --region us-east-1
```

如需詳細資訊，請參閱 Amazon ECR Public 中的為公有儲存庫使用標籤。

範例 2：使用公有登錄檔中的多個標籤來標記現有的公有儲存庫。

下列tag-resource範例使用多個標籤來標記現有的儲存庫。

```
aws ecr-public tag-resource \
    --resource-arn arn:aws:ecr-public::890517186334:repository/project-a/nginx-web-app  \
    --tags Key=key1,Value=value1 Key=key2,Value=value2 Key=key3,Value=value3 \
    --region us-east-1
```

如需詳細資訊，請參閱 Amazon ECR Public 中的為公有儲存庫使用標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

範例 1：取消標記公有登錄檔中現有的公有儲存庫

下列untag-resource範例會在公有登錄檔project-a/nginx-web-app中標記名為 的儲存庫。

```
aws ecr-public untag-resource \
    --resource-arn arn:aws:ecr-public::123456789012:repository/project-a/nginx-web-app \
    --tag-keys stack \
    --region us-east-1
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Amazon ECR Public 中的為公有儲存庫使用標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

使用 的 Amazon ECS 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon ECS 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`capacity-provider-update`

以下程式碼範例顯示如何使用 capacity-provider-update。

AWS CLI

更新 ECS 叢集中的容量提供者

下列update-capacity-provider範例顯示如何修改 ECS 叢集中容量提供者的參數。

```
aws ecs update-capacity-provider \
```

```
    --name Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt \
    --auto-scaling-group-
provider "managedScaling={status=DISABLED,targetCapacity=50,minimumScalingStepSize=2,maximum
```

輸出：

```
{
    "capacityProvider": {
        "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt",
        "name": "Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt",
        "status": "ACTIVE",
        "autoScalingGroupProvider": {
            "autoScalingGroupArn": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:424941d1-b43f-4a17-
adbb-08b6a6e397e1:autoScalingGroupName/Infra-ECS-Cluster-ECS-project-update-cluster-
d6bb6d5b-ECSAutoScalingGroup-f44jrQHS2nRB",
            "managedScaling": {
                "status": "ENABLED",
                "targetCapacity": 100,
                "minimumScalingStepSize": 1,
                "maximumScalingStepSize": 10000,
                "instanceWarmupPeriod": 300
            },
            "managedTerminationProtection": "DISABLED",
            "managedDraining": "ENABLED"
        },
        "updateStatus": "UPDATE_IN_PROGRESS",
        "tags": []
    }
}
```

如需容量提供者的詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 EC2 啟動類型的 Amazon ECS 容量提供者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CapacityProviderUpdate。

create-capacity-provider

以下程式碼範例顯示如何使用 create-capacity-provider。

AWS CLI

建立容量提供者

下列 create-capacity-provider 範例會建立容量提供者，該提供者使用名為 MyASG 的 Auto Scaling 群組，並啟用受管擴展和受管終止保護。此組態用於 Amazon ECS 叢集自動擴展。

```
aws ecs create-capacity-provider \
    --name "MyCapacityProvider" \
    --auto-scaling-group-provider "autoScalingGroupArn=arn:aws:autoscaling:us-
east-1:123456789012:autoScalingGroup:57ffcb94-11f0-4d6d-
bf60-3bac5EXAMPLE:autoScalingGroupName/
MyASG,managedScaling={status=ENABLED,targetCapacity=100},managedTerminationProtection=ENABLE
```

輸出：

```
{
    "capacityProvider": {
    "capacityProviderArn": "arn:aws:ecs:us-east-1:123456789012:capacity-provider/
MyCapacityProvider",
    "name": "MyCapacityProvider",
    "status": "ACTIVE",
    "autoScalingGroupProvider": {
        "autoScalingGroupArn": "arn:aws:autoscaling:us-
east-1:132456789012:autoScalingGroup:57ffcb94-11f0-4d6d-
bf60-3bac5EXAMPLE:autoScalingGroupName/MyASG",
        "managedScaling": {
            "status": "ENABLED",
            "targetCapacity": 100,
            "minimumScalingStepSize": 1,
            "maximumScalingStepSize": 10000,
            "instanceWarmupPeriod": 300
        },
        "managedTerminationProtection": "ENABLED"
    },
    "tags": []
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon ECS 叢集自動擴展。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCapacityProvider。

create-cluster

以下程式碼範例顯示如何使用 create-cluster。

AWS CLI

範例 1：建立新叢集

下列create-cluster範例會建立名為 的叢集MyCluster，並啟用具有增強可觀測性的 CloudWatch Container Insights。

```
aws ecs create-cluster \
    --cluster-name MyCluster \
    --settings name=containerInsights,value=enhanced
```

輸出：

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "clusterName": "MyCluster",
        "status": "ACTIVE",
        "registeredContainerInstancesCount": 0,
        "pendingTasksCount": 0,
        "runningTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "settings": [
            {
                "name": "containerInsights",
                "value": "enhanced"
            }
        ],
        "tags": []
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的建立叢集。

範例 2：使用容量提供者建立新叢集

下列create-cluster範例會建立叢集，並將兩個現有容量提供者與其建立關聯。create-capacity-provider 命令用於建立容量提供者。指定預設容量提供者策略是選用的，但建

議使用。在此範例中，我們會建立名為 的叢集，MyCluster並將 MyCapacityProvider1 和MyCapacityProvider2容量提供者與其建立關聯。系統會指定預設容量提供者策略，將任務平均分散到兩個容量提供者。

```
aws ecs create-cluster \
    --cluster-name MyCluster \
    --capacity-providers MyCapacityProvider1 MyCapacityProvider2 \
    --default-capacity-provider-
strategy capacityProvider=MyCapacityProvider1,weight=1 capacityProvider=MyCapacityProvider2,
```

輸出：

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "clusterName": "MyCluster",
        "status": "PROVISIONING",
        "registeredContainerInstancesCount": 0,
        "pendingTasksCount": 0,
        "runningTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "settings": [
            {
                "name": "containerInsights",
                "value": "enabled"
            }
        ],
        "capacityProviders": [
            "MyCapacityProvider1",
            "MyCapacityProvider2"
        ],
        "defaultCapacityProviderStrategy": [
            {
                "capacityProvider": "MyCapacityProvider1",
                "weight": 1,
                "base": 0
            },
            {
                "capacityProvider": "MyCapacityProvider2",
                "weight": 1,
                "base": 0
            }
```

```
        ],
        "attachments": [
           {
               "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
               "type": "asp",
               "status": "PRECREATED",
               "details": [
                   {
                       "name": "capacityProviderName",
                       "value": "MyCapacityProvider1"
                   },
                   {
                       "name": "scalingPlanName",
                       "value": "ECSManagedAutoScalingPlan-a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
                   }
                ]
            },
            {
                "id": "ae592060-2382-4663-9476-b015c685593c",
                "type": "asp",
                "status": "PRECREATED",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider2"
                    },
                    {
                        "name": "scalingPlanName",
                        "value": "ECSManagedAutoScalingPlan-a1b2c3d4-5678-90ab-cdef-
EXAMPLE22222"
                    }
                ]
            }
        ],
        "attachmentsStatus": "UPDATE_IN_PROGRESS"
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的叢集容量提供者。

範例 3：建立具有多個標籤的新叢集

下列create-cluster範例會建立具有多個標籤的叢集。如需使用速記語法新增標籤的詳細資訊，請參閱《AWS CLI 使用者指南》中的使用速記語法搭配 AWS 命令列界面。

```
aws ecs create-cluster \
    --cluster-name MyCluster \
    --tags key=key1,value=value1 key=key2,value=value2
```

輸出：

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "clusterName": "MyCluster",
        "status": "ACTIVE",
        "registeredContainerInstancesCount": 0,
        "pendingTasksCount": 0,
        "runningTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "tags": [
            {
                "key": "key1",
                "value": "value1"
            },
            {
                "key": "key2",
                "value": "value2"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的建立叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCluster。

create-service

以下程式碼範例顯示如何使用 create-service。

AWS CLI

範例 1：使用 Fargate 任務建立服務

下列create-service範例示範如何使用 Fargate 任務建立服務。

```
aws ecs create-service \
    --cluster MyCluster \
    --service-name MyService \
    --task-definition sample-fargate:1 \
    --desired-count 2 \
    --launch-type FARGATE \
    --platform-version LATEST \
    --network-configuration
 'awsvpcConfiguration={subnets=[subnet-12344321],securityGroups=[sg-12344321],assignPublicIp
 \
    --tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3
```

輸出：

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/
MyService",
        "serviceName": "MyService",
          "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 0,
        "pendingCount": 0,
        "launchType": "FARGATE",
        "platformVersion": "LATEST",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
sample-fargate:1",
        "deploymentConfiguration": {
            "maximumPercent": 200,
            "minimumHealthyPercent": 100
        },
        "deployments": [
            {
                "id": "ecs-svc/1234567890123456789",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/sample-fargate:1",
                "desiredCount": 2,
```

```
                "pendingCount": 0,
                "runningCount": 0,
                "createdAt": 1557119253.821,
                "updatedAt": 1557119253.821,
                "launchType": "FARGATE",
                "platformVersion": "1.3.0",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                }
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [],
        "createdAt": 1557119253.821,
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321"
                ],
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "ENABLED"
            }
        },
        "schedulingStrategy": "REPLICA",
        "tags": [
            {
                "key": "key1",
                "value": "value1"
            },
            {
                "key": "key2",
```

```
                "value": "value2"
            },
            {
                "key": "key3",
                "value": "value3"
            }
        ],
        "enableECSManagedTags": false,
        "propagateTags": "NONE"
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的建立服務。

範例 2：使用 EC2 啟動類型建立服務

下列create-service範例示範如何建立名為 的服務，ecs-simple-service其中包含使用 EC2 啟動類型的任務。服務使用 sleep360 任務定義，並維持 1 個任務的執行個體。

```
aws ecs create-service \
    --cluster MyCluster \
    --service-name ecs-simple-service \
    --task-definition sleep360:2 \
    --desired-count 1
```

輸出：

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/ecs-
simple-service",
        "serviceName": "ecs-simple-service",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 1,
        "runningCount": 0,
        "pendingCount": 0,
        "launchType": "EC2",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
sleep360:2",
        "deploymentConfiguration": {
```

```
            "maximumPercent": 200,
            "minimumHealthyPercent": 100
        },
        "deployments": [
            {
                "id": "ecs-svc/1234567890123456789",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/sleep360:2",
                "desiredCount": 1,
                "pendingCount": 0,
                "runningCount": 0,
                "createdAt": 1557206498.798,
                "updatedAt": 1557206498.798,
                "launchType": "EC2"
            }
        ],
        "events": [],
        "createdAt": 1557206498.798,
        "placementConstraints": [],
        "placementStrategy": [],
        "schedulingStrategy": "REPLICA",
        "enableECSManagedTags": false,
        "propagateTags": "NONE"
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的建立服務。

範例 3：建立使用外部部署控制器的服務

下列create-service範例會建立使用外部部署控制器的服務。

```
aws ecs create-service \
    --cluster MyCluster \
    --service-name MyService \
    --deployment-controller type=EXTERNAL \
    --desired-count 1
```

輸出：

```
{
    "service": {
```

```
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/
MyService",
        "serviceName": "MyService",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 1,
        "runningCount": 0,
        "pendingCount": 0,
        "launchType": "EC2",
        "deploymentConfiguration": {
            "maximumPercent": 200,
            "minimumHealthyPercent": 100
        },
        "taskSets": [],
        "deployments": [],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [],
        "createdAt": 1557128207.101,
        "placementConstraints": [],
        "placementStrategy": [],
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "EXTERNAL"
        },
        "enableECSManagedTags": false,
        "propagateTags": "NONE"
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的建立服務。

範例 4：在負載平衡器後方建立新的服務

下列create-service範例示範如何建立負載平衡器後方的服務。您必須在與容器執行個體相同的區域中設定負載平衡器。此範例使用 --cli-input-json選項和名為 的 JSON 輸入檔案ecs-simple-service-elb.json，內容如下。

```
aws ecs create-service \
    --cluster MyCluster \
    --service-name ecs-simple-service-elb \
```

```
    --cli-input-json file://ecs-simple-service-elb.json
```

ecs-simple-service-elb.json 的內容：

```
 {
    "serviceName": "ecs-simple-service-elb",
    "taskDefinition": "ecs-demo",
    "loadBalancers": [
        {
            "loadBalancerName": "EC2Contai-EcsElast-123456789012",
            "containerName": "simple-demo",
            "containerPort": 80
        }
    ],
    "desiredCount": 10,
    "role": "ecsServiceRole"
}
```

輸出：

```
{
    "service": {
        "status": "ACTIVE",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/ecs-
demo:1",
        "pendingCount": 0,
        "loadBalancers": [
            {
                "containerName": "ecs-demo",
                "containerPort": 80,
                "loadBalancerName": "EC2Contai-EcsElast-123456789012"
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/ecsServiceRole",
        "desiredCount": 10,
        "serviceName": "ecs-simple-service-elb",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/ecs-simple-
service-elb",
        "deployments": [
            {
                "status": "PRIMARY",
                "pendingCount": 0,
```

```
                "createdAt": 1428100239.123,
                "desiredCount": 10,
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/ecs-demo:1",
                "updatedAt": 1428100239.123,
                "id": "ecs-svc/1234567890123456789",
                "runningCount": 0
            }
        ],
        "events": [],
        "runningCount": 0
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的使用負載平衡來分配 Amazon ECS 服務流量。

範例 5：在建立服務時設定 Amazon EBS 磁碟區

下列create-service範例示範如何為 服務管理的每個任務設定 Amazon EBS 磁碟區。您必須將 Amazon ECS 基礎設施角色設定為已連接AmazonECSInfrastructureRolePolicyForVolumes受管政策。您必須指定具有與create-service請求中相同磁碟區名稱的任務定義。此範例使用 --cli-input-json選項和名為 的 JSON 輸入檔案ecs-simple-service-ebs.json，內容如下。

```
aws ecs create-service \
    --cli-input-json file://ecs-simple-service-ebs.json
```

ecs-simple-service-ebs.json 的內容：

```
{
    "cluster": "mycluster",
    "taskDefinition": "mytaskdef",
    "serviceName": "ecs-simple-service-ebs",
    "desiredCount": 2,
    "launchType": "FARGATE",
    "networkConfiguration":{
        "awsvpcConfiguration":{
            "assignPublicIp": "ENABLED",
            "securityGroups": ["sg-12344321"],
            "subnets":["subnet-12344321"]
        }
```

```
    },
    "volumeConfigurations": [
        {
            "name": "myEbsVolume",
            "managedEBSVolume": {
                "roleArn":"arn:aws:iam::123456789012:role/ecsInfrastructureRole",
                "volumeType": "gp3",
                "sizeInGiB": 100,
                "iops": 3000,
                "throughput": 125,
                "filesystemType": "ext4"
            }
        }
   ]
}
```

輸出：

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/mycluster/ecs-
simple-service-ebs",
        "serviceName": "ecs-simple-service-ebs",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/mycluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 0,
        "pendingCount": 0,
        "launchType": "EC2",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
mytaskdef:3",
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": false,
                "rollback": false
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100
        },
        "deployments": [
            {
```

```
                "id": "ecs-svc/7851020056849183687",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:3",
                "desiredCount": 0,
                "pendingCount": 0,
                "runningCount": 0,
                "failedTasks": 0,
                "createdAt": "2025-01-21T11:32:38.034000-06:00",
                "updatedAt": "2025-01-21T11:32:38.034000-06:00",
                "launchType": "EC2",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "DISABLED"
                    }
                },
                "rolloutState": "IN_PROGRESS",
                "rolloutStateReason": "ECS deployment ecs-svc/7851020056849183687 in
 progress.",
                "volumeConfigurations": [
                    {
                        "name": "myEBSVolume",
                        "managedEBSVolume": {
                            "volumeType": "gp3",
                            "sizeInGiB": 100,
                            "iops": 3000,
                            "throughput": 125,
                            "roleArn": "arn:aws:iam::123456789012:role/
ecsInfrastructureRole",
                            "filesystemType": "ext4"
                        }
                    }
                ]
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [],
```

```
        "createdAt": "2025-01-21T11:32:38.034000-06:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321"
                ],
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "DISABLED"
            }
        },
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
        },
        "createdBy": "arn:aws:iam::123456789012:user/AIDACKCEVSQ6C2EXAMPLE",
        "enableECSManagedTags": false,
        "propagateTags": "NONE",
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "DISABLED"
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的搭配使用 Amazon EBS 磁碟區與 Amazon ECS。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateService。

`create-task-set`

以下程式碼範例顯示如何使用 `create-task-set`。

AWS CLI

建立任務集

下列`create-task-set`範例會在使用外部部署控制器的服務中建立任務集。

```
aws ecs create-task-set \
```

```
    --cluster MyCluster \
    --service MyService \
    --task-definition MyTaskDefinition:2 \
    --network-
configuration "awsvpcConfiguration={subnets=[subnet-12344321],securityGroups=[sg-12344321]}"
```

輸出：

```
{
    "taskSet": {
        "id": "ecs-svc/1234567890123456789",
        "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/
MyService/ecs-svc/1234567890123456789",
        "status": "ACTIVE",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
MyTaskDefinition:2",
        "computedDesiredCount": 0,
        "pendingCount": 0,
        "runningCount": 0,
        "createdAt": 1557128360.711,
        "updatedAt": 1557128360.711,
        "launchType": "EC2",
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321"
                ],
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "DISABLED"
            }
        },
        "loadBalancers": [],
        "serviceRegistries": [],
        "scale": {
            "value": 0.0,
            "unit": "PERCENT"
        },
        "stabilityStatus": "STABILIZING",
        "stabilityStatusAt": 1557128360.711
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTaskSet。

delete-account-setting

以下程式碼範例顯示如何使用 delete-account-setting。

AWS CLI

刪除特定 IAM 使用者或 IAM 角色的帳戶設定

下列範例會delete-account-setting刪除特定 IAM 使用者或 IAM 角色的帳戶設定。

```
aws ecs delete-account-setting \
    --name serviceLongArnFormat \
    --principal-arn arn:aws:iam::123456789012:user/MyUser
```

輸出：

```
{
    "setting": {
        "name": "serviceLongArnFormat",
        "value": "enabled",
        "principalArn": "arn:aws:iam::123456789012:user/MyUser"
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon Resource Name (ARNs) 和 IDs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAccountSetting。

delete-attributes

以下程式碼範例顯示如何使用 delete-attributes。

AWS CLI

從 Amazon ECS 資源刪除一或多個自訂屬性

以下內容會從容器執行個體delete-attributes刪除名稱為 stack的屬性。

```
aws ecs delete-attributes \
```

```
    --attributes name=stack,targetId=arn:aws:ecs:us-west-2:130757420319:container-
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34
```

輸出：

```
{
    "attributes": [
        {
            "name": "stack",
            "targetId": "arn:aws:ecs:us-west-2:130757420319:container-
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",
            "value": "production"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAttributes。

delete-capacity-provider

以下程式碼範例顯示如何使用 delete-capacity-provider。

AWS CLI

範例 1：使用 Amazon Resource Name (ARN) 刪除容量提供者

下列delete-capacity-provider範例透過指定容量提供者的 Amazon Resource Name (ARN) 來刪除容量提供者。您可以使用 describe-capacity-providers命令擷取 ARN 以及容量提供者刪除的狀態。

```
aws ecs delete-capacity-provider \
    --capacity-provider arn:aws:ecs:us-west-2:123456789012:capacity-provider/
ExampleCapacityProvider
```

輸出：

```
{
    "capacityProvider": {
        "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/ExampleCapacityProvider",
        "name": "ExampleCapacityProvider",
        "status": "ACTIVE",
```

```
        "autoScalingGroupProvider": {
            "autoScalingGroupArn": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111:autoScalingGroupName/MyAutoScalingGroup",
            "managedScaling": {
                "status": "ENABLED",
                "targetCapacity": 100,
                "minimumScalingStepSize": 1,
                "maximumScalingStepSize": 10000
            },
            "managedTerminationProtection": "DISABLED"
        },
        "updateStatus": "DELETE_IN_PROGRESS",
        "tags": []
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的叢集容量提供者。

範例 2：使用名稱刪除容量提供者

下列delete-capacity-provider範例透過指定容量提供者的簡短名稱來刪除容量提供者。您可以使用 describe-capacity-providers命令擷取簡短名稱以及容量提供者刪除的狀態。

```
aws ecs delete-capacity-provider \
    --capacity-provider ExampleCapacityProvider
```

輸出：

```
{
    "capacityProvider": {
        "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/ExampleCapacityProvider",
        "name": "ExampleCapacityProvider",
        "status": "ACTIVE",
        "autoScalingGroupProvider": {
            "autoScalingGroupArn": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111:autoScalingGroupName/MyAutoScalingGroup",
            "managedScaling": {
                "status": "ENABLED",
                "targetCapacity": 100,
                "minimumScalingStepSize": 1,
```

```
                "maximumScalingStepSize": 10000
            },
            "managedTerminationProtection": "DISABLED"
        },
        "updateStatus": "DELETE_IN_PROGRESS",
        "tags": []
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的叢集容量提供者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCapacityProvider。

delete-cluster

以下程式碼範例顯示如何使用 delete-cluster。

AWS CLI

刪除空叢集

下列delete-cluster範例會刪除指定的空白叢集。

```
aws ecs delete-cluster --cluster MyCluster
```

輸出：

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "status": "INACTIVE",
        "clusterName": "MyCluster",
        "registeredContainerInstancesCount": 0,
        "pendingTasksCount": 0,
        "runningTasksCount": 0,
        "activeServicesCount": 0
        "statistics": [],
        "tags": []
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的刪除叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCluster。

delete-service

以下程式碼範例顯示如何使用 delete-service。

AWS CLI

刪除服務

下列ecs delete-service範例會從叢集刪除指定的服務。您可以包含 --force 參數來刪除服務，即使該服務尚未擴展到零任務。

```
aws ecs delete-service --cluster MyCluster --service MyService1 --force
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的刪除服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteService。

delete-task-definitions

以下程式碼範例顯示如何使用 delete-task-definitions。

AWS CLI

刪除任務定義

下列delete-task-definitions範例會刪除 INACTIVE 任務定義。

```
aws ecs delete-task-definitions \
    --task-definition curltest:1
```

輸出：

```
{
"taskDefinitions": [
    {
        "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/
curltest:1",
        "containerDefinitions": [
            {
                "name": "ctest",
```

```
                "image": "mreferre/eksutils",
                "cpu": 0,
                "portMappings": [],
                "essential": true,
                "entryPoint": [
                    "sh",
                    "-c"
                ],
                "command": [
                    "curl ${ECS_CONTAINER_METADATA_URI_V4}/task"
                ],
                "environment": [],
                "mountPoints": [],
                "volumesFrom": [],
                "logConfiguration": {
                    "logDriver": "awslogs",
                    "options": {
                        "awslogs-create-group": "true",
                        "awslogs-group": "/ecs/curltest",
                        "awslogs-region": "us-east-1",
                        "awslogs-stream-prefix": "ecs"
                    }
                }
            }
        ],
        "family": "curltest",
        "taskRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
        "executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
        "networkMode": "awsvpc",
        "revision": 1,
        "volumes": [],
        "status": "DELETE_IN_PROGRESS",
        "compatibilities": [
            "EC2",
            "FARGATE"
        ],
        "requiresCompatibilities": [
            "FARGATE"
        ],
        "cpu": "256",
        "memory": "512",
        "registeredAt": "2021-09-10T12:56:24.704000+00:00",
        "deregisteredAt": "2023-03-14T15:20:59.419000+00:00",
        "registeredBy": "arn:aws:sts::123456789012:assumed-role/Admin/jdoe"
```

```
        }
    ],
    "failures": []
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon ECS 任務定義。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTaskDefinitions。

delete-task-set

以下程式碼範例顯示如何使用 delete-task-set。

AWS CLI

刪除任務集

下列delete-task-set範例示範如何刪除任務集。您可以包含 --force 參數來刪除任務集，即使任務集尚未擴展到零。

```
aws ecs delete-task-set \
    --cluster MyCluster \
    --service MyService \
    --task-set arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/MyService/ecs-
svc/1234567890123456789 \
    --force
```

輸出：

```
{
    "taskSet": {
        "id": "ecs-svc/1234567890123456789",
        "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/
MyService/ecs-svc/1234567890123456789",
        "status": "DRAINING",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
sample-fargate:2",
        "computedDesiredCount": 0,
        "pendingCount": 0,
        "runningCount": 0,
        "createdAt": 1557130260.276,
        "updatedAt": 1557130290.707,
        "launchType": "EC2",
```

```
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12345678"
                ],
                "securityGroups": [
                    "sg-12345678"
                ],
                "assignPublicIp": "DISABLED"
            }
        },
        "loadBalancers": [],
        "serviceRegistries": [],
        "scale": {
            "value": 0.0,
            "unit": "PERCENT"
        },
        "stabilityStatus": "STABILIZING",
        "stabilityStatusAt": 1557130290.707
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTaskSet。

deregister-container-instance

以下程式碼範例顯示如何使用 deregister-container-instance。

AWS CLI

從叢集取消註冊容器執行個體

下列deregister-container-instance範例會從指定的叢集取消註冊容器執行個體。如果容器執行個體中仍有正在執行的任務，您必須在取消註冊之前停止這些任務，或使用 --force選項。

```
aws ecs deregister-container-instance \
    --cluster arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \
    --container-instance arn:aws:ecs:us-west-2:123456789012:container-instance/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --force
```

輸出：

```
{
    "containerInstance": {
        "remainingResources": [
            {
                "integerValue": 1024,
                "doubleValue": 0.0,
                "type": "INTEGER",
                "longValue": 0,
                "name": "CPU"
            },
            {
                "integerValue": 985,
                "doubleValue": 0.0,
                "type": "INTEGER",
                "longValue": 0,
                "name": "MEMORY"
            },
            {
                "type": "STRINGSET",
                "integerValue": 0,
                "name": "PORTS",
                "stringSetValue": [
                    "22",
                    "2376",
                    "2375",
                    "51678",
                    "51679"
                ],
                "longValue": 0,
                "doubleValue": 0.0
            },
            {
                "type": "STRINGSET",
                "integerValue": 0,
                "name": "PORTS_UDP",
                "stringSetValue": [],
                "longValue": 0,
                "doubleValue": 0.0
            }
        ],
        "agentConnected": true,
        "attributes": [
            {
```

```
            "name": "ecs.capability.secrets.asm.environment-variables"
        },
        {
            "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
        },
        {
            "value": "ami-01a82c3fce2c3ba58",
            "name": "ecs.ami-id"
        },
        {
            "name": "ecs.capability.secrets.asm.bootstrap.log-driver"
        },
        {
            "name": "com.amazonaws.ecs.capability.logging-driver.none"
        },
        {
            "name": "ecs.capability.ecr-endpoint"
        },
        {
            "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
        },
        {
            "value": "vpc-1234567890123467",
            "name": "ecs.vpc-id"
        },
        {
            "name": "ecs.capability.execution-role-awslogs"
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
        },
        {
            "name": "ecs.capability.docker-plugin.local"
        },
        {
            "name": "ecs.capability.task-eni"
        },
        {
```

```
            "name": "ecs.capability.task-cpu-mem-limit"
        },
        {
            "name": "ecs.capability.secrets.ssm.bootstrap.log-driver"
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remote-api.1.30"
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remote-api.1.31"
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remote-api.1.32"
        },
        {
            "name": "ecs.capability.execution-role-ecr-pull"
        },
        {
            "name": "ecs.capability.container-health-check"
        },
        {
            "value": "subnet-1234567890123467",
            "name": "ecs.subnet-id"
        },
        {
            "value": "us-west-2a",
            "name": "ecs.availability-zone"
        },
        {
            "value": "t2.micro",
            "name": "ecs.instance-type"
        },
        {
            "name": "com.amazonaws.ecs.capability.task-iam-role-network-host"
        },
        {
            "name": "ecs.capability.aws-appmesh"
        },
        {
            "name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remote-api.1.24"
        },
```

```
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.25"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.26"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.27"
            },
            {
                "name": "com.amazonaws.ecs.capability.privileged-container"
            },
            {
                "name": "ecs.capability.container-ordering"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.28"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.29"
            },
            {
                "value": "x86_64",
                "name": "ecs.cpu-architecture"
            },
            {
                "value": "93f43776-2018.10.0",
                "name": "ecs.capability.cni-plugin-version"
            },
            {
                "name": "ecs.capability.secrets.ssm.environment-variables"
            },
            {
                "name": "ecs.capability.pid-ipc-namespace-sharing"
            },
            {
                "name": "com.amazonaws.ecs.capability.ecr-auth"
            },
            {
                "value": "linux",
                "name": "ecs.os-type"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.20"
```

```
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.21"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.22"
            },
            {
                "name": "ecs.capability.task-eia"
            },
            {
                "name": "ecs.capability.private-registry-
authentication.secretsmanager"
            },
            {
                "name": "com.amazonaws.ecs.capability.task-iam-role"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.23"
            }
        ],
        "pendingTasksCount": 0,
        "tags": [],
        "containerInstanceArn": "arn:aws:ecs:us-west-2:123456789012:container-
instance/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "registeredResources": [
            {
                "integerValue": 1024,
                "doubleValue": 0.0,
                "type": "INTEGER",
                "longValue": 0,
                "name": "CPU"
            },
            {
                "integerValue": 985,
                "doubleValue": 0.0,
                "type": "INTEGER",
                "longValue": 0,
                "name": "MEMORY"
            },
            {
                "type": "STRINGSET",
                "integerValue": 0,
                "name": "PORTS",
```

```
                "stringSetValue": [
                    "22",
                    "2376",
                    "2375",
                    "51678",
                    "51679"
                ],
                "longValue": 0,
                "doubleValue": 0.0
            },
            {
                "type": "STRINGSET",
                "integerValue": 0,
                "name": "PORTS_UDP",
                "stringSetValue": [],
                "longValue": 0,
                "doubleValue": 0.0
            }
        ],
        "status": "INACTIVE",
        "registeredAt": 1557768075.681,
        "version": 4,
        "versionInfo": {
            "agentVersion": "1.27.0",
            "agentHash": "aabe65ee",
            "dockerVersion": "DockerVersion: 18.06.1-ce"
        },
        "attachments": [],
        "runningTasksCount": 0,
        "ec2InstanceId": "i-12345678901234678"
    }
}
```

如需詳細資訊，請參閱 ECS 開發人員指南中的取消註冊容器執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterContainerInstance。

deregister-task-definition

以下程式碼範例顯示如何使用 deregister-task-definition。

AWS CLI

取消註冊任務定義

下列deregister-task-definition範例會取消註冊預設區域中curler任務定義的第一個修訂。

```
aws ecs deregister-task-definition --task-definition curler:1
```

請注意，在產生的輸出中，任務定義狀態會顯示 INACTIVE：

```
{
    "taskDefinition": {
        "status": "INACTIVE",
        "family": "curler",
        "volumes": [],
        "taskDefinitionArn": "arn:aws:ecs:us-west-2:123456789012:task-definition/
curler:1",
        "containerDefinitions": [
            {
                "environment": [],
                "name": "curler",
                "mountPoints": [],
                "image": "curl:latest",
                "cpu": 100,
                "portMappings": [],
                "entryPoint": [],
                "memory": 256,
                "command": [
                    "curl -v http://example.com/"
                ],
                "essential": true,
                "volumesFrom": []
            }
        ],
        "revision": 1
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon ECS 任務定義。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterTaskDefinition。

describe-capacity-providers

以下程式碼範例顯示如何使用 describe-capacity-providers。

AWS CLI

範例 1：描述所有容量提供者

下列describe-capacity-providers範例會擷取所有容量提供者的詳細資訊。

```
aws ecs describe-capacity-providers
```

輸出：

```
{
    "capacityProviders": [
        {
            "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/MyCapacityProvider",
            "name": "MyCapacityProvider",
            "status": "ACTIVE",
            "autoScalingGroupProvider": {
                "autoScalingGroupArn": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111:autoScalingGroupName/MyAutoScalingGroup",
                "managedScaling": {
                    "status": "ENABLED",
                    "targetCapacity": 100,
                    "minimumScalingStepSize": 1,
                    "maximumScalingStepSize": 1000
                },
                "managedTerminationProtection": "ENABLED"
            },
            "tags": []
        },
        {
            "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/FARGATE",
            "name": "FARGATE",
            "status": "ACTIVE",
            "tags": []
        },
        {
            "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/FARGATE_SPOT",
            "name": "FARGATE_SPOT",
            "status": "ACTIVE",
```

```
            "tags": []
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的叢集容量提供者。

範例 2：描述特定容量提供者

下列describe-capacity-providers範例會擷取特定容量提供者的詳細資訊。使用 --include TAGS 參數會將與容量提供者相關聯的標籤新增至輸出。

```
aws ecs describe-capacity-providers \
    --capacity-providers MyCapacityProvider \
    --include TAGS
```

輸出：

```
{
    "capacityProviders": [
        {
            "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/MyCapacityProvider",
            "name": "MyCapacityProvider",
            "status": "ACTIVE",
            "autoScalingGroupProvider": {
                "autoScalingGroupArn": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111:autoScalingGroupName/MyAutoScalingGroup",
                "managedScaling": {
                    "status": "ENABLED",
                    "targetCapacity": 100,
                    "minimumScalingStepSize": 1,
                    "maximumScalingStepSize": 1000
                },
                "managedTerminationProtection": "ENABLED"
            },
            "tags": [
                {
                    "key": "environment",
                    "value": "production"
                }
            ]
```

```
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的叢集容量提供者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCapacityProviders。

describe-clusters

以下程式碼範例顯示如何使用 describe-clusters。

AWS CLI

範例 1：描述叢集

下列describe-clusters範例會擷取指定叢集的詳細資訊。

```
aws ecs describe-clusters \
    --cluster default
```

輸出：

```
{
    "clusters": [
        {
            "status": "ACTIVE",
            "clusterName": "default",
            "registeredContainerInstancesCount": 0,
            "pendingTasksCount": 0,
            "runningTasksCount": 0,
            "activeServicesCount": 1,
            "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default"
        }
    ],
    "failures": []
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon ECS 叢集。

範例 2：使用附件選項描述叢集

下列describe-clusters範例指定 ATTACHMENTS 選項。它會擷取指定叢集的詳細資訊，以及以附件形式連接至叢集的資源清單。將容量提供者與叢集搭配使用時，資源會以 asp 或 as_policy ATTACHMENTS 表示，可以是 AutoScaling 計劃或擴展政策。

```
aws ecs describe-clusters \
    --include ATTACHMENTS \
    --clusters sampleCluster
```

輸出：

```
{
    "clusters": [
        {
            "clusterArn": "arn:aws:ecs:af-south-1:123456789222:cluster/
sampleCluster",
            "clusterName": "sampleCluster",
            "status": "ACTIVE",
            "registeredContainerInstancesCount": 0,
            "runningTasksCount": 0,
            "pendingTasksCount": 0,
            "activeServicesCount": 0,
            "statistics": [],
            "tags": [],
            "settings": [],
            "capacityProviders": [
                "sampleCapacityProvider"
            ],
            "defaultCapacityProviderStrategy": [],
            "attachments": [
                {
                    "id": "a1b2c3d4-5678-901b-cdef-EXAMPLE22222",
                    "type": "as_policy",
                    "status": "CREATED",
                    "details": [
                        {
                            "name": "capacityProviderName",
                            "value": "sampleCapacityProvider"
                        },
                        {
                            "name": "scalingPolicyName",
                            "value": "ECSManagedAutoScalingPolicy-3048e262-
fe39-4eaf-826d-6f975d303188"
                        }
```

```
                    ]
                }
            ],
            "attachmentsStatus": "UPDATE_COMPLETE"
        }
    ],
    "failures": []
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon ECS 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClusters。

describe-container-instances

以下程式碼範例顯示如何使用 describe-container-instances。

AWS CLI

描述容器執行個體

下列describe-container-instances範例會使用容器執行個體 UUID 做為識別符，擷取update叢集中容器執行個體的詳細資訊。

```
aws ecs describe-container-instances \
    --cluster update \
    --container-instances a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

輸出：

```
{
    "failures": [],
    "containerInstances": [
        {
            "status": "ACTIVE",
            "registeredResources": [
                {
                    "integerValue": 2048,
                    "longValue": 0,
                    "type": "INTEGER",
                    "name": "CPU",
                    "doubleValue": 0.0
                },
```

```
                {
                    "integerValue": 3955,
                    "longValue": 0,
                    "type": "INTEGER",
                    "name": "MEMORY",
                    "doubleValue": 0.0
                },
                {
                    "name": "PORTS",
                    "longValue": 0,
                    "doubleValue": 0.0,
                    "stringSetValue": [
                        "22",
                        "2376",
                        "2375",
                        "51678"
                    ],
                    "type": "STRINGSET",
                    "integerValue": 0
                }
            ],
            "ec2InstanceId": "i-A1B2C3D4",
            "agentConnected": true,
            "containerInstanceArn": "arn:aws:ecs:us-west-2:123456789012:container-
instance/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "pendingTasksCount": 0,
            "remainingResources": [
                {
                    "integerValue": 2048,
                    "longValue": 0,
                    "type": "INTEGER",
                    "name": "CPU",
                    "doubleValue": 0.0
                },
                {
                    "integerValue": 3955,
                    "longValue": 0,
                    "type": "INTEGER",
                    "name": "MEMORY",
                    "doubleValue": 0.0
                },
                {
                    "name": "PORTS",
                    "longValue": 0,
```

```
                    "doubleValue": 0.0,
                    "stringSetValue": [
                        "22",
                        "2376",
                        "2375",
                        "51678"
                    ],
                    "type": "STRINGSET",
                    "integerValue": 0
                }
            ],
            "runningTasksCount": 0,
            "versionInfo": {
                "agentVersion": "1.0.0",
                "agentHash": "4023248",
                "dockerVersion": "DockerVersion: 1.5.0"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon ECS 容器執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeContainerInstances。

describe-service-deployments

以下程式碼範例顯示如何使用 describe-service-deployments。

AWS CLI

描述服務部署詳細資訊

下列describe-service-deployments範例會使用 ARN 傳回服務部署的服務部署詳細資訊arn:aws:ecs:us-east-1:123456789012:service-deployment/example-cluster/example-service/ejGvqq2ilnbKT9qj0vLJe。

```
aws ecs describe-service-deployments \
    --service-deployment-arn arn:aws:ecs:us-east-1:123456789012:service-deployment/example-cluster/example-service/ejGvqq2ilnbKT9qj0vLJe
```

輸出：

```
{
    "serviceDeployments": [
        {
            "serviceDeploymentArn": "arn:aws:ecs:us-east-1:123456789012:service-
deployment/example-cluster/example-service/ejGvqq2ilnbKT9qj0vLJe",
            "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/example-
cluster/example-service",
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/example-
cluster",
            "createdAt": "2024-10-31T08:03:30.917000-04:00",
            "startedAt": "2024-10-31T08:03:32.510000-04:00",
            "finishedAt": "2024-10-31T08:05:04.527000-04:00",
            "updatedAt": "2024-10-31T08:05:04.527000-04:00",
            "sourceServiceRevisions": [],
            "targetServiceRevision": {
                "arn": "arn:aws:ecs:us-east-1:123456789012:service-revision/example-
cluster/example-service/1485800978477494678",
                "requestedTaskCount": 1,
                "runningTaskCount": 1,
                "pendingTaskCount": 0
            },
            "status": "SUCCESSFUL",
            "deploymentConfiguration": {
                "deploymentCircuitBreaker": {
                    "enable": true,
                    "rollback": true
                },
                "maximumPercent": 200,
                "minimumHealthyPercent": 100,
                "alarms": {
                    "alarmNames": [],
                    "rollback": false,
                    "enable": false
                }
            },
            "deploymentCircuitBreaker": {
                "status": "MONITORING_COMPLETE",
                "failureCount": 0,
                "threshold": 3
            },
            "alarms": {
                "status": "DISABLED"
            }
```

```
        }
    ],
    "failures": []
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的使用 Amazon ECS 服務部署檢視服務歷史記錄。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeServiceDeployments。

describe-service-revisions

以下程式碼範例顯示如何使用 describe-service-revisions。

AWS CLI

描述服務修訂詳細資訊

下列describe-service-revisions範例會使用 ARN 傳回服務修訂的服務修訂詳細資訊arn:aws:ecs:us-east-1:123456789012:service-revision/example-cluster/example-service/1485800978477494678。

```
aws ecs describe-service-revisions \
    --service-revision-arns arn:aws:ecs:us-east-1:123456789012:service-revision/example-cluster/example-service/1485800978477494678
```

輸出：

```
{
    "serviceRevisions": [
        {
            "serviceRevisionArn": "arn:aws:ecs:us-east-1:123456789012:service-revision/example-cluster/example-service/1485800978477494678",
            "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/example-cluster/example-service",
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/example-cluster",
            "taskDefinition": "arn:aws:ecs:us-east-1:123456789012:task-definition/webserver:5",
            "capacityProviderStrategy": [
                {
                    "capacityProvider": "FARGATE",
                    "weight": 1,
```

```
                    "base": 0
                }
            ],
            "platformVersion": "1.4.0",
            "platformFamily": "Linux",
            "networkConfiguration": {
                "awsvpcConfiguration": {
                    "subnets": [
                        "subnet-0d0eab1bb38d5ca64",
                        "subnet-0db5010045995c2d5"
                    ],
                    "securityGroups": [
                        "sg-02556bf85a191f59a"
                    ],
                    "assignPublicIp": "ENABLED"
                }
            },
            "containerImages": [
                {
                    "containerName": "aws-otel-collector",
                    "imageDigest":
 "sha256:7a1b3560655071bcacd66902c20ebe9a69470d5691fe3bd36baace7c2f3c4640",
                    "image": "public.ecr.aws/aws-observability/aws-otel-
collector:v0.32.0"
                },
                {
                    "containerName": "web",
                    "imageDigest":
 "sha256:28402db69fec7c17e179ea87882667f1e054391138f77ffaf0c3eb388efc3ffb",
                    "image": "nginx"
                }
            ],
            "guardDutyEnabled": false,
            "serviceConnectConfiguration": {
                "enabled": false
            },
            "createdAt": "2024-10-31T08:03:29.302000-04:00"
        }
    ],
    "failures": []
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon ECS 服務修訂。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeServiceRevisions。

describe-services

以下程式碼範例顯示如何使用 describe-services。

AWS CLI

描述服務

下列describe-services範例會擷取預設叢集中my-http-service服務的詳細資訊。

```
aws ecs describe-services --services my-http-service
```

輸出：

```
{
    "services": [
        {
            "status": "ACTIVE",
            "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
amazon-ecs-sample:1",
            "pendingCount": 0,
            "loadBalancers": [],
            "desiredCount": 10,
            "createdAt": 1466801808.595,
            "serviceName": "my-http-service",
            "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default",
            "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/my-http-
service",
            "deployments": [
                {
                    "status": "PRIMARY",
                    "pendingCount": 0,
                    "createdAt": 1466801808.595,
                    "desiredCount": 10,
                    "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/amazon-ecs-sample:1",
                    "updatedAt": 1428326312.703,
                    "id": "ecs-svc/1234567890123456789",
                    "runningCount": 10
                }
            ],
```

```
            "events": [
                {
                    "message": "(service my-http-service) has reached a steady
 state.",
                    "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
                    "createdAt": 1466801812.435
                }
            ],
            "runningCount": 10
        }
    ],
    "failures": []
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeServices。

describe-task-definition

以下程式碼範例顯示如何使用 describe-task-definition。

AWS CLI

描述任務定義

下列describe-task-definition範例會擷取任務定義的詳細資訊。

```
aws ecs describe-task-definition \
    --task-definition hello_world:8
```

輸出：

```
{
    "taskDefinition": {
        "taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-definition/
hello_world:8",
        "containerDefinitions": [
            {
                "cpu": 10,
                "environment": [],
                "essential": true,
                "image": "wordpress",
```

```
                "links": [
                    "mysql"
                ] ,
                "memory": 500,
                "mountPoints": [],
                "name": "wordpress",
                "portMappings": [
                    {
                        "containerPort": 80,
                        "hostPort": 80
                    }
                ],
                "volumesFrom": []
            },
            {
                "cpu": 10,
                "environment": [
                    {
                        "name": "MYSQL_ROOT_PASSWORD",
                        "value": "password"
                    }
                ],
                "essential": true,
                "image": "mysql",
                "memory": 500,
                "mountPoints": [],
                "name": "mysql",
                "portMappings": [],
                "volumesFrom": []
            }
        ],
    "family": "hello_world",
    "revision": 8,
    "volumes": [],
    "status": "ACTIVE",
    "placementConstraints": [],
    "compatibilities": [
        "EXTERNAL",
        "EC2"
    ],
    "registeredAt": "2024-06-21T11:15:12.669000-05:00",
    "registeredBy": "arn:aws:sts::012345678910:assumed-role/demo-role/jane-doe"
    },
    "tags": []
```

```
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon ECS 任務定義。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTaskDefinition。

describe-task-sets

以下程式碼範例顯示如何使用 describe-task-sets。

AWS CLI

描述任務集

下列describe-task-sets範例說明使用外部部署器的服務中的任務集。

```
aws ecs describe-task-sets \
    --cluster MyCluster \
    --service MyService \
    --task-sets arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/MyService/ecs-
svc/1234567890123456789
```

輸出：

```
{
    "taskSets": [
        {
            "id": "ecs-svc/1234567890123456789",
            "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/
MyService/ecs-svc/1234567890123456789",
            "status": "ACTIVE",
            "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
sample-fargate:2",
            "computedDesiredCount": 0,
            "pendingCount": 0,
            "runningCount": 0,
            "createdAt": 1557207715.195,
            "updatedAt": 1557207740.014,
            "launchType": "EC2",
            "networkConfiguration": {
                "awsvpcConfiguration": {
                    "subnets": [
                        "subnet-12344321"
```

```
                    ],
                    "securityGroups": [
                        "sg-1234431"
                    ],
                    "assignPublicIp": "DISABLED"
                }
            },
            "loadBalancers": [],
            "serviceRegistries": [],
            "scale": {
                "value": 0.0,
                "unit": "PERCENT"
            },
            "stabilityStatus": "STEADY_STATE",
            "stabilityStatusAt": 1557207740.014
        }
    ],
    "failures": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTaskSets。

describe-tasks

以下程式碼範例顯示如何使用 describe-tasks。

AWS CLI

檢查 1：描述單一任務任務

下列describe-tasks範例會擷取叢集中任務的詳細資訊。您可以使用任務的 ID 或完整 ARN 來指定任務。此範例使用任務的完整 ARN。

```
aws ecs describe-tasks \
    --cluster MyCluster \
    --tasks arn:aws:ecs:us-east-1:123456789012:task/
MyCluster/4d590253bb114126b7afa7b58EXAMPLE
```

輸出：

```
{
    "tasks": [
```

```
        {
            "attachments": [],
            "attributes": [
                {
                    "name": "ecs.cpu-architecture",
                    "value": "x86_64"
                }
            ],
            "availabilityZone": "us-east-1b",
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
            "connectivity": "CONNECTED",
            "connectivityAt": "2021-08-11T12:21:26.681000-04:00",
            "containerInstanceArn": "arn:aws:ecs:us-east-1:123456789012:container-
instance/test/025c7e2c5e054a6790a29fc1fEXAMPLE",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/
MyCluster/4d590253bb114126b7afa7b58eea9221/a992d1cc-ea46-474a-b6e8-24688EXAMPLE",
                    "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/
MyCluster/4d590253bb114126b7afa7b58EXAMPLE",
                    "name": "simple-app",
                    "image": "httpd:2.4",
                    "runtimeId":
 "91251eed27db90006ad67b1a08187290869f216557717dd5c39b37c94EXAMPLE",
                    "lastStatus": "RUNNING",
                    "networkBindings": [
                        {
                            "bindIP": "0.0.0.0",
                            "containerPort": 80,
                            "hostPort": 80,
                            "protocol": "tcp"
                        }
                    ],
                    "networkInterfaces": [],
                    "healthStatus": "UNKNOWN",
                    "cpu": "10",
                    "memory": "300"
                }
            ],
            "cpu": "10",
            "createdAt": "2021-08-11T12:21:26.681000-04:00",
            "desiredStatus": "RUNNING",
            "enableExecuteCommand": false,
            "group": "service:testupdate",
```

```
            "healthStatus": "UNKNOWN",
            "lastStatus": "RUNNING",
            "launchType": "EC2",
            "memory": "300",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "simple-app"
                    }
                ],
                "inferenceAcceleratorOverrides": []
            },
            "pullStartedAt": "2021-08-11T12:21:28.234000-04:00",
            "pullStoppedAt": "2021-08-11T12:21:33.793000-04:00",
            "startedAt": "2021-08-11T12:21:34.945000-04:00",
            "startedBy": "ecs-svc/968695068243EXAMPLE",
            "tags": [],
            "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/
MyCluster/4d590253bb114126b7afa7b58eea9221",
            "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-
definition/console-sample-app-static2:1",
            "version": 2
        }
    ],
    "failures": []
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon ECS 任務定義。

檢查 2：描述多個任務

下列describe-tasks範例會擷取叢集中多個任務的詳細資訊。您可以使用任務的 ID 或完整 ARN 來指定任務。此範例使用任務的完整 IDs。

```
aws ecs describe-tasks \
    --cluster MyCluster \
    --tasks "74de0355a10a4f979ac495c14EXAMPLE" "d789e94343414c25b9f6bd59eEXAMPLE"
```

輸出：

```
{
    "tasks": [
        {
```

```
            "attachments": [
                {
                    "id": "d9e7735a-16aa-4128-bc7a-b2d51EXAMPLE",
                    "type": "ElasticNetworkInterface",
                    "status": "ATTACHED",
                    "details": [
                        {
                            "name": "subnetId",
                            "value": "subnet-0d0eab1bb3EXAMPLE"
                        },
                        {
                            "name": "networkInterfaceId",
                            "value": "eni-0fa40520aeEXAMPLE"
                        },
                        {
                            "name": "macAddress",
                            "value": "0e:89:76:28:07:b3"
                        },
                        {
                            "name": "privateDnsName",
                            "value": "ip-10-0-1-184.ec2.internal"
                        },
                        {
                            "name": "privateIPv4Address",
                            "value": "10.0.1.184"
                        }
                    ]
                }
            ],
            "attributes": [
                {
                    "name": "ecs.cpu-architecture",
                    "value": "x86_64"
                }
            ],
            "availabilityZone": "us-east-1b",
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
            "connectivity": "CONNECTED",
            "connectivityAt": "2021-12-20T12:13:37.875000-05:00",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/
MyCluster/74de0355a10a4f979ac495c14EXAMPLE/aad3ba00-83b3-4dac-84d4-11f8cEXAMPLE",
```

```
                    "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/
MyCluster/74de0355a10a4f979ac495c14EXAMPLE",
                    "name": "web",
                    "image": "nginx",
                    "runtimeId": "74de0355a10a4f979ac495c14EXAMPLE-265927825",
                    "lastStatus": "RUNNING",
                    "networkBindings": [],
                    "networkInterfaces": [
                        {
                            "attachmentId": "d9e7735a-16aa-4128-bc7a-b2d51EXAMPLE",
                            "privateIpv4Address": "10.0.1.184"
                        }
                    ],
                    "healthStatus": "UNKNOWN",
                    "cpu": "99",
                    "memory": "100"
                }
            ],
            "cpu": "256",
            "createdAt": "2021-12-20T12:13:20.226000-05:00",
            "desiredStatus": "RUNNING",
            "enableExecuteCommand": false,
            "group": "service:tdsevicetag",
            "healthStatus": "UNKNOWN",
            "lastStatus": "RUNNING",
            "launchType": "FARGATE",
            "memory": "512",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "web"
                    }
                ],
                "inferenceAcceleratorOverrides": []
            },
            "platformVersion": "1.4.0",
            "platformFamily": "Linux",
            "pullStartedAt": "2021-12-20T12:13:42.665000-05:00",
            "pullStoppedAt": "2021-12-20T12:13:46.543000-05:00",
            "startedAt": "2021-12-20T12:13:48.086000-05:00",
            "startedBy": "ecs-svc/988401040018EXAMPLE",
            "tags": [],
            "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/
MyCluster/74de0355a10a4f979ac495c14EXAMPLE",
```

```
            "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-
definition/webserver:2",
            "version": 3,
            "ephemeralStorage": {
            "sizeInGiB": 20
            }
        },
        {
            "attachments": [
                {
                    "id": "214eb5a9-45cd-4bf8-87bc-57fefEXAMPLE",
                    "type": "ElasticNetworkInterface",
                    "status": "ATTACHED",
                    "details": [
                        {
                            "name": "subnetId",
                            "value": "subnet-0d0eab1bb3EXAMPLE"
                        },
                        {
                            "name": "networkInterfaceId",
                            "value": "eni-064c7766daEXAMPLE"
                        },
                        {
                            "name": "macAddress",
                            "value": "0e:76:83:01:17:a9"
                        },
                        {
                            "name": "privateDnsName",
                            "value": "ip-10-0-1-41.ec2.internal"
                        },
                        {
                            "name": "privateIPv4Address",
                            "value": "10.0.1.41"
                        }
                    ]
                }
            ],
            "attributes": [
                {
                    "name": "ecs.cpu-architecture",
                    "value": "x86_64"
                }
            ],
            "availabilityZone": "us-east-1b",
```

```
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
            "connectivity": "CONNECTED",
            "connectivityAt": "2021-12-20T12:13:35.243000-05:00",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/
MyCluster/d789e94343414c25b9f6bd59eEXAMPLE/9afef792-609b-43a5-bb6a-3efdbEXAMPLE",
                    "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/MyCluster/
d789e94343414c25b9f6bd59eEXAMPLE",
                    "name": "web",
                    "image": "nginx",
                    "runtimeId": "d789e94343414c25b9f6bd59eEXAMPLE-265927825",
                    "lastStatus": "RUNNING",
                    "networkBindings": [],
                    "networkInterfaces": [
                        {
                            "attachmentId": "214eb5a9-45cd-4bf8-87bc-57fefEXAMPLE",
                            "privateIpv4Address": "10.0.1.41"
                        }
                    ],
                    "healthStatus": "UNKNOWN",
                    "cpu": "99",
                    "memory": "100"
                }
            ],
            "cpu": "256",
            "createdAt": "2021-12-20T12:13:20.226000-05:00",
            "desiredStatus": "RUNNING",
            "enableExecuteCommand": false,
            "group": "service:tdsevicetag",
            "healthStatus": "UNKNOWN",
            "lastStatus": "RUNNING",
            "launchType": "FARGATE",
            "memory": "512",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "web"
                    }
                ],
                "inferenceAcceleratorOverrides": []
            },
            "platformVersion": "1.4.0",
            "platformFamily": "Linux",
```

```
            "pullStartedAt": "2021-12-20T12:13:44.611000-05:00",
            "pullStoppedAt": "2021-12-20T12:13:48.251000-05:00",
            "startedAt": "2021-12-20T12:13:49.326000-05:00",
            "startedBy": "ecs-svc/988401040018EXAMPLE",
            "tags": [],
            "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/MyCluster/
d789e94343414c25b9f6bd59eEXAMPLE",
            "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-
definition/webserver:2",
            "version": 3,
            "ephemeralStorage": {
                "sizeInGiB": 20
            }
        }
    ],
    "failures": []
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon ECS 任務定義。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTasks。

execute-command

以下程式碼範例顯示如何使用 execute-command。

AWS CLI

執行互動式 /bin/sh 命令

下列execute-command範例針對 ID 為 的任務，對名為 MyContainer 的容器執行互動式 /bin/sh 命令arn:aws:ecs:us-east-1:123456789012:task/MyCluster/d789e94343414c25b9f6bd59eEXAMPLE。

```
aws ecs execute-command \
    --cluster MyCluster \
    --task arn:aws:ecs:us-east-1:123456789012:task/MyCluster/
d789e94343414c25b9f6bd59eEXAMPLE \
    --container MyContainer \
    --interactive \
    --command "/bin/sh"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的使用 Amazon ECS Exec 進行偵錯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ExecuteCommand。

`get-task-protection`

以下程式碼範例顯示如何使用 `get-task-protection`。

AWS CLI

擷取 ECS 服務中任務的保護狀態

以下`get-task-protection`提供屬於 Amazon ECS 服務的 ECS 任務的保護狀態。

```
aws ecs get-task-protection \
    --cluster ECS-project-update-cluster \
    --tasks c43ed3b1331041f289316f958adb6a24
```

輸出：

```
{
    "protectedTasks": [
        {
            "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
c43ed3b1331041f289316f958adb6a24",
            "protectionEnabled": false
        }
    ],
    "failures": []
}
```

如需任務保護的更多形式，請參閱《Amazon ECS 開發人員指南》中的保護 Amazon ECS 任務不受縮減事件終止。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTaskProtection。

`list-account-settings`

以下程式碼範例顯示如何使用 `list-account-settings`。

AWS CLI

範例 1：檢視帳戶的帳戶設定

下列list-account-settings範例顯示 帳戶的有效帳戶設定。

```
aws ecs list-account-settings --effective-settings
```

輸出：

```
{
    "settings": [
        {
            "name": "containerInstanceLongArnFormat",
            "value": "enabled",
            "principalArn": "arn:aws:iam::123456789012:root"
        },
        {
            "name": "serviceLongArnFormat",
            "value": "enabled",
            "principalArn": "arn:aws:iam::123456789012:root"
        },
        {
            "name": "taskLongArnFormat",
            "value": "enabled",
            "principalArn": "arn:aws:iam::123456789012:root"
        }
    ]
}
```

範例 2：檢視特定 IAM 使用者或 IAM 角色的帳戶設定

下列list-account-settings範例顯示指定 IAM 使用者或 IAM 角色的帳戶設定。

```
aws ecs list-account-settings --principal-arn arn:aws:iam::123456789012:user/MyUser
```

輸出：

```
{
    "settings": [
        {
            "name": "serviceLongArnFormat",
            "value": "enabled",
            "principalArn": "arn:aws:iam::123456789012:user/MyUser"
        }
```

```
    ]
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon Resource Name (ARNs) 和 IDs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAccountSettings。

`list-attributes`

以下程式碼範例顯示如何使用 `list-attributes`。

AWS CLI

列出包含特定屬性的容器執行個體

下列範例列出預設叢集中具有 屬性之容器執行個體的`stack=production`屬性。

```
aws ecs list-attributes \
    --target-type container-instance \
    --attribute-name stack \
    --attribute-value production \
    --cluster default
```

輸出：

```
{
    "attributes": [
        {
            "name": "stack",
            "targetId": "arn:aws:ecs:us-west-2:130757420319:container-
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",
            "value": "production"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon ECS 容器代理程式組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAttributes。

list-clusters

以下程式碼範例顯示如何使用 list-clusters。

AWS CLI

列出可用的叢集

下列list-clusters範例列出所有可用的叢集。

```
aws ecs list-clusters
```

輸出：

```
{
    "clusterArns": [
        "arn:aws:ecs:us-west-2:123456789012:cluster/MyECSCluster1",
        "arn:aws:ecs:us-west-2:123456789012:cluster/AnotherECSCluster"
    ]
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon ECS 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListClusters。

list-container-instances

以下程式碼範例顯示如何使用 list-container-instances。

AWS CLI

列出叢集中的容器執行個體

下列list-container-instances範例列出叢集中所有可用的容器執行個體。

```
aws ecs list-container-instances --cluster MyCluster
```

輸出：

```
{
    "containerInstanceArns": [
        "arn:aws:ecs:us-west-2:123456789012:container-instance/MyCluster/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
```

```
        "arn:aws:ecs:us-west-2:123456789012:container-instance/MyCluster/
a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
    ]
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon ECS 容器執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListContainerInstances。

`list-service-deployments`

以下程式碼範例顯示如何使用 `list-service-deployments`。

AWS CLI

列出服務部署

下列`list-service-deployments`範例會擷取名為 之服務的服務部署`example-service`。

```
aws ecs list-service-deployments \
    --service arn:aws:ecs:us-east-1:123456789012:service/example-cluster/example-
service
```

輸出：

```
{
    "serviceDeployments": [
        {
            "serviceDeploymentArn": "arn:aws:ecs:us-east-1:123456789012:service-
deployment/example-cluster/example-service/ejGvqq2ilnbKT9qj0vLJe",
            "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/example-
cluster/example-service",
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/example-
cluster",
            "startedAt": "2024-10-31T08:03:32.510000-04:00",
            "createdAt": "2024-10-31T08:03:30.917000-04:00",
            "finishedAt": "2024-10-31T08:05:04.527000-04:00",
            "targetServiceRevisionArn": "arn:aws:ecs:us-east-1:123456789012:service-
revision/example-cluster/example-service/1485800978477494678",
            "status": "SUCCESSFUL"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的使用 Amazon ECS 服務部署檢視服務歷史記錄。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListServiceDeployments。

`list-services-by-namespace`

以下程式碼範例顯示如何使用 `list-services-by-namespace`。

AWS CLI

列出命名空間中的服務

下列`list-services-by-namespace`範例列出為預設區域中指定的命名空間設定的所有 服務。

```
aws ecs list-services-by-namespace \
    --namespace service-connect
```

輸出：

```
{
    "serviceArns": [
        "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/MyService",
        "arn:aws:ecs:us-west-2:123456789012:service/tutorial/service-connect-nginx-
service"
    ]
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Service Connect。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListServicesByNamespace。

`list-services`

以下程式碼範例顯示如何使用 `list-services`。

AWS CLI

列出叢集中的服務

下列`list-services`範例顯示如何列出叢集中執行的服務。

```
aws ecs list-services --cluster MyCluster
```

輸出：

```
 {
     "serviceArns": [
         "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/MyService"
     ]
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListServices。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出資源的標籤

下列list-tags-for-resource範例列出特定叢集的標籤。

```
aws ecs list-tags-for-resource \
    --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster
```

輸出：

```
{
    "tags": [
        {
            "key": "key1",
            "value": "value1"
        },
        {
            "key": "key2",
            "value": "value2"
        },
        {
            "key": "key3",
            "value": "value3"
```

```
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`list-task-definition-families`

以下程式碼範例顯示如何使用 `list-task-definition-families`。

AWS CLI

範例 1：列出已註冊的任務定義系列

下列`list-task-definition-families`範例列出所有已註冊的任務定義系列。

```
aws ecs list-task-definition-families
```

輸出：

```
{
    "families": [
        "node-js-app",
        "web-timer",
        "hpcc",
        "hpcc-c4-8xlarge"
    ]
}
```

範例 2：篩選已註冊的任務定義系列

下列`list-task-definition-families`範例列出以 "hpcc" 開頭的任務定義修訂。

```
aws ecs list-task-definition-families --family-prefix hpcc
```

輸出：

```
{
    "families": [
        "hpcc",
        "hpcc-c4-8xlarge"
```

```
    ]
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的任務定義參數。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTaskDefinitionFamilies。

`list-task-definitions`

以下程式碼範例顯示如何使用 `list-task-definitions`。

AWS CLI

範例 1：列出已註冊的任務定義

下列`list-task-definitions`範例列出所有已註冊的任務定義。

```
aws ecs list-task-definitions
```

輸出：

```
{
    "taskDefinitionArns": [
        "arn:aws:ecs:us-west-2:123456789012:task-definition/sleep300:2",
        "arn:aws:ecs:us-west-2:123456789012:task-definition/sleep360:1",
        "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:3",
        "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:4",
        "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:5",
        "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:6"
    ]
}
```

範例 2：列出系列中已註冊的任務定義

下列 list-task-definitions 範例會列出指定系列的任務定義修訂。

```
aws ecs list-task-definitions --family-prefix wordpress
```

輸出：

```
{
    "taskDefinitionArns": [
```

```
        "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:3",
        "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:4",
        "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:5",
        "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:6"
    ]
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon ECS 任務定義。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTaskDefinitions。

`list-tasks`

以下程式碼範例顯示如何使用 `list-tasks`。

AWS CLI

範例 1：列出叢集中的任務

下列`list-tasks`範例列出叢集中的所有任務。

```
aws ecs list-tasks --cluster default
```

輸出：

```
{
    "taskArns": [
        "arn:aws:ecs:us-west-2:123456789012:task/a1b2c3d4-5678-90ab-
cdef-11111EXAMPLE",
        "arn:aws:ecs:us-west-2:123456789012:task/a1b2c3d4-5678-90ab-
cdef-22222EXAMPLE"
    ]
}
```

範例 2：列出特定容器執行個體上的任務

下列`list-tasks`範例使用容器執行個體 UUID 做為篩選條件，列出容器執行個體上的任務。

```
aws ecs list-tasks --cluster default --container-instance a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE
```

輸出：

```
{
    "taskArns": [
        "arn:aws:ecs:us-west-2:123456789012:task/a1b2c3d4-5678-90ab-
cdef-44444EXAMPLE"
    ]
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon ECS 任務定義。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTasks。

put-account-setting-default

以下程式碼範例顯示如何使用 put-account-setting-default。

AWS CLI

修改預設帳戶設定

下列put-account-setting-default範例會修改您帳戶中所有 IAM 使用者或角色的預設帳戶設定。除非 IAM 使用者或角色自行明確覆寫這些設定，否則這些變更會套用至整個 AWS 帳戶。

```
aws ecs put-account-setting-default --name serviceLongArnFormat --value enabled
```

輸出：

```
{
    "setting": {
        "name": "serviceLongArnFormat",
        "value": "enabled",
        "principalArn": "arn:aws:iam::123456789012:root"
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的 Amazon Resource Name (ARNs) 和 IDs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutAccountSettingDefault。

put-account-setting

以下程式碼範例顯示如何使用 put-account-setting。

AWS CLI

修改 IAM 使用者帳戶的帳戶設定

下列put-account-setting範例會將 containerInsightsIAM 使用者帳戶enhanced的帳戶設定設為 。這會開啟具有增強可觀測性的 Container Insights。

```
aws ecs put-account-setting \
    --name containerInsights \
    --value enhanced
```

輸出：

```
{
    "setting": {
        "name": "containerInsights",
        "value": "enhanced",
        "principalArn": "arn:aws:iam::123456789012:user/johndoe",
        "type": "user"
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的修改帳戶設定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutAccountSetting。

put-account-settings

以下程式碼範例顯示如何使用 put-account-settings。

AWS CLI

修改 IAM 使用者或 IAM 角色的帳戶設定

下列put-account-setting範例會修改指定 IAM 使用者或 IAM 角色的帳戶設定。

```
aws ecs put-account-setting \
    --name serviceLongArnFormat \
    --value enabled \
    --principal-arn arn:aws:iam::123456789012:user/MyUser
```

輸出：

```
{
    "setting": {
        "name": "serviceLongArnFormat",
        "value": "enabled",
        "principalArn": "arn:aws:iam::123456789012:user/MyUser"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutAccountSettings。

put-attributes

以下程式碼範例顯示如何使用 put-attributes。

AWS CLI

建立屬性並將其與 Amazon ECS 資源建立關聯

下列會將具有名稱堆疊和值生產的 屬性put-attributes套用至容器執行個體。

```
aws ecs put-attributes \
    --attributes name=stack,value=production,targetId=arn:aws:ecs:us-
west-2:130757420319:container-instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34
```

輸出：

```
{
    "attributes": [
        {
            "name": "stack",
            "targetId": "arn:aws:ecs:us-west-2:130757420319:container-
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",
            "value": "production"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutAttributes。

put-cluster-capacity-providers

以下程式碼範例顯示如何使用 put-cluster-capacity-providers。

AWS CLI

範例 1：將現有容量提供者新增至叢集

下列put-cluster-capacity-providers範例會將現有的容量提供者新增至叢集。create-capacity-provider 命令用於建立容量提供者。describe-clusters 命令用於描述目前容量提供者，以及與叢集相關聯的預設容量提供者策略。將新的容量提供者新增至叢集時，除了要與叢集建立關聯的新容量提供者之外，您還必須指定所有現有的容量提供者。您還必須指定要與叢集建立關聯的預設容量提供者策略。在此範例中，MyCluster叢集具有與其相關聯的MyCapacityProvider1容量提供者，而且您想要新增MyCapacityProvider2容量提供者，並將其包含在預設容量提供者策略中，以便將任務平均分散到兩個容量提供者。

```
aws ecs put-cluster-capacity-providers \
    --cluster MyCluster \
    --capacity-providers MyCapacityProvider1 MyCapacityProvider2 \
    --default-capacity-provider-
strategy capacityProvider=MyCapacityProvider1,weight=1 capacityProvider=MyCapacityProvider2,
```

輸出：

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "clusterName": "MyCluster",
        "status": "ACTIVE",
        "registeredContainerInstancesCount": 0,
        "runningTasksCount": 0,
        "pendingTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "tags": [],
        "settings": [
            {
                "name": "containerInsights",
                "value": "enabled"
            }
        ],
        "capacityProviders": [
```

```
            "MyCapacityProvider1",
            "MyCapacityProvider2"
        ],
        "defaultCapacityProviderStrategy": [
            {
                "capacityProvider": "MyCapacityProvider1",
                "weight": 1,
                "base": 0
            },
            {
                "capacityProvider": "MyCapacityProvider2",
                "weight": 1,
                "base": 0
            }
        ],
        "attachments": [
           {
                "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
                "type": "as_policy",
                "status": "ACTIVE",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider1"
                    },
                    {
                        "name": "scalingPolicyName",
                        "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111"
                    }
                ]
            },
            {
                "id": "ae592060-2382-4663-9476-b015c685593c",
                "type": "as_policy",
                "status": "ACTIVE",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider2"
                    },
                    {
                        "name": "scalingPolicyName",
```

```
                        "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222"
                    }
                ]
            }
        ],
        "attachmentsStatus": "UPDATE_IN_PROGRESS"
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的叢集容量提供者。

範例 2：從叢集中移除容量提供者

下列put-cluster-capacity-providers範例會從叢集中移除容量提供者。describe-clusters 命令用於描述與叢集相關聯的目前容量提供者。從叢集移除容量提供者時，您必須指定要與叢集保持關聯的容量提供者，以及要與叢集建立關聯的預設容量提供者策略。在此範例中，叢集具有與其相關聯的 MyCapacityProvider1 和MyCapacityProvider2容量提供者，而且您想要移除MyCapacityProvider2容量提供者，因此您只能在 命令MyCapacityProvider1中指定 ，以及更新的預設容量提供者策略。

```
aws ecs put-cluster-capacity-providers \
    --cluster MyCluster \
    --capacity-providers MyCapacityProvider1 \
    --default-capacity-provider-
strategy capacityProvider=MyCapacityProvider1,weight=1,base=0
```

輸出：

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "clusterName": "MyCluster",
        "status": "ACTIVE",
        "registeredContainerInstancesCount": 0,
        "runningTasksCount": 0,
        "pendingTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "tags": [],
        "settings": [
            {
```

```
                "name": "containerInsights",
                "value": "enabled"
            }
        ],
        "capacityProviders": [
            "MyCapacityProvider1"
        ],
        "defaultCapacityProviderStrategy": [
            "capacityProvider": "MyCapacityProvider1",
            "weight": 1,
            "base": 0
        ],
        "attachments": [
           {
                "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
                "type": "as_policy",
                "status": "ACTIVE",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider1"
                    },
                    {
                        "name": "scalingPolicyName",
                        "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111"
                    }
                ]
            },
            {
                "id": "ae592060-2382-4663-9476-b015c685593c",
                "type": "as_policy",
                "status": "DELETING",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider2"
                    },
                    {
                        "name": "scalingPolicyName",
                        "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222"
                    }
                ]
```

```
            }
        ],
        "attachmentsStatus": "UPDATE_IN_PROGRESS"
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的叢集容量提供者。

範例 3：從叢集中移除所有容量提供者

下列put-cluster-capacity-providers範例會從叢集中移除所有現有的容量提供者。

```
aws ecs put-cluster-capacity-providers \
    --cluster MyCluster \
    --capacity-providers [] \
    --default-capacity-provider-strategy []
```

輸出：

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "clusterName": "MyCluster",
        "status": "ACTIVE",
        "registeredContainerInstancesCount": 0,
        "runningTasksCount": 0,
        "pendingTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "tags": [],
        "settings": [
            {
                "name": "containerInsights",
                "value": "enabled"
            }
        ],
        "capacityProviders": [],
        "defaultCapacityProviderStrategy": [],
        "attachments": [
           {
                "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
                "type": "as_policy",
                "status": "DELETING",
```

```
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider1"
                    },
                    {
                        "name": "scalingPolicyName",
                        "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111"
                    }
                ]
            },
            {
                "id": "ae592060-2382-4663-9476-b015c685593c",
                "type": "as_policy",
                "status": "DELETING",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider2"
                    },
                    {
                        "name": "scalingPolicyName",
                        "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222"
                    }
                ]
            }
        ],
        "attachmentsStatus": "UPDATE_IN_PROGRESS"
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的叢集容量提供者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutClusterCapacityProviders。

register-task-definition

以下程式碼範例顯示如何使用 register-task-definition。

AWS CLI

範例 1：向 JSON 檔案註冊任務定義

下列register-task-definition範例會將任務定義註冊到指定的系列。容器定義會以 JSON 格式儲存在指定的檔案位置。

```
aws ecs register-task-definition \
    --cli-input-json file://<path_to_json_file>/sleep360.json
```

sleep360.json 的內容：

```
{
    "containerDefinitions": [
        {
            "name": "sleep",
            "image": "busybox",
            "cpu": 10,
            "command": [
                "sleep",
                "360"
            ],
            "memory": 10,
            "essential": true
        }
    ],
    "family": "sleep360"
}
```

輸出：

```
{
    "taskDefinition": {
        "status": "ACTIVE",
        "family": "sleep360",
        "placementConstraints": [],
        "compatibilities": [
                "EXTERNAL",
                "EC2"
        ],
        "volumes": [],
        "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/
sleep360:1",
        "containerDefinitions": [
            {
                "environment": [],
```

```
                "name": "sleep",
                "mountPoints": [],
                "image": "busybox",
                "cpu": 10,
                "portMappings": [],
                "command": [
                    "sleep",
                    "360"
                ],
            "memory": 10,
            "essential": true,
            "volumesFrom": []
            }
        ],
            "revision": 1
        }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的範例任務定義。

範例 2：使用 JSON 字串參數註冊任務定義

下列register-task-definition範例使用以 JSON 字串參數提供的容器定義，搭配逸出雙引號來註冊任務定義。

```
aws ecs register-task-definition \
    --family sleep360 \
    --container-definitions "[{\"name\":\"sleep\",\"image\":\"busybox\",\"cpu\":10,
\"command\":[\"sleep\",\"360\"],\"memory\":10,\"essential\":true}]"
```

輸出與上一個範例相同。

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的建立任務定義。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterTaskDefinition。

run-task

以下程式碼範例顯示如何使用 run-task。

AWS CLI

範例 1：在預設叢集上執行任務

下列run-task範例會在預設叢集上執行任務，並使用用戶端字符。

```
aws ecs run-task \
    --cluster default \
    --task-definition sleep360:1 \
    --client-token 550e8400-e29b-41d4-a716-446655440000
```

輸出：

```
{
    "tasks": [
        {
            "attachments": [],
            "attributes": [
                {
                    "name": "ecs.cpu-architecture",
                    "value": "x86_64"
                }
            ],
            "availabilityZone": "us-east-1b",
            "capacityProviderName": "example-capacity-provider",
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/default",
            "containerInstanceArn": "arn:aws:ecs:us-east-1:123456789012:container-
instance/default/bc4d2ec611d04bb7bb97e83ceEXAMPLE",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/
default/d6f51cc5bbc94a47969c92035e9f66f8/75853d2d-711e-458a-8362-0f0aEXAMPLE",
                    "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/default/
d6f51cc5bbc94a47969c9203EXAMPLE",
                    "name": "sleep",
                    "image": "busybox",
                    "lastStatus": "PENDING",
                    "networkInterfaces": [],
                    "cpu": "10",
                    "memory": "10"
                }
            ],
            "cpu": "10",
            "createdAt": "2023-11-21T16:59:34.403000-05:00",
            "desiredStatus": "RUNNING",
            "enableExecuteCommand": false,
            "group": "family:sleep360",
```

```
            "lastStatus": "PENDING",
            "launchType": "EC2",
            "memory": "10",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "sleep"
                    }
                ],
                "inferenceAcceleratorOverrides": []
            },
            "tags": [],
            "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/default/
d6f51cc5bbc94a47969c9203EXAMPLE",
            "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-
definition/sleep360:1",
            "version": 1
        }
    ],
    "failures": []
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的將應用程式作為獨立任務執行。

範例 2：為獨立任務設定 Amazon EBS 磁碟區

下列run-task範例會為預設叢集上的 Fargate 任務設定加密的 Amazon EBS 磁碟區。您必須將 Amazon ECS 基礎設施角色設定為已連接AmazonECSInfrastructureRolePolicyForVolumes受管政策。您必須指定具有與run-task請求中相同磁碟區名稱的任務定義。此範例使用 --cli-input-json選項和名為 的 JSON 輸入檔案ebs.json。

```
aws ecs run-task \
    --cli-input-json file://ebs.json
```

ebs.json 的內容：

```
{
   "cluster": "default",
   "taskDefinition": "mytaskdef",
   "launchType": "FARGATE",
   "networkConfiguration":{
```

```
        "awsvpcConfiguration":{
            "assignPublicIp": "ENABLED",
            "securityGroups": ["sg-12344321"],
            "subnets":["subnet-12344321"]
        }
    },
   "volumeConfigurations": [
        {
            "name": "myEBSVolume",
            "managedEBSVolume": {
                "volumeType": "gp3",
                "sizeInGiB": 100,
                "roleArn":"arn:aws:iam::1111222333:role/ecsInfrastructureRole",
                "encrypted": true,
                "kmsKeyId":
 "arn:aws:kms:region:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
            }
        }
    ]
}
```

輸出：

```
{
    "tasks": [
        {
            "attachments": [
                {
                    "id": "ce868693-15ca-4083-91ac-f782f64000c9",
                    "type": "ElasticNetworkInterface",
                    "status": "PRECREATED",
                    "details": [
                        {
                        "name": "subnetId",
                        "value": "subnet-070982705451dad82"
                        }
                    ]
                },
                {
                    "id": "a17ed863-786c-4372-b5b3-b23e53f37877",
                    "type": "AmazonElasticBlockStorage",
                    "status": "CREATED",
                    "details": [
```

```
                        {
                            "name": "roleArn",
                            "value": "arn:aws:iam::123456789012:role/
ecsInfrastructureRole"
                        },
                        {
                            "name": "volumeName",
                            "value": "myEBSVolume"
                        },
                        {
                            "name": "deleteOnTermination",
                            "value": "true"
                        }
                    ]
                }
            ],
            "attributes": [
                {
                    "name": "ecs.cpu-architecture",
                    "value": "x86_64"
                }
            ],
            "availabilityZone": "us-west-2b",
            "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-west-2:123456789012:container/
default/7f1fbd3629434cc4b82d72d2f09b67c9/e21962a2-f328-4699-98a3-5161ac2c186a",
                    "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
default/7f1fbd3629434cc4b82d72d2f09b67c9",
                    "name": "container-using-ebs",
                    "image": "amazonlinux:2",
                    "lastStatus": "PENDING",
                    "networkInterfaces": [],
                    "cpu": "0"
                }
            ],
            "cpu": "1024",
            "createdAt": "2025-01-23T10:29:46.650000-06:00",
            "desiredStatus": "RUNNING",
            "enableExecuteCommand": false,
            "group": "family:mytaskdef",
            "lastStatus": "PROVISIONING",
            "launchType": "FARGATE",
```

```
            "memory": "3072",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "container-using-ebs"
                    }
                ],
                "inferenceAcceleratorOverrides": []
            },
            "platformVersion": "1.4.0",
            "platformFamily": "Linux",
            "tags": [],
            "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
default/7f1fbd3629434cc4b82d72d2f09b67c9",
            "taskDefinitionArn": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:4",
            "version": 1,
            "ephemeralStorage": {
                "sizeInGiB": 20
            },
            "fargateEphemeralStorage": {
                "sizeInGiB": 20
            }
        }
    ],
    "failures": []
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的搭配使用 Amazon EBS 磁碟區與 Amazon ECS。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RunTask。

start-task

以下程式碼範例顯示如何使用 start-task。

AWS CLI

範例 1：啟動新任務

下列start-task範例使用預設叢集中指定容器執行個體上任務定義的最新版本來啟動sleep360任務。

```
aws ecs start-task \
    --task-definition sleep360 \
    --container-instances 765936fadbdd46b5991a4bd70c2a43d4
```

輸出：

```
{
    "tasks": [
        {
            "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
default/666fdccc2e2d4b6894dd422f4eeee8f8",
            "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default",
            "taskDefinitionArn": "arn:aws:ecs:us-west-2:123456789012:task-
definition/sleep360:3",
            "containerInstanceArn": "arn:aws:ecs:us-west-2:123456789012:container-
instance/default/765936fadbdd46b5991a4bd70c2a43d4",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "sleep"
                    }
                ]
            },
            "lastStatus": "PENDING",
            "desiredStatus": "RUNNING",
            "cpu": "128",
            "memory": "128",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-
west-2:123456789012:container/75f11ed4-8a3d-4f26-a33b-ad1db9e02d41",
                    "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
default/666fdccc2e2d4b6894dd422f4eeee8f8",
                    "name": "sleep",
                    "lastStatus": "PENDING",
                    "networkInterfaces": [],
                    "cpu": "10",
                    "memory": "10"
                }
            ],
            "version": 1,
            "createdAt": 1563421494.186,
            "group": "family:sleep360",
```

```
            "launchType": "EC2",
            "attachments": [],
            "tags": []
        }
    ],
    "failures": []
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的在 Amazon ECS 上排程容器。

範例 2：在任務開始時設定 Amazon EBS 磁碟區

下列start-task範例會為指定容器執行個體上的任務設定加密的 Amazon EBS 磁碟區。您必須將 Amazon ECS 基礎設施角色設定為已連接AmazonECSInfrastructureRolePolicyForVolumes受管政策。您必須指定具有與start-task請求中相同磁碟區名稱的任務定義。此範例使用 --cli-input-json選項和名為 的 JSON 輸入檔案ebs.json，內容如下。

```
aws ecs start-task \
    --cli-input-json file://ebs.json \
    --container-instances 765936fadbdd46b5991a4bd70c2a43d4
```

ebs.json 的內容：

```
{
   "cluster": "default",
   "taskDefinition": "mytaskdef",
   "networkConfiguration":{
        "awsvpcConfiguration":{
            "assignPublicIp": "ENABLED",
            "securityGroups": ["sg-12344321"],
            "subnets":["subnet-12344321"]
        }
    },
   "volumeConfigurations": [
        {
            "name": "myEBSVolume",
            "managedEBSVolume": {
                "volumeType": "gp3",
                "sizeInGiB": 100,
                "roleArn":"arn:aws:iam::123456789012:role/ecsInfrastructureRole",
                "encrypted": true,
```

```
                "kmsKeyId":
 "arn:aws:kms:region:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
            }
        }
    ]
}
```

輸出：

```
{
    "tasks": [
        {
            "attachments": [
                {
                    "id": "aea29489-9dcd-49f1-8164-4d91566e1113",
                    "type": "ElasticNetworkInterface",
                    "status": "PRECREATED",
                    "details": [
                        {
                            "name": "subnetId",
                            "value": "subnet-12344321"
                        }
                    ]
                },
                {
                    "id": "f29e1222-9a1e-410f-b499-a12a7cd6d42e",
                    "type": "AmazonElasticBlockStorage",
                    "status": "CREATED",
                    "details": [
                        {
                            "name": "roleArn",
                            "value": "arn:aws:iam::123456789012:role/
ecsInfrastructureRole"
                        },
                        {
                            "name": "volumeName",
                            "value": "myEBSVolume"
                        },
                        {
                            "name": "deleteOnTermination",
                            "value": "true"
                        }
                    ]
```

```
                }
            ],
            "attributes": [
                {
                    "name": "ecs.cpu-architecture",
                    "value": "arm64"
                }
            ],
            "availabilityZone": "us-west-2c",
            "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default",
            "containerInstanceArn": "arn:aws:ecs:us-west-2:123456789012:container-
instance/default/765936fadbdd46b5991a4bd70c2a43d4",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-west-2:123456789012:container/
default/bb122ace3ed84add92c00a351a03c69e/a4a9ed10-51c7-4567-9653-50e71b94f867",
                    "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/default/
bb122ace3ed84add92c00a351a03c69e",
                    "name": "container-using-ebs",
                    "image": "amazonlinux:2",
                    "lastStatus": "PENDING",
                    "networkInterfaces": [],
                    "cpu": "0"
                }
            ],
            "cpu": "1024",
            "createdAt": "2025-01-23T14:51:05.191000-06:00",
            "desiredStatus": "RUNNING",
            "enableExecuteCommand": false,
            "group": "family:mytaskdef",
            "lastStatus": "PROVISIONING",
            "launchType": "EC2",
            "memory": "3072",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "container-using-ebs"
                    }
                ],
                "inferenceAcceleratorOverrides": []
            },
             "tags": [],
            "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/default/
bb122ace3ed84add92c00a351a03c69e",
```

```
            "taskDefinitionArn": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:4",
            "version": 1
        }
    ],
    "failures": []
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的搭配使用 Amazon EBS 磁碟區與 Amazon ECS。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartTask。

stop-task

以下程式碼範例顯示如何使用 stop-task。

AWS CLI

停止任務

以下內容會停止指定的任務stop-task在預設叢集中執行。

```
aws ecs stop-task \
    --task 666fdccc2e2d4b6894dd422f4eeee8f8
```

輸出：

```
{
    "task": {
        "taskArn": "arn:aws:ecs:us-west-2:130757420319:task/
default/666fdccc2e2d4b6894dd422f4eeee8f8",
        "clusterArn": "arn:aws:ecs:us-west-2:130757420319:cluster/default",
        "taskDefinitionArn": "arn:aws:ecs:us-west-2:130757420319:task-definition/
sleep360:3",
        "containerInstanceArn": "arn:aws:ecs:us-west-2:130757420319:container-
instance/default/765936fadbdd46b5991a4bd70c2a43d4",
        "overrides": {
            "containerOverrides": []
        },
        "lastStatus": "STOPPED",
        "desiredStatus": "STOPPED",
        "cpu": "128",
```

```
        "memory": "128",
        "containers": [],
        "version": 2,
        "stoppedReason": "Taskfailedtostart",
        "stopCode": "TaskFailedToStart",
        "connectivity": "CONNECTED",
        "connectivityAt": 1563421494.186,
        "pullStartedAt": 1563421494.252,
        "pullStoppedAt": 1563421496.252,
        "executionStoppedAt": 1563421497,
        "createdAt": 1563421494.186,
        "stoppingAt": 1563421497.252,
        "stoppedAt": 1563421497.252,
        "group": "family:sleep360",
        "launchType": "EC2",
        "attachments": [],
        "tags": []
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopTask。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

標記資源

下列tag-resource範例會將單一標籤新增至指定的資源。

```
aws ecs tag-resource \
    --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster
    --tags key=key1,value=value1
```

此命令不會產生輸出。

將多個標籤新增至資源

下列tag-resource範例會將多個標籤新增至指定的資源。

```
aws ecs tag-resource \
```

```
--resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \
--tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源移除標籤

下列untag-resource範例會從指定的資源移除列出的標籤。

```
aws ecs untag-resource \
    --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \
    --tag-keys key1,key2
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-cluster-settings

以下程式碼範例顯示如何使用 update-cluster-settings。

AWS CLI

修改叢集的設定

下列update-cluster-settings範例可讓 CloudWatch Container Insights 增強MyCluster叢集的可觀測性。

```
aws ecs update-cluster-settings \
    --cluster MyCluster \
    --settings name=containerInsights,value=enhanced
```

輸出：

```
{
```

```
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-esat-1:123456789012:cluster/MyCluster",
        "clusterName": "default",
        "status": "ACTIVE",
        "registeredContainerInstancesCount": 0,
        "runningTasksCount": 0,
        "pendingTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "tags": [],
        "settings": [
            {
                "name": "containerInsights",
                "value": "enhanced"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的修改帳戶設定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateClusterSettings。

update-cluster

以下程式碼範例顯示如何使用 update-cluster。

AWS CLI

範例 1：更新啟用 containerInsights 的 ECS 叢集

以下內容會將 containerInsights 值update-cluster更新為已建立叢集enabled中的 。預設為停用。

```
aws ecs update-cluster \
    --cluster ECS-project-update-cluster \
    --settings name=containerInsights,value=enabled
```

輸出：

```
"cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/ECS-project-update-
cluster",
```

```
    "clusterName": "ECS-project-update-cluster",
    "status": "ACTIVE",
    "registeredContainerInstancesCount": 0,
    "runningTasksCount": 0,
    "pendingTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "tags": [],
    "settings": [
        {
            "name": "containerInsights",
            "value": "enabled"
        }
    ],
    "capacityProviders": [
        "Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt"
    ],
    "defaultCapacityProviderStrategy": [
        {
            "capacityProvider": "Infra-ECS-Cluster-ECS-project-update-cluster-
d6bb6d5b-EC2CapacityProvider-3fIpdkLywwFt",
            "weight": 1,
            "base": 0
        }
    ],
    "attachments": [
        {
            "id": "069d002b-7634-42e4-b1d4-544f4c8f6380",
            "type": "as_policy",
            "status": "CREATED",
            "details": [
                {
                    "name": "capacityProviderName",
                    "value": "Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt"
                },
                {
                    "name": "scalingPolicyName",
                    "value": "ECSManagedAutoScalingPolicy-152363a6-8c65-484c-
b721-42c3e070ae93"
                }
            ]
        },
```

```
        {
            "id": "08b5b6ca-45e9-4209-a65d-e962a27c490a",
            "type": "managed_draining",
            "status": "CREATED",
            "details": [
                {
                    "name": "capacityProviderName",
                    "value": "Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt"
                },
                {
                    "name": "autoScalingLifecycleHookName",
                    "value": "ecs-managed-draining-termination-hook"
                }
            ]
        },
        {
            "id": "45d0b36f-8cff-46b6-9380-1288744802ab",
            "type": "sc",
            "status": "ATTACHED",
            "details": []
        }
    ],
    "attachmentsStatus": "UPDATE_COMPLETE",
    "serviceConnectDefaults": {
        "namespace": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
igwrsylmy3kwvcdx"
    }
}
```

範例 2：更新 ECS 叢集以設定預設 Service Connect namspace

以下透過設定預設 Service Connect 命名空間來`update-cluster`更新 ECS 叢集。

```
aws ecs update-cluster \
    --cluster ECS-project-update-cluster \
    --service-connect-defaults namespace=test
```

輸出：

```
 {
     "cluster": {
```

```
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/ECS-project-
update-cluster",
        "clusterName": "ECS-project-update-cluster",
        "status": "ACTIVE",
        "registeredContainerInstancesCount": 0,
        "runningTasksCount": 0,
        "pendingTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "tags": [],
        "settings": [
            {
                "name": "containerInsights",
                "value": "enabled"
            }
        ],
        "capacityProviders": [
            "Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt"
        ],
        "defaultCapacityProviderStrategy": [
            {
                "capacityProvider": "Infra-ECS-Cluster-ECS-project-update-cluster-
d6bb6d5b-EC2CapacityProvider-3fIpdkLywwFt",
                "weight": 1,
                "base": 0
            }
        ],
        "attachments": [
            {
                "id": "069d002b-7634-42e4-b1d4-544f4c8f6380",
                "type": "as_policy",
                "status": "CREATED",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "Infra-ECS-Cluster-ECS-project-update-cluster-
d6bb6d5b-EC2CapacityProvider-3fIpdkLywwFt"
                    },
                    {
                        "name": "scalingPolicyName",
                        "value": "ECSManagedAutoScalingPolicy-152363a6-8c65-484c-
b721-42c3e070ae93"
                    }
```

```
                ]
            },
            {
                "id": "08b5b6ca-45e9-4209-a65d-e962a27c490a",
                "type": "managed_draining",
                "status": "CREATED",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "Infra-ECS-Cluster-ECS-project-update-cluster-
d6bb6d5b-EC2CapacityProvider-3fIpdkLywwFt"
                    },
                    {
                        "name": "autoScalingLifecycleHookName",
                        "value": "ecs-managed-draining-termination-hook"
                    }
                ]
            },
            {
                "id": "45d0b36f-8cff-46b6-9380-1288744802ab",
                "type": "sc",
                "status": "DELETED",
                "details": []
            },
            {
                "id": "3e6890c3-609c-4832-91de-d6ca891b3ef1",
                "type": "sc",
                "status": "ATTACHED",
                "details": []
            },
            {
                "id": "961b8ec1-c2f1-4070-8495-e669b7668e90",
                "type": "sc",
                "status": "DELETED",
                "details": []
            }
        ],
        "attachmentsStatus": "UPDATE_COMPLETE",
        "serviceConnectDefaults": {
            "namespace": "arn:aws:servicediscovery:us-
west-2:123456789012:namespace/ns-dtjmxqpfi46ht7dr"
        }
    }
```

```
}
```

如需 Service Connect 的詳細資訊，請參閱《Amazon ECS 開發人員指南》中的使用 Service Connect 連接具有簡短名稱的 Amazon ECS 服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateCluster。

update-container-agent

以下程式碼範例顯示如何使用 update-container-agent。

AWS CLI

更新 Amazon ECS 容器執行個體上的容器代理程式

下列update-container-agent範例會更新預設叢集中指定容器執行個體上的容器代理程式。

```
aws ecs update-container-agent --cluster default --container-
instance a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

輸出：

```
{
    "containerInstance": {
        "status": "ACTIVE",
...
        "agentUpdateStatus": "PENDING",
        "versionInfo": {
            "agentVersion": "1.0.0",
            "agentHash": "4023248",
            "dockerVersion": "DockerVersion: 1.5.0"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的更新 Amazon ECS 容器代理程式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateContainerAgent。

update-container-instances-state

以下程式碼範例顯示如何使用 update-container-instances-state。

AWS CLI

更新容器執行個體的狀態

下列 會update-container-instances-state更新指定容器執行個體的狀態DRAINING，該執行個體會將其從註冊的叢集中移除。

```
aws ecs update-container-instances-state \
    --container-instances 765936fadbdd46b5991a4bd70c2a43d4 \
    --status DRAINING
```

輸出：

```
{
    "containerInstances": [
        {
            "containerInstanceArn": "arn:aws:ecs:us-west-2:130757420319:container-
instance/default/765936fadbdd46b5991a4bd70c2a43d4",
            "ec2InstanceId": "i-013d87ffbb4d513bf",
            "version": 4390,
            "versionInfo": {
                "agentVersion": "1.29.0",
                "agentHash": "a190a73f",
                "dockerVersion": "DockerVersion:18.06.1-ce"
            },
            "remainingResources": [
                {
                    "name": "CPU",
                    "type": "INTEGER",
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 1536
                },
                {
                    "name": "MEMORY",
                    "type": "INTEGER",
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 2681
                },
                {
                    "name": "PORTS",
                    "type": "STRINGSET",
```

```
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 0,
                    "stringSetValue": [
                        "22",
                        "2376",
                        "2375",
                        "51678",
                        "51679"
                    ]
                },
                {
                    "name": "PORTS_UDP",
                    "type": "STRINGSET",
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 0,
                    "stringSetValue": []
                }
            ],
            "registeredResources": [
                {
                    "name": "CPU",
                    "type": "INTEGER",
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 2048
                },
                {
                    "name": "MEMORY",
                    "type": "INTEGER",
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 3705
                },
                {
                    "name": "PORTS",
                    "type": "STRINGSET",
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 0,
                    "stringSetValue": [
                        "22",
                        "2376",
```

```
                    "2375",
                    "51678",
                    "51679"
                ]
            },
            {
                "name": "PORTS_UDP",
                "type": "STRINGSET",
                "doubleValue": 0,
                "longValue": 0,
                "integerValue": 0,
                "stringSetValue": []
            }
        ],
        "status": "DRAINING",
        "agentConnected": true,
        "runningTasksCount": 2,
        "pendingTasksCount": 0,
        "attributes": [
            {
                "name": "ecs.capability.secrets.asm.environment-variables"
            },
            {
                "name": "ecs.capability.branch-cni-plugin-version",
                "value": "e0703516-"
            },
            {
                "name": "ecs.ami-id",
                "value": "ami-00e0090ac21971297"
            },
            {
                "name": "ecs.capability.secrets.asm.bootstrap.log-driver"
            },
            {
                "name": "com.amazonaws.ecs.capability.logging-driver.none"
            },
            {
                "name": "ecs.capability.ecr-endpoint"
            },
            {
                "name": "ecs.capability.docker-plugin.local"
            },
            {
                "name": "ecs.capability.task-cpu-mem-limit"
```

```
                },
                {
                    "name": "ecs.capability.secrets.ssm.bootstrap.log-driver"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.30"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.31"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.32"
                },
                {
                    "name": "ecs.availability-zone",
                    "value": "us-west-2c"
                },
                {
                    "name": "ecs.capability.aws-appmesh"
                },
                {
                    "name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.24"
                },
                {
                    "name": "ecs.capability.task-eni-trunking"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.25"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.26"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.27"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.28"
                },
                {
                    "name": "com.amazonaws.ecs.capability.privileged-container"
                },
```

```
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.29"
                },
                {
                    "name": "ecs.cpu-architecture",
                    "value": "x86_64"
                },
                {
                    "name": "com.amazonaws.ecs.capability.ecr-auth"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.20"
                },
                {
                    "name": "ecs.os-type",
                    "value": "linux"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.21"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.22"
                },
                {
                    "name": "ecs.capability.task-eia"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.23"
                },
                {
                    "name": "ecs.capability.private-registry-
authentication.secretsmanager"
                },
                {
                    "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
                },
                {
                    "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
                },
                {
                    "name": "ecs.capability.execution-role-awslogs"
                },
                {
                    "name": "ecs.vpc-id",
```

```
                    "value": "vpc-1234"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
                },
                {
                    "name": "ecs.capability.task-eni"
                },
                {
                    "name": "ecs.capability.execution-role-ecr-pull"
                },
                {
                    "name": "ecs.capability.container-health-check"
                },
                {
                    "name": "ecs.subnet-id",
                    "value": "subnet-1234"
                },
                {
                    "name": "ecs.instance-type",
                    "value": "c5.large"
                },
                {
                    "name": "com.amazonaws.ecs.capability.task-iam-role-network-
host"
                },
                {
                    "name": "ecs.capability.container-ordering"
                },
                {
                    "name": "ecs.capability.cni-plugin-version",
                    "value": "91ccefc8-2019.06.0"
                },
                {
                    "name": "ecs.capability.pid-ipc-namespace-sharing"
                },
                {
                    "name": "ecs.capability.secrets.ssm.environment-variables"
```

```
                },
                {
                    "name": "com.amazonaws.ecs.capability.task-iam-role"
                }
            ],
            "registeredAt": 1560788724.507,
            "attachments": [],
            "tags": []
        }
    ],
    "failures": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateContainerInstancesState。

update-service-primary-task-set

以下程式碼範例顯示如何使用 update-service-primary-task-set。

AWS CLI

更新服務的主要任務集

下列update-service-primary-task-set範例會更新指定服務的主要任務集。

```
aws ecs update-service-primary-task-set \
    --cluster MyCluster \
    --service MyService \
    --primary-task-set arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/
MyService/ecs-svc/1234567890123456789
```

輸出：

```
{
    "taskSet": {
        "id": "ecs-svc/1234567890123456789",
        "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/
MyService/ecs-svc/1234567890123456789",
        "status": "PRIMARY",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
sample-fargate:2",
        "computedDesiredCount": 1,
```

```
        "pendingCount": 0,
        "runningCount": 0,
        "createdAt": 1557128360.711,
        "updatedAt": 1557129412.653,
        "launchType": "EC2",
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321"
                ],
                "securityGroups": [
                    "sg-12344312"
                ],
                "assignPublicIp": "DISABLED"
            }
        },
        "loadBalancers": [],
        "serviceRegistries": [],
        "scale": {
            "value": 50.0,
            "unit": "PERCENT"
        },
        "stabilityStatus": "STABILIZING",
        "stabilityStatusAt": 1557129279.914
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateServicePrimaryTaskSet。

update-service

以下程式碼範例顯示如何使用 update-service。

AWS CLI

範例 1：變更服務中使用的任務定義

下列update-service範例會將my-http-service服務更新為使用 amazon-ecs-sample任務定義。

```
aws ecs update-service \
    --cluster test \
    --service my-http-service \
```

```
    --task-definition amazon-ecs-sample
```

輸出：

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/test/my-http-
service",
        "serviceName": "my-http-service",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/test",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 2,
        "pendingCount": 0,
        "launchType": "FARGATE",
        "platformVersion": "1.4.0",
        "platformFamily": "Linux",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
amazon-ecs-sample:2",
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": true,
                "rollback": true
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100,
            "alarms": {
                "alarmNames": [],
                "rollback": false,
                "enable": false
            }
        },
        "deployments": [
            {
                "id": "ecs-svc/7419115625193919142",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/amazon-ecs-sample:2",
                "desiredCount": 0,
                "pendingCount": 0,
                "runningCount": 0,
```

```
                "failedTasks": 0,
                "createdAt": "2025-02-21T13:26:02.734000-06:00",
                "updatedAt": "2025-02-21T13:26:02.734000-06:00",
                "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "IN_PROGRESS",
                "rolloutStateReason": "ECS deployment ecs-svc/7419115625193919142 in
 progress."
            },
            {
                "id": "ecs-svc/1709597507655421668",
                "status": "ACTIVE",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/old-amazon-ecs-sample:4",
                "desiredCount": 2,
                "pendingCount": 0,
                "runningCount": 2,
                "failedTasks": 0,
                "createdAt": "2025-01-24T11:13:07.621000-06:00",
                "updatedAt": "2025-02-02T16:11:30.838000-06:00",
                "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                             "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
```

```
                    }
                },
                "rolloutState": "COMPLETED",
                "rolloutStateReason": "ECS deployment ecs-svc/1709597507655421668
 completed."
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [
            {
                "id": "e40b4d1c-80d9-4834-aaf3-6a268e530e17",
                "createdAt": "2025-02-21T10:31:26.037000-06:00",
                "message": "(my-http-service) has reached a steady state."
            },
            {
                "id": "6ac069ad-fc8b-4e49-a35d-b5574a964c8e",
                "createdAt": "2025-02-21T04:31:22.703000-06:00",
                "message": "(my-http-service) has reached a steady state."
            },
            {
                "id": "265f7d37-dfd1-4880-a846-ec486f341919",
                "createdAt": "2025-02-20T22:31:22.514000-06:00",
                "message": "(my-http-service) has reached a steady state."
            }
        ],
        "createdAt": "2024-10-30T17:12:43.218000-05:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321",
                ],
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "ENABLED"
            }
        },
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
```

```
        },
        "createdBy": "arn:aws:iam::123456789012:role/AIDACKCEVSQ6C2EXAMPLE",
        "enableECSManagedTags": true,
        "propagateTags": "NONE",
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "DISABLED"
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的使用主控台更新 Amazon ECS 服務。

範例 2：變更服務中的任務數量

下列update-service範例會將服務所需的任務計數my-http-service從 更新為 2。

```
aws ecs update-service \
    --cluster MyCluster \
    --service my-http-service \
    --desired-count 2
```

輸出：

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/MyCluster/my-http-
service",
        "serviceName": "my-http-service",
        "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 1,
        "pendingCount": 0,
        "capacityProviderStrategy": [
            {
                "capacityProvider": "FARGATE",
                "weight": 1,
                "base": 0
            }
        ],
        "platformVersion": "LATEST",
        "platformFamily": "Linux",
```

```
        "taskDefinition": "arn:aws:ecs:us-east-1:123456789012:task-definition/
MyTaskDefinition",
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": true,
                "rollback": true
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100,
            "alarms": {
                "alarmNames": [],
                "rollback": false,
                "enable": false
            }
        },
        "deployments": [
            {
                "id": "ecs-svc/1976744184940610707",
                "status": "PRIMARY",
                "taskkDefinition": "arn:aws:ecs:us-east-1:123456789012:task-
definition/MyTaskDefinition",
                "desiredCount": 1,
                "pendingCount": 0,
                "runningCount": 1,
                "failedTasks": 0,
                "createdAt": "2024-12-03T16:24:25.225000-05:00",
                "updatedAt": "2024-12-03T16:25:15.837000-05:00",
                "capacityProviderStrategy": [
                    {
                        "capacityProvider": "FARGATE",
                        "weight": 1,
                        "base": 0
                    }
                ],
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-0d0eab1bb38d5ca64",
                            "subnet-0db5010045995c2d5"
                        ],
                        "securityGroups": [
                            "sg-02556bf85a191f59a"
```

```
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "COMPLETED",
                "rolloutStateReason": "ECS deployment ecs-svc/1976744184940610707
 completed."
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [
            {
                "id": "f27350b9-4b2a-4e2e-b72e-a4b68380de45",
                "createdAt": "2024-12-30T13:24:07.345000-05:00",
                "message": "(service my-http-service) has reached a steady state."
            },
            {
                "id": "e764ec63-f53f-45e3-9af2-d99f922d2957",
                "createdAt": "2024-12-30T12:32:21.600000-05:00",
                "message": "(service my-http-service) has reached a steady state."
            },
            {
                "id": "28444756-c2fa-47f8-bd60-93a8e05f3991",
                "createdAt": "2024-12-08T19:26:10.367000-05:00",
                "message": "(service my-http-service) has reached a steady state."
            }
        ],
        "createdAt": "2024-12-03T16:24:25.225000-05:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-0d0eab1bb38d5ca64",
                    "subnet-0db5010045995c2d5"
                ],
                "securityGroups": [
                    "sg-02556bf85a191f59a"
                ],
                "assignPublicIp": "ENABLED"
            }
        },
        "healthCheckGracePeriodSeconds": 0,
```

```
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
        },
        "createdBy": "arn:aws:iam::123456789012:role/Admin",
        "enableECSManagedTags": true,
        "propagateTags": "NONE",
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "ENABLED"
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的使用主控台更新 Amazon ECS 服務。

範例 3：設定 Amazon EBS 磁碟區以在服務更新時連接

下列update-service範例會將服務更新my-http-service為使用 Amazon EBS 磁碟區。您必須具有已連接 AmazonECSInfrastructureRolePolicyForVolumes受管政策的 Amazon ECS 基礎設施角色。您也必須指定與update-service請求中磁碟區名稱相同的任務定義，並將configuredAtLaunch設定為 true。此範例使用 --cli-input-json選項和名為 的 JSON 輸入檔案ebs.json。

```
aws ecs update-service \
    --cli-input-json file://ebs.json
```

ebs.json 的內容：

```
{
   "cluster": "mycluster",
   "taskDefinition": "mytaskdef",
   "service": "my-http-service",
   "desiredCount": 2,
   "volumeConfigurations": [
        {
            "name": "myEbsVolume",
            "managedEBSVolume": {
                "roleArn":"arn:aws:iam::123456789012:role/ecsInfrastructureRole",
                "volumeType": "gp3",
                "sizeInGiB": 100,
                "iops": 3000,
                "throughput": 125,
                "filesystemType": "ext4"
```

```
            }
        }
    ]
}
```

輸出：

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/mycluster/my-http-
service",
        "serviceName": "my-http-service",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/mycluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 2,
        "pendingCount": 0,
        "launchType": "FARGATE",
        "platformVersion": "LATEST",
        "platformFamily": "Linux",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
mytaskdef:1",
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": true,
                "rollback": true
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100,
            "alarms": {
                "alarmNames": [],
                "rollback": false,
                "enable": false
            }
        },
        "deployments": [
            {
                "id": "ecs-svc/2420458347226626275",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:1",
```

```
                "desiredCount": 0,
                "pendingCount": 0,
                "runningCount": 0,
                "failedTasks": 0,
                "createdAt": "2025-02-21T15:07:20.519000-06:00",
                "updatedAt": "2025-02-21T15:07:20.519000-06:00",
                "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321",
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "IN_PROGRESS",
                "rolloutStateReason": "ECS deployment ecs-svc/2420458347226626275 in
 progress.",
                "volumeConfigurations": [
                    {
                        "name": "ebs-volume",
                        "managedEBSVolume": {
                            "volumeType": "gp3",
                            "sizeInGiB": 100,
                            "iops": 3000,
                            "throughput": 125,
                            "roleArn": "arn:aws:iam::123456789012:role/
ecsInfrastructureRole",
                            "filesystemType": "ext4"
                        }
                    }
                ]
            },
            {
                "id": "ecs-svc/5191625155316533644",
                "status": "ACTIVE",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:2",
                "desiredCount": 2,
```

```
                "pendingCount": 0,
                "runningCount": 2,
                "failedTasks": 0,
                "createdAt": "2025-02-21T14:54:48.862000-06:00",
                "updatedAt": "2025-02-21T14:57:22.502000-06:00",
                "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "COMPLETED",
                "rolloutStateReason": "ECS deployment ecs-svc/5191625155316533644
 completed."
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [
            {
                "id": "b5823113-c2c5-458e-9649-8c2ed38f23a5",
                "createdAt": "2025-02-21T14:57:22.508000-06:00",
                "message": "(service my-http-service) has reached a steady state."
            },
            {
                "id": "b05a48e8-da35-4074-80aa-37ceb3167357",
                "createdAt": "2025-02-21T14:57:22.507000-06:00",
                "message": "(service my-http-service) (deployment ecs-
svc/5191625155316533644) deployment completed."
            },
            {
                "id": "a10cd55d-4ba6-4cea-a655-5a5d32ada8a0",
                "createdAt": "2025-02-21T14:55:32.833000-06:00",
                "message": "(service my-http-service) has started 1 tasks: (task
 fb9c8df512684aec92f3c57dc3f22361)."
            },
```

```
        ],
        "createdAt": "2025-02-21T14:54:48.862000-06:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321"
                ],
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "ENABLED"
            }
        },
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
        },
        "createdBy": "arn:aws:iam::123456789012:role/AIDACKCEVSQ6C2EXAMPLE",
        "enableECSManagedTags": true,
        "propagateTags": "NONE",
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "ENABLED"
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的搭配使用 Amazon EBS 磁碟區與 Amazon ECS。

範例 4：更新服務以不再使用 Amazon EBS 磁碟區

下列update-service範例會將服務更新my-http-service為不再使用 Amazon EBS 磁碟區。您必須指定任務定義修訂版，並將 configuredAtLaunch設為 false。

```
aws ecs update-service \
    --cluster mycluster \
    --task-definition mytaskdef \
    --service my-http-service \
    --desired-count 2 \
    --volume-configurations "[]"
```

輸出：


```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/mycluster/my-http-
service",
        "serviceName": "my-http-service",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/mycluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 2,
        "pendingCount": 0,
        "launchType": "FARGATE",
        "platformVersion": "LATEST",
        "platformFamily": "Linux",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
mytaskdef:3",
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": true,
                "rollback": true
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100,
            "alarms": {
                "alarmNames": [],
                "rollback": false,
                "enable": false
            }
        },
        "deployments": [
            {
                "id": "ecs-svc/7522791612543716777",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:3",
                "desiredCount": 0,
                "pendingCount": 0,
                "runningCount": 0,
                "failedTasks": 0,
                "createdAt": "2025-02-21T15:25:38.598000-06:00",
                "updatedAt": "2025-02-21T15:25:38.598000-06:00",
```

```
                    "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "IN_PROGRESS",
                "rolloutStateReason": "ECS deployment ecs-svc/7522791612543716777 in
 progress."
            },
            {
                "id": "ecs-svc/2420458347226626275",
                "status": "ACTIVE",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/myoldtaskdef:1",
                "desiredCount": 2,
                "pendingCount": 0,
                "runningCount": 2,
                "failedTasks": 0,
                "createdAt": "2025-02-21T15:07:20.519000-06:00",
                "updatedAt": "2025-02-21T15:10:59.955000-06:00",
                "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "COMPLETED",
```

```
                "rolloutStateReason": "ECS deployment ecs-svc/2420458347226626275
 completed.",
                "volumeConfigurations": [
                    {
                        "name": "ebs-volume",
                        "managedEBSVolume": {
                            "volumeType": "gp3",
                            "sizeInGiB": 100,
                            "iops": 3000,
                            "throughput": 125,
                            "roleArn": "arn:aws:iam::123456789012:role/
ecsInfrastructureRole",
                            "filesystemType": "ext4"
                        }
                    }
                ]
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [
            {
                "id": "4f2c3ca1-7800-4048-ba57-bba210ada2ad",
                "createdAt": "2025-02-21T15:10:59.959000-06:00",
                "message": "(service my-http-service) has reached a steady state."
            },
            {
                "id": "4b36a593-2d40-4ed6-8be8-b9b699eb6198",
                "createdAt": "2025-02-21T15:10:59.958000-06:00",
                "message": "(service my-http-service) (deployment ecs-
svc/2420458347226626275) deployment completed."
            },
            {
                "id": "88380089-14e2-4ef0-8dbb-a33991683371",
                "createdAt": "2025-02-21T15:09:39.055000-06:00",
                "message": "(service my-http-service) has stopped 1 running tasks:
 (task fb9c8df512684aec92f3c57dc3f22361)."
            },
            {
                "id": "97d84243-d52f-4255-89bb-9311391c61f6",
                "createdAt": "2025-02-21T15:08:57.653000-06:00",
                "message": "(service my-http-service) has stopped 1 running tasks:
 (task 33eff090ad2c40539daa837e6503a9bc)."
            },
```

```
            {
                "id": "672ece6c-e2d0-4021-b5da-eefb14001687",
                "createdAt": "2025-02-21T15:08:15.631000-06:00",
                "message": "(service my-http-service) has started 1 tasks: (task
 996c02a66ff24f3190a4a8e0c841740f)."
            },
            {
                "id": "a3cf9bea-9be6-4175-ac28-4c68360986eb",
                "createdAt": "2025-02-21T15:07:36.931000-06:00",
                "message": "(service my-http-service) has started 1 tasks: (task
 d5d23c39f89e46cf9a647b9cc6572feb)."
            },
            {
                "id": "b5823113-c2c5-458e-9649-8c2ed38f23a5",
                "createdAt": "2025-02-21T14:57:22.508000-06:00",
                "message": "(service my-http-service) has reached a steady state."
            },
            {
                "id": "b05a48e8-da35-4074-80aa-37ceb3167357",
                "createdAt": "2025-02-21T14:57:22.507000-06:00",
                "message": "(service my-http-service) (deployment ecs-
svc/5191625155316533644) deployment completed."
            },
            {
                "id": "a10cd55d-4ba6-4cea-a655-5a5d32ada8a0",
                "createdAt": "2025-02-21T14:55:32.833000-06:00",
                "message": "(service my-http-service) has started 1 tasks: (task
 fb9c8df512684aec92f3c57dc3f22361)."
            },
            {
                "id": "42da91fa-e26d-42ef-88c3-bb5965c56b2f",
                "createdAt": "2025-02-21T14:55:02.703000-06:00",
                "message": "(service my-http-service) has started 1 tasks: (task
 33eff090ad2c40539daa837e6503a9bc)."
            }
        ],
        "createdAt": "2025-02-21T14:54:48.862000-06:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321"
                ],
```

```
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "ENABLED"
            }
        },
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
        },
        "createdBy": "arn:aws:iam::123456789012:role/AIDACKCEVSQ6C2EXAMPLE",
        "enableECSManagedTags": true,
        "propagateTags": "NONE",
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "ENABLED"
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的搭配使用 Amazon EBS 磁碟區與 Amazon ECS。

範例 5：開啟服務的可用區域重新平衡

下列update-service範例會開啟服務的可用區域重新平衡my-http-service。

```
aws ecs update-service \
    --cluster MyCluster \
    --service my-http-service \
    --availability-zone-rebalancing ENABLED
```

輸出：

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/MyCluster/my-http-
service",
        "serviceName": "my-http-service",
        "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
```

```
        "runningCount": 1,
        "pendingCount": 0,
        "capacityProviderStrategy": [
            {
                "capacityProvider": "FARGATE",
                "weight": 1,
                "base": 0
            }
        ],
        "platformVersion": "LATEST",
        "platformFamily": "Linux",
        "taskDefinition": "arn:aws:ecs:us-east-1:123456789012:task-definition/
MyTaskDefinition",
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": true,
                "rollback": true
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100,
            "alarms": {
                "alarmNames": [],
                "rollback": false,
                "enable": false
            }
        },
        "deployments": [
            {
                "id": "ecs-svc/1976744184940610707",
                "status": "PRIMARY",
                "taskkDefinition": "arn:aws:ecs:us-east-1:123456789012:task-
definition/MyTaskDefinition",
                "desiredCount": 1,
                "pendingCount": 0,
                "runningCount": 1,
                "failedTasks": 0,
                "createdAt": "2024-12-03T16:24:25.225000-05:00",
                "updatedAt": "2024-12-03T16:25:15.837000-05:00",
                "capacityProviderStrategy": [
                    {
                        "capacityProvider": "FARGATE",
                        "weight": 1,
                        "base": 0
                    }
```

```
                ],
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-0d0eab1bb38d5ca64",
                            "subnet-0db5010045995c2d5"
                        ],
                        "securityGroups": [
                            "sg-02556bf85a191f59a"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "COMPLETED",
                "rolloutStateReason": "ECS deployment ecs-svc/1976744184940610707
 completed."
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [],
        "createdAt": "2024-12-03T16:24:25.225000-05:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-0d0eab1bb38d5ca64",
                    "subnet-0db5010045995c2d5"
                ],
                "securityGroups": [
                    "sg-02556bf85a191f59a"
                ],
                "assignPublicIp": "ENABLED"
            }
        },
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
        },
        "createdBy": "arn:aws:iam::123456789012:role/Admin",
```

```
        "enableECSManagedTags": true,
        "propagateTags": "NONE",
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "ENABLED"
    }
}
```

如需詳細資訊，請參閱《Amazon ECS 開發人員指南》中的使用主控台更新 Amazon ECS 服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateService。

update-task-protection

以下程式碼範例顯示如何使用 update-task-protection。

AWS CLI

範例 1：啟用 ECS 任務的任務保護

以下內容可在從部署或服務 AutoScaling 縮減期間update-task-protection保護您的 ECS 任務免於終止。您可以指定任務保護的自訂過期期間，從 1 到 2，880 分鐘 (48 小時）。如果您未指定過期期間，則啟用任務保護預設時間為 2 小時。

```
aws ecs update-task-protection \
    --cluster ECS-project-update-cluster \
    --tasks c43ed3b1331041f289316f958adb6a24 \
    --protection-enabled \
    --expires-in-minutes 300
```

輸出：

```
{
"protectedTasks": [
    {
        "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
c43ed3b1331041f289316f958adb6a24",
        "protectionEnabled": true,
        "expirationDate": "2024-09-14T19:53:36.687000-05:00"
    }
],
"failures": []
```

```
}
```

範例 2：停用 ECS 任務的任務保護

下列會update-task-protection停用從部署或服務 AutoScaling 縮減保護的任務。

```
aws ecs update-task-protection \
    --cluster ECS-project-update-cluster \
    --tasks c43ed3b1331041f289316f958adb6a24 \
    --no-protection-enabled
```

輸出：

```
{
    "protectedTasks": [
        {
            "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
c43ed3b1331041f289316f958adb6a24",
            "protectionEnabled": false
        }
    ],
    "failures": []
}
```

如需任務保護的更多形式，請參閱《Amazon ECS 開發人員指南》中的保護您的 Amazon ECS 任務不受縮減事件終止。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateTaskProtection。

update-task-set

以下程式碼範例顯示如何使用 update-task-set。

AWS CLI

更新任務集

下列update-task-set範例會更新任務集以調整規模。

```
aws ecs update-task-set \
```

```
    --cluster MyCluster \
    --service MyService \
    --task-set arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/MyService/ecs-
svc/1234567890123456789 \
    --scale value=50,unit=PERCENT
```

輸出：

```
{
    "taskSet": {
        "id": "ecs-svc/1234567890123456789",
        "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/
MyService/ecs-svc/1234567890123456789",
        "status": "ACTIVE",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
sample-fargate:2",
        "computedDesiredCount": 0,
        "pendingCount": 0,
        "runningCount": 0,
        "createdAt": 1557128360.711,
        "updatedAt": 1557129279.914,
        "launchType": "EC2",
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321"
                ],
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "DISABLED"
            }
        },
        "loadBalancers": [],
        "serviceRegistries": [],
        "scale": {
            "value": 50.0,
            "unit": "PERCENT"
        },
        "stabilityStatus": "STABILIZING",
        "stabilityStatusAt": 1557129279.914
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateTaskSet。

使用 的 Amazon EFS 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon EFS 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-file-system`

以下程式碼範例顯示如何使用 `create-file-system`。

AWS CLI

建立加密的檔案系統

下列`create-file-system`範例會使用預設 CMK 建立加密的檔案系統。它也會新增標籤 `Name=my-file-system`。

```
aws efs create-file-system \
    --performance-mode generalPurpose \
    --throughput-mode bursting \
    --encrypted \
    --tags Key=Name,Value=my-file-system
```

輸出：

```
{
    "OwnerId": "123456789012",
    "CreationToken": "console-d7f56c5f-e433-41ca-8307-9d9c0example",
    "FileSystemId": "fs-c7a0456e",
```

```
    "FileSystemArn": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-system/
fs-48499b4d",
    "CreationTime": 1595286880.0,
    "LifeCycleState": "creating",
    "Name": "my-file-system",
    "NumberOfMountTargets": 0,
    "SizeInBytes": {
        "Value": 0,
        "ValueInIA": 0,
        "ValueInStandard": 0
    },
    "PerformanceMode": "generalPurpose",
    "Encrypted": true,
    "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/a59b3472-e62c-42e4-
adcf-30d92example",
    "ThroughputMode": "bursting",
    "Tags": [
        {
            "Key": "Name",
            "Value": "my-file-system"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的建立 Amazon EFS 檔案系統。 Amazon Elastic File System

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFileSystem。

create-mount-target

以下程式碼範例顯示如何使用 create-mount-target。

AWS CLI

建立掛載目標

下列create-mount-target範例會為指定的檔案系統建立掛載目標。

```
aws efs create-mount-target \
    --file-system-id fs-c7a0456e \
    --subnet-id subnet-02bf4c428bexample \
    --security-groups sg-068f739363example
```

輸出：

```
{
    "OwnerId": "123456789012",
    "MountTargetId": "fsmt-f9a14450",
    "FileSystemId": "fs-c7a0456e",
    "SubnetId": "subnet-02bf4c428bexample",
    "LifeCycleState": "creating",
    "IpAddress": "10.0.1.24",
    "NetworkInterfaceId": "eni-02d542216aexample",
    "AvailabilityZoneId": "use2-az2",
    "AvailabilityZoneName": "us-east-2b",
    "VpcId": "vpc-0123456789abcdef0"
}
```

如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的建立掛載目標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateMountTarget。

delete-file-system

以下程式碼範例顯示如何使用 delete-file-system。

AWS CLI

刪除檔案系統

下列delete-file-system範例會刪除指定的檔案系統。

```
aws efs delete-file-system \
    --file-system-id fs-c7a0456e
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的刪除 Amazon EFS 檔案系統。 Amazon Elastic File System

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFileSystem。

delete-mount-target

以下程式碼範例顯示如何使用 delete-mount-target。

AWS CLI

刪除掛載目標

下列delete-mount-target範例會刪除指定的掛載目標。

```
aws efs delete-mount-target \
    --mount-target-id fsmt-f9a14450
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的建立掛載目標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteMountTarget。

describe-file-systems

以下程式碼範例顯示如何使用 describe-file-systems。

AWS CLI

描述檔案系統

下列describe-file-systems範例說明指定的檔案系統。

```
aws efs describe-file-systems \
    --file-system-id fs-c7a0456e
```

輸出：

```
{
    "FileSystems": [
        {
            "OwnerId": "123456789012",
            "CreationToken": "console-d7f56c5f-e433-41ca-8307-9d9c0example",
            "FileSystemId": "fs-c7a0456e",
            "FileSystemArn": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-
system/fs-48499b4d",
            "CreationTime": 1595286880.0,
            "LifeCycleState": "available",
            "Name": "my-file-system",
            "NumberOfMountTargets": 3,
```

```
            "SizeInBytes": {
                "Value": 6144,
                "Timestamp": 1600991437.0,
                "ValueInIA": 0,
                "ValueInStandard": 6144
            },
            "PerformanceMode": "generalPurpose",
            "Encrypted": true,
            "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/a59b3472-e62c-42e4-
adcf-30d92example",
            "ThroughputMode": "bursting",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "my-file-system"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的管理 Amazon EFS 檔案系統。 Amazon Elastic File System

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFileSystems。

describe-mount-targets

以下程式碼範例顯示如何使用 describe-mount-targets。

AWS CLI

描述掛載目標

下列describe-mount-targets範例說明指定的掛載目標。

```
aws efs describe-mount-targets \
    --mount-target-id fsmt-f9a14450
```

輸出：

```
{
```

```
    "MountTargets": [
        {
            "OwnerId": "123456789012",
            "MountTargetId": "fsmt-f9a14450",
            "FileSystemId": "fs-c7a0456e",
            "SubnetId": "subnet-02bf4c428bexample",
            "LifeCycleState": "creating",
            "IpAddress": "10.0.1.24",
            "NetworkInterfaceId": "eni-02d542216aexample",
            "AvailabilityZoneId": "use2-az2",
            "AvailabilityZoneName": "us-east-2b",
            "VpcId": "vpc-0123456789abcdef0"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的建立掛載目標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeMountTargets。

describe-tags

以下程式碼範例顯示如何使用 describe-tags。

AWS CLI

描述檔案系統的標籤

下列describe-tags範例說明指定檔案系統的標籤。

```
aws efs describe-tags \
    --file-system-id fs-c7a0456e
```

輸出：

```
{
    "Tags": [
        {
            "Key": "Name",
            "Value": "my-file-system"
        },
        {
            "Key": "Department",
```

```
            "Value": "Business Intelligence"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Elastic File System File System 使用者指南》中的管理檔案系統標籤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTags。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

擷取資源的標籤

下列`list-tags-for-resource`範例會擷取與指定檔案系統相關聯的標籤。

```
aws efs list-tags-for-resource \
    --resource-id fs-c7a0456e
```

輸出：

```
{
    "Tags": [
        {
            "Key": "Name",
            "Value": "my-file-system"
        },
        {
            "Key": "Department",
            "Value": "Business Intelligence"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Elastic File System File System 使用者指南》中的管理檔案系統標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

標記資源

下列tag-resource範例會將 標籤新增至Department=Business Intelligence指定的檔案系統。

```
aws efs tag-resource \
    --resource-id fs-c7a0456e \
    --tags Key=Department,Value="Business Intelligence"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Elastic File System File System 使用者指南》中的管理檔案系統標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源移除標籤

下列untag-resource範例會從指定的檔案系統移除具有標籤索引鍵的Department標籤。

```
aws efs untag-resource \
    --resource-id fs-c7a0456e \
    --tag-keys Department
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Elastic File System File System 使用者指南》中的管理檔案系統標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

使用 的 Amazon EKS 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon EKS 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-access-policy`

以下程式碼範例顯示如何使用 `associate-access-policy`。

AWS CLI

將存取政策及其範圍與叢集的存取項目建立關聯

下列會將存取政策及其範圍與指定叢集的存取項目`associate-access-policy`建立關聯。

```
aws eks associate-access-policy \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/Admin \
    --policy-arn arn:aws:eks::aws:cluster-access-policy/AmazonEKSEditPolicy \
    --access-scope type=namespace,namespaces=default
```

輸出：

```
{
    "clusterName": "eks-customer",
    "principalArn": "arn:aws:iam::111122223333:role/Admin",
    "associatedAccessPolicy": {
        "policyArn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSEditPolicy",
        "accessScope": {
            "type": "namespace",
```

```
            "namespaces": [
                "default"
            ]
        },
        "associatedAt": "2025-05-24T15:59:51.981000-05:00",
        "modifiedAt": "2025-05-24T15:59:51.981000-05:00"
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的將存取政策與存取項目建立關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateAccessPolicy。

associate-encryption-config

以下程式碼範例顯示如何使用 associate-encryption-config。

AWS CLI

將加密組態與現有叢集建立關聯

下列associate-encryption-config範例會在尚未啟用加密的現有 EKS 叢集上啟用 加密。

```
aws eks associate-encryption-config \
    --cluster-name my-eks-cluster \
    --encryption-config '[{"resources":["secrets"],"provider":
{"keyArn":"arn:aws:kms:region-code:account:key/key"}}]'
```

輸出：

```
{
    "update": {
        "id": "3141b835-8103-423a-8e68-12c2521ffa4d",
        "status": "InProgress",
        "type": "AssociateEncryptionConfig",
        "params": [
            {
                "type": "EncryptionConfig",
                "value": "[{\"resources\":[\"secrets\"],\"provider\":{\"keyArn\":
\"arn:aws:kms:region-code:account:key/key\"}}]"
            }
        ],
```

```
        "createdAt": "2024-03-14T11:01:26.297000-04:00",
        "errors": []
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的在現有叢集上啟用秘密加密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateEncryptionConfig。

associate-identity-provider-config

以下程式碼範例顯示如何使用 associate-identity-provider-config。

AWS CLI

將身分提供者與您的 Amazon EKS 叢集建立關聯

下列associate-identity-provider-config範例會將身分提供者與您的 Amazon EKS 叢集建立關聯。

```
aws eks associate-identity-provider-config \
    --cluster-name my-eks-cluster \
    --oidc 'identityProviderConfigName=my-identity-provider,issuerUrl=https://
oidc.eks.us-east-2.amazonaws.com/
id/38D6A4619A0A69E342B113ED7F1A7652,clientId=kubernetes,usernameClaim=email,usernamePrefix=m
username-prefix,groupsClaim=my-claim,groupsPrefix=my-groups-
prefix,requiredClaims={Claim1=value1,Claim2=value2}' \
    --tags env=dev
```

輸出：

```
{
    "update": {
        "id": "8c6c1bef-61fe-42ac-a242-89412387b8e7",
        "status": "InProgress",
        "type": "AssociateIdentityProviderConfig",
        "params": [
            {
                "type": "IdentityProviderConfig",
                "value": "[{\"type\":\"oidc\",\"name\":\"my-identity-provider\"}]"
            }
        ],
```

```
        "createdAt": "2024-04-11T13:46:49.648000-04:00",
        "errors": []
    },
    "tags": {
        "env": "dev"
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的從 OpenID Connect 身分提供者驗證叢集的使用者 - 關聯 OIDC 身分提供者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateIdentityProviderConfig。

create-access-entry

以下程式碼範例顯示如何使用 create-access-entry。

AWS CLI

範例 1：建立 EKS 叢集的存取項目

下列create-access-entry範例會建立存取項目，允許 IAM 主體存取 EKS 叢集。

```
aws eks create-access-entry \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:user/eks-user
```

輸出：

```
{
    "accessEntry": {
        "clusterName": "eks-customer",
        "principalArn": "arn:aws:iam::111122223333:user/eks-user",
        "kubernetesGroups": [],
        "accessEntryArn": "arn:aws:eks:us-west-2:111122223333:access-entry/eks-
customer/user/111122223333/eks-user/a1b2c3d4-5678-90ab-cdef-a6506e3d36p0",
        "createdAt": "2025-04-14T22:45:48.097000-05:00",
        "modifiedAt": "2025-04-14T22:45:48.097000-05:00",
        "tags": {},
        "username": "arn:aws:iam::111122223333:user/eks-user",
        "type": "STANDARD"
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的建立存取項目。

範例 2：指定存取項目的類型，以建立 EKS 叢集的存取項目

下列create-access-entry範例會在 EKS 叢集EC2_LINUX中建立 類型的存取項目。根據預設，會建立類型STANDARD存取項目。除了預設值之外，如果我們指定任何其他存取項目類型，則需要在 CLI 中傳遞 IAM 角色 ARN。

```
aws eks create-access-entry \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/admin-test-ip \
    --type EC2_LINUX
```

輸出：

```
{
    "accessEntry": {
        "clusterName": "eks-customer",
        "principalArn": "arn:aws:iam::111122223333:role/admin-test-ip",
        "kubernetesGroups": [
            "system:nodes"
        ],
        "accessEntryArn": "arn:aws:eks:us-west-2:111122223333:access-entry/eks-
customer/role/111122223333/admin-test-ip/accb5418-f493-f390-3e6e-c3f19f725fcp",
        "createdAt": "2025-05-06T19:42:45.453000-05:00",
        "modifiedAt": "2025-05-06T19:42:45.453000-05:00",
        "tags": {},
        "username": "system:node:{{EC2PrivateDNSName}}",
        "type": "EC2_LINUX"
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的建立存取項目。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAccessEntry。

create-addon

以下程式碼範例顯示如何使用 create-addon。

AWS CLI

範例 1：為個別 EKS 叢集版本建立具有預設相容版本的 Amazon EKS 附加元件

下列create-addon範例命令會為個別 EKS 叢集版本建立具有預設相容版本的 Amazon EKS 附加元件。

```
aws eks create-addon \
    --cluster-name my-eks-cluster \
    --addon-name my-eks-addon \
    --service-account-role-arn arn:aws:iam::111122223333:role/role-name
```

輸出：

```
{
    "addon": {
        "addonName": "my-eks-addon",
        "clusterName": "my-eks-cluster",
        "status": "CREATING",
        "addonVersion": "v1.15.1-eksbuild.1",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-
addon/1ec71ee1-b9c2-8915-4e17-e8be0a55a149",
        "createdAt": "2024-03-14T12:20:03.264000-04:00",
        "modifiedAt": "2024-03-14T12:20:03.283000-04:00",
        "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的管理 Amazon EKS 附加元件 - 建立附加元件。

範例 2：使用特定附加元件版本建立 Amazon EKS 附加元件

下列create-addon範例命令會建立具有特定附加元件版本的 Amazon EKS 附加元件。

```
aws eks create-addon \
    --cluster-name my-eks-cluster \
    --addon-name my-eks-addon \
```

```
    --service-account-role-arn arn:aws:iam::111122223333:role/role-name \
    --addon-version v1.16.4-eksbuild.2
```

輸出：

```
{
    "addon": {
        "addonName": "my-eks-addon",
        "clusterName": "my-eks-cluster",
        "status": "CREATING",
        "addonVersion": "v1.16.4-eksbuild.2",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-
addon/34c71ee6-7738-6c8b-c6bd-3921a176b5ff",
        "createdAt": "2024-03-14T12:30:24.507000-04:00",
        "modifiedAt": "2024-03-14T12:30:24.521000-04:00",
        "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的管理 Amazon EKS 附加元件 - 建立附加元件。

範例 3：使用自訂組態值建立 Amazon EKS 附加元件並解決衝突詳細資訊

下列create-addon範例命令會建立具有自訂組態值的 Amazon EKS 附加元件，並解決衝突詳細資訊。

```
aws eks create-addon \
    --cluster-name my-eks-cluster \
    --addon-name my-eks-addon \
    --service-account-role-arn arn:aws:iam::111122223333:role/role-name \
    --addon-version v1.16.4-eksbuild.2 \
    --configuration-values '{"resources":{"limits":{"cpu":"100m"}}}' \
    --resolve-conflicts OVERWRITE
```

輸出：

```
{
```

```
    "addon": {
        "addonName": "my-eks-addon",
        "clusterName": "my-eks-cluster",
        "status": "CREATING",
        "addonVersion": "v1.16.4-eksbuild.2",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-
addon/a6c71ee9-0304-9237-1be8-25af1b0f1ffb",
        "createdAt": "2024-03-14T12:35:58.313000-04:00",
        "modifiedAt": "2024-03-14T12:35:58.327000-04:00",
        "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "tags": {},
        "configurationValues": "{\"resources\":{\"limits\":{\"cpu\":\"100m\"}}}"
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的管理 Amazon EKS 附加元件 - 建立附加元件。

範例 4：使用自訂 JSON 組態值檔案建立 Amazon EKS 附加元件

下列create-addon範例命令會使用自訂組態值建立 Amazon EKS 附加元件，並解決衝突詳細資訊。

```
aws eks create-addon \
    --cluster-name my-eks-cluster \
    --addon-name my-eks-addon \
    --service-account-role-arn arn:aws:iam::111122223333:role/role-name \
    --addon-version v1.16.4-eksbuild.2 \
    --configuration-values 'file://configuration-values.json' \
    --resolve-conflicts OVERWRITE \
    --tags '{"eks-addon-key-1": "value-1" , "eks-addon-key-2": "value-2"}'
```

configuration-values.json 的內容：

```
{
    "resources": {
        "limits": {
            "cpu": "150m"
        }
```

```
    },
    "env": {
        "AWS_VPC_K8S_CNI_LOGLEVEL": "ERROR"
    }
}
```

輸出：

```
{
    "addon": {
        "addonName": "my-eks-addon",
        "clusterName": "my-eks-cluster",
        "status": "CREATING",
        "addonVersion": "v1.16.4-eksbuild.2",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-
addon/d8c71ef8-fbd8-07d0-fb32-6a7be19ececd",
        "createdAt": "2024-03-14T13:10:51.763000-04:00",
        "modifiedAt": "2024-03-14T13:10:51.777000-04:00",
        "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "tags": {
            "eks-addon-key-1": "value-1",
            "eks-addon-key-2": "value-2"
        },
        "configurationValues": "{\n    \"resources\": {\n        \"limits\":
 {\n            \"cpu\": \"150m\"\n        }\n    },\n    \"env\": {\n
 \"AWS_VPC_K8S_CNI_LOGLEVEL\": \"ERROR\"\n    }\n}"
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的管理 Amazon EKS 附加元件 - 建立附加元件。

範例 5：使用自訂 YAML 組態值檔案建立 Amazon EKS 附加元件

下列create-addon範例命令會使用自訂組態值建立 Amazon EKS 附加元件，並解決衝突詳細資訊。

```
aws eks create-addon \
    --cluster-name my-eks-cluster \
```

```
    --addon-name my-eks-addon \
    --service-account-role-arn arn:aws:iam::111122223333:role/role-name \
    --addon-version v1.16.4-eksbuild.2 \
    --configuration-values 'file://configuration-values.yaml' \
    --resolve-conflicts OVERWRITE \
    --tags '{"eks-addon-key-1": "value-1" , "eks-addon-key-2": "value-2"}'
```

configuration-values.yaml 的內容：

```
resources:
    limits:
        cpu: '100m'
env:
    AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'
```

輸出：

```
{
    "addon": {
        "addonName": "my-eks-addon",
        "clusterName": "my-eks-cluster",
        "status": "CREATING",
        "addonVersion": "v1.16.4-eksbuild.2",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-
addon/d4c71efb-3909-6f36-a548-402cd4b5d59e",
        "createdAt": "2024-03-14T13:15:45.220000-04:00",
        "modifiedAt": "2024-03-14T13:15:45.237000-04:00",
        "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "tags": {
            "eks-addon-key-3": "value-3",
            "eks-addon-key-4": "value-4"
        },
        "configurationValues": "resources:\n    limits:\n        cpu: '100m'\nenv:\n
    AWS_VPC_K8S_CNI_LOGLEVEL: 'INFO'"
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的管理 Amazon EKS 附加元件 - 建立附加元件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAddon。

create-cluster

以下程式碼範例顯示如何使用 create-cluster。

AWS CLI

建立新的叢集

此範例命令會在您的預設區域中建立名為 prod的叢集。

命令：

```
aws eks create-cluster --name prod \
--role-arn arn:aws:iam::012345678910:role/eks-service-role-
AWSServiceRoleForAmazonEKS-J7ONKE3BQ4PI \
--resources-vpc-config subnetIds=subnet-6782e71e,subnet-
e7e761ac,securityGroupIds=sg-6979fe18
```

輸出：

```
{
    "cluster": {
        "name": "prod",
        "arn": "arn:aws:eks:us-west-2:012345678910:cluster/prod",
        "createdAt": 1527808069.147,
        "version": "1.10",
        "roleArn": "arn:aws:iam::012345678910:role/eks-service-role-
AWSServiceRoleForAmazonEKS-J7ONKE3BQ4PI",
        "resourcesVpcConfig": {
            "subnetIds": [
                "subnet-6782e71e",
                "subnet-e7e761ac"
            ],
            "securityGroupIds": [
                "sg-6979fe18"
            ],
            "vpcId": "vpc-950809ec"
        },
        "status": "CREATING",
        "certificateAuthority": {}
```

```
    }
}
```

建立啟用私有端點存取和記錄的新叢集

此範例命令會在預設example區域中建立名為 的叢集，並停用公有端點存取、啟用私有端點存取，以及啟用所有記錄類型。

命令：

```
aws eks create-cluster --name example --kubernetes-version 1.12 \
--role-arn arn:aws:iam::012345678910:role/example-cluster-ServiceRole-1XWBQWYSFRE2Q
 \
--resources-vpc-
config subnetIds=subnet-0a188dccd2f9a632f,subnet-09290d93da4278664,subnet-0f21dd86e0e91134a,
 \
--logging '{"clusterLogging":[{"types":
["api","audit","authenticator","controllerManager","scheduler"],"enabled":true}]}'
```

輸出：

```
{
    "cluster": {
        "name": "example",
        "arn": "arn:aws:eks:us-west-2:012345678910:cluster/example",
        "createdAt": 1565804921.901,
        "version": "1.12",
        "roleArn": "arn:aws:iam::012345678910:role/example-cluster-
ServiceRole-1XWBQWYSFRE2Q",
        "resourcesVpcConfig": {
            "subnetIds": [
                "subnet-0a188dccd2f9a632f",
                "subnet-09290d93da4278664",
                "subnet-0f21dd86e0e91134a",
                "subnet-0173dead68481a583",
                "subnet-051f70a57ed6fcab6",
                "subnet-01322339c5c7de9b4"
            ],
            "securityGroupIds": [
                "sg-0c5b580845a031c10"
            ],
            "vpcId": "vpc-0f622c01f68d4afec",
```

```
            "endpointPublicAccess": false,
            "endpointPrivateAccess": true
        },
        "logging": {
            "clusterLogging": [
                {
                    "types": [
                        "api",
                        "audit",
                        "authenticator",
                        "controllerManager",
                        "scheduler"
                    ],
                    "enabled": true
                }
            ]
        },
        "status": "CREATING",
        "certificateAuthority": {},
        "platformVersion": "eks.3"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCluster。

create-fargate-profile

以下程式碼範例顯示如何使用 create-fargate-profile。

AWS CLI

範例 1：為具有命名空間的選擇器建立 EKS Fargate 設定檔

下列create-fargate-profile範例會為具有命名空間的選取器建立 EKS Fargate 設定檔。

```
aws eks create-fargate-profile \
    --cluster-name my-eks-cluster \
    --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \
    --fargate-profile-name my-fargate-profile \
    --selectors '[{"namespace": "default"}]'
```

輸出：

```
{
    "fargateProfile": {
        "fargateProfileName": "my-fargate-profile",
        "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-
eks-cluster/my-fargate-profile/a2c72bca-318e-abe8-8ed1-27c6d4892e9e",
        "clusterName": "my-eks-cluster",
        "createdAt": "2024-03-19T12:38:47.368000-04:00",
        "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "subnets": [
            "subnet-09d912bb63ef21b9a",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-0e2907431c9988b72"
        ],
        "selectors": [
            {
                "namespace": "default"
            }
        ],
        "status": "CREATING",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的 AWS Fargate 設定檔 - 建立 Fargate 設定檔。

範例 2：為具有命名空間和標籤的選擇器建立 EKS Fargate 設定檔

下列create-fargate-profile範例會為具有命名空間和標籤的選擇器建立 EKS Fargate 設定檔。

```
aws eks create-fargate-profile \
    --cluster-name my-eks-cluster \
    --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \
    --fargate-profile-name my-fargate-profile \
    --selectors '[{"namespace": "default", "labels": {"labelname1":
 "labelvalue1"}}]'
```

輸出：

```
{
    "fargateProfile": {
```

```
        "fargateProfileName": "my-fargate-profile",
        "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-
eks-cluster/my-fargate-profile/88c72bc7-e8a4-fa34-44e4-2f1397224bb3",
        "clusterName": "my-eks-cluster",
        "createdAt": "2024-03-19T12:33:48.125000-04:00",
        "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "subnets": [
            "subnet-09d912bb63ef21b9a",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-0e2907431c9988b72"
        ],
        "selectors": [
            {
                "namespace": "default",
                "labels": {
                    "labelname1": "labelvalue1"
                }
            }
        ],
        "status": "CREATING",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的 AWS Fargate 設定檔 - 建立 Fargate 設定檔。

範例 3：為具有命名空間和標籤的選擇器建立 EKS Fargate 設定檔，以及要啟動 Pod IDs。

下列create-fargate-profile範例會為具有命名空間和標籤的選擇器建立 EKS Fargate 設定檔，以及要啟動 Pod IDs。

```
aws eks create-fargate-profile \
    --cluster-name my-eks-cluster \
    --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \
    --fargate-profile-name my-fargate-profile \
    --selectors '[{"namespace": "default", "labels": {"labelname1":
 "labelvalue1"}}]' \
    --subnets '["subnet-09d912bb63ef21b9a", "subnet-04ad87f71c6e5ab4d",
 "subnet-0e2907431c9988b72"]'
```

輸出：

```
{
    "fargateProfile": {
        "fargateProfileName": "my-fargate-profile",
        "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-
eks-cluster/my-fargate-profile/e8c72bc8-e87b-5eb6-57cb-ed4fe57577e3",
        "clusterName": "my-eks-cluster",
        "createdAt": "2024-03-19T12:35:58.640000-04:00",
        "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "subnets": [
            "subnet-09d912bb63ef21b9a",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-0e2907431c9988b72"
        ],
        "selectors": [
            {
                "namespace": "default",
                "labels": {
                    "labelname1": "labelvalue1"
                }
            }
        ],
        "status": "CREATING",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的 AWS Fargate 描述檔 - 建立 Fargate 描述檔。

範例 4：為具有多個命名空間和標籤的選擇器建立 EKS Fargate 設定檔，以及要在其中啟動 Pod IDs

下列create-fargate-profile範例會為具有多個命名空間和標籤的選擇器建立 EKS Fargate 設定檔，以及要在其中啟動 Pod IDs。

```
aws eks create-fargate-profile \
    --cluster-name my-eks-cluster \
    --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \
    --fargate-profile-name my-fargate-profile \
    --selectors '[{"namespace": "default1", "labels": {"labelname1": "labelvalue1",
 "labelname2": "labelvalue2"}}, {"namespace": "default2", "labels": {"labelname1":
 "labelvalue1", "labelname2": "labelvalue2"}}]' \
```

```
    --subnets '["subnet-09d912bb63ef21b9a", "subnet-04ad87f71c6e5ab4d",
 "subnet-0e2907431c9988b72"]' \
    --tags '{"eks-fargate-profile-key-1": "value-1" , "eks-fargate-profile-key-2":
 "value-2"}'
```

輸出：

```
{
    "fargateProfile": {
        "fargateProfileName": "my-fargate-profile",
        "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-
eks-cluster/my-fargate-profile/4cc72bbf-b766-8ee6-8d29-e62748feb3cd",
        "clusterName": "my-eks-cluster",
        "createdAt": "2024-03-19T12:15:55.271000-04:00",
        "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "subnets": [
            "subnet-09d912bb63ef21b9a",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-0e2907431c9988b72"
        ],
        "selectors": [
            {
                "namespace": "default1",
                "labels": {
                    "labelname2": "labelvalue2",
                    "labelname1": "labelvalue1"
                }
            },
            {
                "namespace": "default2",
                "labels": {
                    "labelname2": "labelvalue2",
                    "labelname1": "labelvalue1"
                }
            }
        ],
        "status": "CREATING",
        "tags": {
            "eks-fargate-profile-key-2": "value-2",
            "eks-fargate-profile-key-1": "value-1"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的 AWS Fargate 設定檔 - 建立 Fargate 設定檔。

範例 5：使用命名空間和標籤的萬用字元選取器建立 EKS Fargate 設定檔，以及要在其中啟動 Pod IDs

下列create-fargate-profile範例會為具有多個命名空間和標籤的選擇器建立 EKS Fargate 設定檔，以及要在其中啟動 Pod IDs。

```
aws eks create-fargate-profile \
    --cluster-name my-eks-cluster \
    --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \
    --fargate-profile-name my-fargate-profile \
    --selectors '[{"namespace": "prod*", "labels": {"labelname*?": "*value1"}},
 {"namespace": "*dev*", "labels": {"labelname*?": "*value*"}}]' \
    --subnets '["subnet-09d912bb63ef21b9a", "subnet-04ad87f71c6e5ab4d",
 "subnet-0e2907431c9988b72"]' \
    --tags '{"eks-fargate-profile-key-1": "value-1" , "eks-fargate-profile-key-2":
 "value-2"}'
```

輸出：

```
{
    "fargateProfile": {
        "fargateProfileName": "my-fargate-profile",
        "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-
eks-cluster/my-fargate-profile/e8c72bd6-5966-0bfe-b77b-1802893e5a6f",
        "clusterName": "my-eks-cluster",
        "createdAt": "2024-03-19T13:05:20.550000-04:00",
        "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "subnets": [
            "subnet-09d912bb63ef21b9a",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-0e2907431c9988b72"
        ],
        "selectors": [
            {
                "namespace": "prod*",
                "labels": {
                    "labelname*?": "*value1"
                }
            },
            {
```

```
                "namespace": "*dev*",
                "labels": {
                    "labelname*?": "*value*"
                }
            }
        ],
        "status": "CREATING",
        "tags": {
            "eks-fargate-profile-key-2": "value-2",
            "eks-fargate-profile-key-1": "value-1"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的 AWS Fargate 描述檔 - 建立 Fargate 描述檔。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFargateProfile。

create-nodegroup

以下程式碼範例顯示如何使用 create-nodegroup。

AWS CLI

範例 1：建立 Amazon EKS 叢集的受管節點群組

下列create-nodegroup範例會為 Amazon EKS 叢集建立受管節點群組。

```
aws eks create-nodegroup \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --node-role arn:aws:iam::111122223333:role/role-name \
    --
subnets "subnet-0e2907431c9988b72" "subnet-04ad87f71c6e5ab4d" "subnet-09d912bb63ef21b9a"
 \
    --scaling-config minSize=1,maxSize=3,desiredSize=1 \
    --region us-east-2
```

輸出：

```
{
    "nodegroup": {
```

```
        "nodegroupName": "my-eks-nodegroup",
        "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-
cluster/my-eks-nodegroup/bac7550f-b8b8-5fbb-4f3e-7502a931119e",
        "clusterName": "my-eks-cluster",
        "version": "1.26",
        "releaseVersion": "1.26.12-20240329",
        "createdAt": "2024-04-04T13:19:32.260000-04:00",
        "modifiedAt": "2024-04-04T13:19:32.260000-04:00",
        "status": "CREATING",
        "capacityType": "ON_DEMAND",
        "scalingConfig": {
            "minSize": 1,
            "maxSize": 3,
            "desiredSize": 1
        },
        "instanceTypes": [
            "t3.medium"
        ],
        "subnets": [
            "subnet-0e2907431c9988b72, subnet-04ad87f71c6e5ab4d,
 subnet-09d912bb63ef21b9a"
        ],
        "amiType": "AL2_x86_64",
        "nodeRole": "arn:aws:iam::111122223333:role/role-name",
        "diskSize": 20,
        "health": {
            "issues": []
        },
        "updateConfig": {
            "maxUnavailable": 1
        },
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的建立受管節點群組。

範例 2：為具有自訂執行個體類型和磁碟大小的 Amazon EKS 叢集建立受管節點群組

下列create-nodegroup範例會為具有自訂執行個體類型和磁碟大小的 Amazon EKS 叢集建立受管節點群組。

```
aws eks create-nodegroup \
```

```
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --node-role arn:aws:iam::111122223333:role/role-name \
    --
subnets "subnet-0e2907431c9988b72" "subnet-04ad87f71c6e5ab4d" "subnet-09d912bb63ef21b9a"
 \
    --scaling-config minSize=1,maxSize=3,desiredSize=1 \
    --capacity-type ON_DEMAND \
    --instance-types 'm5.large' \
    --disk-size 50 \
    --region us-east-2
```

輸出：

```
{
    "nodegroup": {
        "nodegroupName": "my-eks-nodegroup",
        "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-
cluster/my-eks-nodegroup/c0c7551b-e4f9-73d9-992c-a450fdb82322",
        "clusterName": "my-eks-cluster",
        "version": "1.26",
        "releaseVersion": "1.26.12-20240329",
        "createdAt": "2024-04-04T13:46:07.595000-04:00",
        "modifiedAt": "2024-04-04T13:46:07.595000-04:00",
        "status": "CREATING",
        "capacityType": "ON_DEMAND",
        "scalingConfig": {
            "minSize": 1,
            "maxSize": 3,
            "desiredSize": 1
        },
        "instanceTypes": [
            "m5.large"
        ],
        "subnets": [
            "subnet-0e2907431c9988b72",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-09d912bb63ef21b9a"
        ],
        "amiType": "AL2_x86_64",
        "nodeRole": "arn:aws:iam::111122223333:role/role-name",
        "diskSize": 50,
        "health": {
```

```
            "issues": []
        },
        "updateConfig": {
            "maxUnavailable": 1
        },
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的建立受管節點群組。

範例 3：為具有自訂執行個體類型、磁碟大小、ami 類型、容量類型、update-config、標籤、污點和標籤的 Amazon EKS 叢集建立受管節點群組。

下列create-nodegroup範例會為具有自訂執行個體類型、磁碟大小、ami-type、容量類型、update-config、標籤、污點和標籤的 Amazon EKS 叢集建立受管節點群組。

```
aws eks create-nodegroup  \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --node-role arn:aws:iam::111122223333:role/role-name \
    --
subnets "subnet-0e2907431c9988b72" "subnet-04ad87f71c6e5ab4d" "subnet-09d912bb63ef21b9a"
 \
    --scaling-config minSize=1,maxSize=5,desiredSize=4 \
    --instance-types 't3.large' \
    --disk-size 50 \
    --ami-type AL2_x86_64 \
    --capacity-type SPOT \
    --update-config maxUnavailable=2 \
    --labels '{"my-eks-nodegroup-label-1": "value-1" , "my-eks-nodegroup-label-2":
 "value-2"}' \
    --taints '{"key": "taint-key-1" , "value": "taint-value-1", "effect":
 "NO_EXECUTE"}' \
    --tags '{"my-eks-nodegroup-key-1": "value-1" , "my-eks-nodegroup-key-2":
 "value-2"}'
```

輸出：

```
{
    "nodegroup": {
        "nodegroupName": "my-eks-nodegroup",
```

```
        "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-
cluster/my-eks-nodegroup/88c75524-97af-0cb9-a9c5-7c0423ab5314",
        "clusterName": "my-eks-cluster",
        "version": "1.26",
        "releaseVersion": "1.26.12-20240329",
        "createdAt": "2024-04-04T14:05:07.940000-04:00",
        "modifiedAt": "2024-04-04T14:05:07.940000-04:00",
        "status": "CREATING",
        "capacityType": "SPOT",
        "scalingConfig": {
            "minSize": 1,
            "maxSize": 5,
            "desiredSize": 4
        },
        "instanceTypes": [
            "t3.large"
        ],
        "subnets": [
            "subnet-0e2907431c9988b72",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-09d912bb63ef21b9a"
        ],
        "amiType": "AL2_x86_64",
        "nodeRole": "arn:aws:iam::111122223333:role/role-name",
        "labels": {
            "my-eks-nodegroup-label-2": "value-2",
            "my-eks-nodegroup-label-1": "value-1"
        },
        "taints": [
            {
                "key": "taint-key-1",
                "value": "taint-value-1",
                "effect": "NO_EXECUTE"
            }
        ],
        "diskSize": 50,
        "health": {
            "issues": []
        },
        "updateConfig": {
            "maxUnavailable": 2
        },
        "tags": {
            "my-eks-nodegroup-key-1": "value-1",
```

```
            "my-eks-nodegroup-key-2": "value-2"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的建立受管節點群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateNodegroup。

`create-pod-identity-association`

以下程式碼範例顯示如何使用 `create-pod-identity-association`。

AWS CLI

範例 1：在 EKS 叢集中建立 EKS Pod 身分關聯

下列`create-pod-identity-association`範例會在 EKS 叢集中的服務帳戶與 IAM 角色之間建立 EKS Pod Identity 關聯。

```
aws eks create-pod-identity-association \
    --cluster-name eks-customer \
    --namespace default \
    --service-account default \
    --role-arn arn:aws:iam::111122223333:role/my-role
```

輸出：

```
{
    "association": {
        "clusterName": "eks-customer",
        "namespace": "default",
        "serviceAccount": "default",
        "roleArn": "arn:aws:iam::111122223333:role/my-role",
        "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-8mvwvh57cu74mgcst",
        "associationId": "a-8mvwvh57cu74mgcst",
        "tags": {},
        "createdAt": "2025-05-24T19:40:13.961000-05:00",
        "modifiedAt": "2025-05-24T19:40:13.961000-05:00"
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的了解 EKS Pod Identity 如何授予 Pod 對 AWS 服務的存取權。

範例 2：在具有標籤的 EKS 叢集中建立 EKS Pod 身分關聯

以下內容會在服務帳戶與 EKS 叢集中具有標籤的 IAM 角色之間`create-pod-identity-association`建立 EKS Pod Identity 關聯。

```
aws eks create-pod-identity-association \
    --cluster-name eks-customer \
    --namespace default \
    --service-account default \
    --role-arn arn:aws:iam::111122223333:role/my-role \
    --tags Key1=value1,Key2=value2
```

輸出：

```
{
    "association": {
        "clusterName": "eks-customer",
        "namespace": "default",
        "serviceAccount": "default",
        "roleArn": "arn:aws:iam::111122223333:role/my-role",
        "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-9njjin9gfghecgoda",
        "associationId": "a-9njjin9gfghecgoda",
        "tags": {
            "Key2": "value2",
            "Key1": "value1"
        },
        "createdAt": "2025-05-24T19:52:14.135000-05:00",
        "modifiedAt": "2025-05-24T19:52:14.135000-05:00"
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的了解 EKS Pod Identity 如何授予 Pod 對 AWS 服務的存取權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePodIdentityAssociation。

delete-access-entry

以下程式碼範例顯示如何使用 delete-access-entry。

AWS CLI

刪除與叢集相關聯的存取項目

以下內容會delete-access-entry刪除與名為 的 EKS 叢集相關聯的存取項目eks-customer。

```
aws eks delete-access-entry \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/Admin
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的刪除存取項目。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAccessEntry。

delete-addon

以下程式碼範例顯示如何使用 delete-addon。

AWS CLI

範例 1。刪除 Amazon EKS 附加元件，但保留 EKS 叢集上的附加元件軟體

下列delete-addon範例命令會刪除 Amazon EKS 附加元件，但會保留 EKS 叢集上的附加元件軟體。

```
aws eks delete-addon \
    --cluster-name my-eks-cluster \
    --addon-name my-eks-addon \
    --preserve
```

輸出：

```
{
    "addon": {
        "addonName": "my-eks-addon",
        "clusterName": "my-eks-cluster",
        "status": "DELETING",
```

```
        "addonVersion": "v1.9.3-eksbuild.7",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-
addon/a8c71ed3-944e-898b-9167-c763856af4b8",
        "createdAt": "2024-03-14T11:49:09.009000-04:00",
        "modifiedAt": "2024-03-14T12:03:49.776000-04:00",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱在 Amazon EKS 中管理 Amazon EKS 附加元件 - 刪除附加元件。

範例 2。刪除 Amazon EKS 附加元件，以及從 EKS 叢集刪除附加元件軟體

下列delete-addon範例命令會刪除 Amazon EKS 附加元件，也會從 EKS 叢集刪除附加元件軟體。

```
aws eks delete-addon \
    --cluster-name my-eks-cluster \
    --addon-name my-eks-addon
```

輸出：

```
{
    "addon": {
        "addonName": "my-eks-addon",
        "clusterName": "my-eks-cluster",
        "status": "DELETING",
        "addonVersion": "v1.15.1-eksbuild.1",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-
addon/bac71ed1-ec43-3bb6-88ea-f243cdb58954",
        "createdAt": "2024-03-14T11:45:31.983000-04:00",
        "modifiedAt": "2024-03-14T11:58:40.136000-04:00",
        "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱管理 Amazon EKS 附加元件 - 刪除 Amazon EKS 中的附加元件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAddon。

delete-cluster

以下程式碼範例顯示如何使用 delete-cluster。

AWS CLI

刪除 Amazon EKS 叢集控制平面

下列delete-cluster範例會刪除 Amazon EKS 叢集控制平面。

```
aws eks delete-cluster \
    --name my-eks-cluster
```

輸出：

```
{
    "cluster": {
        "name": "my-eks-cluster",
        "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster",
        "createdAt": "2024-03-14T11:31:44.348000-04:00",
        "version": "1.27",
        "endpoint": "https://DALSJ343KE23J3RN45653DSKJTT647TYD.yl4.us-
east-2.eks.amazonaws.com",
        "roleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-cluster-
ServiceRole-zMF6CBakwwbW",
        "resourcesVpcConfig": {
            "subnetIds": [
                "subnet-0fb75d2d8401716e7",
                "subnet-02184492f67a3d0f9",
                "subnet-04098063527aab776",
                "subnet-0e2907431c9988b72",
                "subnet-04ad87f71c6e5ab4d",
                "subnet-09d912bb63ef21b9a"
            ],
            "securityGroupIds": [
                "sg-0c1327f6270afbb36"
            ],
            "clusterSecurityGroupId": "sg-01c84d09d70f39a7f",
            "vpcId": "vpc-0012b8e1cc0abb17d",
```

```
            "endpointPublicAccess": true,
            "endpointPrivateAccess": true,
            "publicAccessCidrs": [
                "0.0.0.0/0"
            ]
        },
        "kubernetesNetworkConfig": {
            "serviceIpv4Cidr": "10.100.0.0/16",
            "ipFamily": "ipv4"
        },
        "logging": {
            "clusterLogging": [
                {
                    "types": [
                        "api",
                        "audit",
                        "authenticator",
                        "controllerManager",
                        "scheduler"
                    ],
                    "enabled": true
                }
            ]
        },
        "identity": {
            "oidc": {
                "issuer": "https://oidc.eks.us-east-2.amazonaws.com/id/
DALSJ343KE23J3RN45653DSKJTT647TYD"
            }
        },
        "status": "DELETING",
        "certificateAuthority": {
            "data": "XXX_CA_DATA_XXX"
        },
        "platformVersion": "eks.16",
        "tags": {
            "aws:cloudformation:stack-name": "eksctl-my-eks-cluster-cluster",
            "alpha.eksctl.io/cluster-name": "my-eks-cluster",
            "karpenter.sh/discovery": "my-eks-cluster",
            "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-
east-2:111122223333:stack/eksctl-my-eks-cluster-cluster/e752ea00-e217-11ee-
beae-0a9599c8c7ed",
            "auto-delete": "no",
            "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-eks-cluster",
```

```
            "EKS-Cluster-Name": "my-eks-cluster",
            "alpha.eksctl.io/cluster-oidc-enabled": "true",
            "aws:cloudformation:logical-id": "ControlPlane",
            "alpha.eksctl.io/eksctl-version": "0.173.0-dev
+a7ee89342.2024-03-01T03:40:57Z",
            "Name": "eksctl-my-eks-cluster-cluster/ControlPlane"
        },
        "accessConfig": {
            "authenticationMode": "API_AND_CONFIG_MAP"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的刪除 Amazon EKS 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCluster。

delete-fargate-profile

以下程式碼範例顯示如何使用 delete-fargate-profile。

AWS CLI

範例 1：為具有命名空間的選擇器建立 EKS Fargate 設定檔

下列delete-fargate-profile範例會為具有命名空間的選擇器建立 EKS Fargate 設定檔。

```
aws eks delete-fargate-profile \
    --cluster-name my-eks-cluster \
    --fargate-profile-name my-fargate-profile
```

輸出：

```
{
    "fargateProfile": {
        "fargateProfileName": "my-fargate-profile",
        "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-
eks-cluster/my-fargate-profile/1ac72bb3-3fc6-2631-f1e1-98bff53bed62",
        "clusterName": "my-eks-cluster",
        "createdAt": "2024-03-19T11:48:39.975000-04:00",
        "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "subnets": [
            "subnet-09d912bb63ef21b9a",
```

```
            "subnet-04ad87f71c6e5ab4d",
            "subnet-0e2907431c9988b72"
        ],
        "selectors": [
            {
                "namespace": "default",
                "labels": {
                    "foo": "bar"
                }
            }
        ],
        "status": "DELETING",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的 AWS Fargate 描述檔 - 刪除 Fargate。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFargateProfile。

delete-nodegroup

以下程式碼範例顯示如何使用 delete-nodegroup。

AWS CLI

範例 1：刪除 Amazon EKS 叢集的受管節點群組

下列delete-nodegroup範例會刪除 Amazon EKS 叢集的受管節點群組。

```
aws eks delete-nodegroup \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup
```

輸出：

```
{
    "nodegroup": {
        "nodegroupName": "my-eks-nodegroup",
        "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-
cluster/my-eks-nodegroup/1ec75f5f-0e21-dcc0-b46e-f9c442685cd8",
        "clusterName": "my-eks-cluster",
        "version": "1.26",
```

```
        "releaseVersion": "1.26.12-20240329",
        "createdAt": "2024-04-08T13:25:15.033000-04:00",
        "modifiedAt": "2024-04-08T13:25:31.252000-04:00",
        "status": "DELETING",
        "capacityType": "SPOT",
        "scalingConfig": {
            "minSize": 1,
            "maxSize": 5,
            "desiredSize": 4
        },
        "instanceTypes": [
            "t3.large"
        ],
        "subnets": [
            "subnet-0e2907431c9988b72",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-09d912bb63ef21b9a"
        ],
        "amiType": "AL2_x86_64",
        "nodeRole": "arn:aws:iam::111122223333:role/role-name",
        "labels": {
            "my-eks-nodegroup-label-2": "value-2",
            "my-eks-nodegroup-label-1": "value-1"
        },
        "taints": [
            {
                "key": "taint-key-1",
                "value": "taint-value-1",
                "effect": "NO_EXECUTE"
            }
        ],
        "diskSize": 50,
        "health": {
            "issues": []
        },
        "updateConfig": {
            "maxUnavailable": 2
        },
        "tags": {
            "my-eks-nodegroup-key-1": "value-1",
            "my-eks-nodegroup-key-2": "value-2"
        }
    }
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNodegroup。

delete-pod-identity-association

以下程式碼範例顯示如何使用 delete-pod-identity-association。

AWS CLI

刪除 EKS Pod 身分關聯

下列delete-pod-identity-association範例a-9njjin9gfghecgocd會從名為 的 EKS 叢集中刪除 EKS Pod Identity 關聯與關聯 IDeks-customer。

```
aws eks delete-pod-identity-association \
    --cluster-name eks-customer \
    --association-id a-9njjin9gfghecgocd
```

輸出：

```
{
    "association": {
        "clusterName": "eks-customer",
        "namespace": "default",
        "serviceAccount": "default",
        "roleArn": "arn:aws:iam::111122223333:role/s3-role",
        "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-9njjin9gfghecgocd",
        "associationId": "a-9njjin9gfghecgocd",
        "tags": {
            "Key2": "value2",
            "Key1": "value1"
        },
        "createdAt": "2025-05-24T19:52:14.135000-05:00",
        "modifiedAt": "2025-05-25T21:10:56.923000-05:00"
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的了解 EKS Pod Identity 如何授予 Pod 對 AWS 服務的存取權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePodIdentityAssociation。

deregister-cluster

以下程式碼範例顯示如何使用 deregister-cluster。

AWS CLI

取消註冊已連線的叢集，將其從 Amazon EKS 控制平面中移除

下列deregister-cluster範例會取消註冊已連線的叢集，以將其從 Amazon EKS 控制平面中移除。

```
aws eks deregister-cluster \
    --name my-eks-anywhere-cluster
```

輸出：

```
{
    "cluster": {
        "name": "my-eks-anywhere-cluster",
        "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-anywhere-cluster",
        "createdAt": "2024-04-12T12:38:37.561000-04:00",
        "status": "DELETING",
        "tags": {},
        "connectorConfig": {
            "activationId": "dfb5ad28-13c3-4e26-8a19-5b2457638c74",
            "activationExpiry": "2024-04-15T12:38:37.082000-04:00",
            "provider": "EKS_ANYWHERE",
            "roleArn": "arn:aws:iam::111122223333:role/AmazonEKSConnectorAgentRole"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的取消註冊叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterCluster。

describe-access-entry

以下程式碼範例顯示如何使用 describe-access-entry。

AWS CLI

描述 EKS 叢集的存取項目

下列describe-access-entry範例說明 EKS 叢集的存取項目。

```
aws eks describe-access-entry \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:user/eks-admin-user
```

輸出：

```
{
    "accessEntry": {
        "clusterName": "eks-customer",
        "principalArn": "arn:aws:iam::111122223333:user/eks-admin-user",
        "kubernetesGroups": [],
        "accessEntryArn": "arn:aws:eks:us-west-2:111122223333:access-entry/eks-
customer/user/111122223333/eks-admin-user/0acb1bc6-cb0a-ede6-11ae-a6506e3d36p0",
        "createdAt": "2025-04-14T22:45:48.097000-05:00",
        "modifiedAt": "2025-04-14T22:45:48.097000-05:00",
        "tags": {},
        "username": "arn:aws:iam::111122223333:user/eks-admin-user",
        "type": "STANDARD"
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的使用 EKS 存取項目授予 IAM 使用者對 Kubernetes 的存取權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAccessEntry。

describe-addon-configuration

以下程式碼範例顯示如何使用 describe-addon-configuration。

AWS CLI

範例 1：建立或更新 Amazon vpc-cni AddOns 時可用的組態選項

下列describe-addon-configuration範例會傳回您在為 vpc-cni 附加元件建立或更新具有個別版本的附加元件時所使用的所有可用組態結構描述。

```
aws eks describe-addon-configuration \
    --addon-name vpc-cni \
    --addon-version v1.15.1-eksbuild.1
```

輸出：

```
{
    "addonName": "vpc-cni",
    "addonVersion": "v1.15.1-eksbuild.1",
    "configurationSchema": "{\"$ref\":\"#/definitions/VpcCni\",\"$schema\":\"http://
json-schema.org/draft-06/schema#\",\"definitions\":{\"Affinity\":{\"type\":
[\"object\",\"null\"]},\"EniConfig\":{\"additionalProperties\":false,\"properties
\":{\"create\":{\"type\":\"boolean\"},\"region\":{\"type\":\"string\"},\"subnets\":
{\"additionalProperties\":{\"additionalProperties\":false,\"properties\":{\"id\":
{\"type\":\"string\"},\"securityGroups\":{\"items\":{\"type\":\"string\"},\"type\":
\"array\"}},\"required\":[\"id\"],\"type\":\"object\"},\"minProperties\":1,\"type
\":\"object\"}},\"required\":[\"create\",\"region\",\"subnets\"],\"type\":\"object
\"},\"Env\":{\"additionalProperties\":false,\"properties\":{\"ADDITIONAL_ENI_TAGS
\":{\"type\":\"string\"},\"ANNOTATE_POD_IP\":{\"format\":\"boolean\",\"type\":
\"string\"},\"AWS_EC2_ENDPOINT\":{\"type\":\"string\"},\"AWS_EXTERNAL_SERVICE_CIDRS
\":{\"type\":\"string\"},\"AWS_MANAGE_ENIS_NON_SCHEDULABLE\":{\"format\":\"boolean
\",\"type\":\"string\"},\"AWS_VPC_CNI_NODE_PORT_SUPPORT\":{\"format\":\"boolean
\",\"type\":\"string\"},\"AWS_VPC_ENI_MTU\":{\"format\":\"integer\",\"type\":
\"string\"},\"AWS_VPC_K8S_CNI_CUSTOM_NETWORK_CFG\":{\"format\":\"boolean\",\"type
\":\"string\"},\"AWS_VPC_K8S_CNI_EXCLUDE_SNAT_CIDRS\":{\"type\":\"string\"},
\"AWS_VPC_K8S_CNI_EXTERNALSNAT\":{\"format\":\"boolean\",\"type\":\"string\"},
\"AWS_VPC_K8S_CNI_LOGLEVEL\":{\"type\":\"string\"},\"AWS_VPC_K8S_CNI_LOG_FILE\":
{\"type\":\"string\"},\"AWS_VPC_K8S_CNI_RANDOMIZESNAT\":{\"type\":\"string\"},
\"AWS_VPC_K8S_CNI_VETHPREFIX\":{\"type\":\"string\"},\"AWS_VPC_K8S_PLUGIN_LOG_FILE
\":{\"type\":\"string\"},\"AWS_VPC_K8S_PLUGIN_LOG_LEVEL\":{\"type\":\"string
\"},\"CLUSTER_ENDPOINT\":{\"type\":\"string\"},\"DISABLE_INTROSPECTION\":
{\"format\":\"boolean\",\"type\":\"string\"},\"DISABLE_LEAKED_ENI_CLEANUP\":
{\"format\":\"boolean\",\"type\":\"string\"},\"DISABLE_METRICS\":{\"format
\":\"boolean\",\"type\":\"string\"},\"DISABLE_NETWORK_RESOURCE_PROVISIONING
\":{\"format\":\"boolean\",\"type\":\"string\"},\"DISABLE_POD_V6\":{\"format
\":\"boolean\",\"type\":\"string\"},\"ENABLE_BANDWIDTH_PLUGIN\":{\"format\":
\"boolean\",\"type\":\"string\"},\"ENABLE_POD_ENI\":{\"format\":\"boolean\",
\"type\":\"string\"},\"ENABLE_PREFIX_DELEGATION\":{\"format\":\"boolean\",
\"type\":\"string\"},\"ENABLE_V4_EGRESS\":{\"format\":\"boolean\",\"type\":
\"string\"},\"ENABLE_V6_EGRESS\":{\"format\":\"boolean\",\"type\":\"string\"},
\"ENI_CONFIG_ANNOTATION_DEF\":{\"type\":\"string\"},\"ENI_CONFIG_LABEL_DEF\":
{\"type\":\"string\"},\"INTROSPECTION_BIND_ADDRESS\":{\"type\":\"string\"},
\"IP_COOLDOWN_PERIOD\":{\"format\":\"integer\",\"type\":\"string\"},\"MAX_ENI
```

```
\":{\"format\":\"integer\",\"type\":\"string\"},\"MINIMUM_IP_TARGET\":{\"format
\":\"integer\",\"type\":\"string\"},\"POD_SECURITY_GROUP_ENFORCING_MODE\":
{\"type\":\"string\"},\"WARM_ENI_TARGET\":{\"format\":\"integer\",\"type\":
\"string\"},\"WARM_IP_TARGET\":{\"format\":\"integer\",\"type\":\"string\"},
\"WARM_PREFIX_TARGET\":{\"format\":\"integer\",\"type\":\"string\"}},\"title
\":\"Env\",\"type\":\"object\"},\"Init\":{\"additionalProperties\":false,
\"properties\":{\"env\":{\"$ref\":\"#/definitions/InitEnv\"}},\"title\":\"Init
\",\"type\":\"object\"},\"InitEnv\":{\"additionalProperties\":false,\"properties
\":{\"DISABLE_TCP_EARLY_DEMUX\":{\"format\":\"boolean\",\"type\":\"string\"},
\"ENABLE_V6_EGRESS\":{\"format\":\"boolean\",\"type\":\"string\"}},\"title\":
\"InitEnv\",\"type\":\"object\"},\"Limits\":{\"additionalProperties\":false,
\"properties\":{\"cpu\":{\"type\":\"string\"},\"memory\":{\"type\":\"string\"}},
\"title\":\"Limits\",\"type\":\"object\"},\"NodeAgent\":{\"additionalProperties
\":false,\"properties\":{\"enableCloudWatchLogs\":{\"format\":\"boolean\",
\"type\":\"string\"},\"enablePolicyEventLogs\":{\"format\":\"boolean\",\"type\":
\"string\"},\"healthProbeBindAddr\":{\"format\":\"integer\",\"type\":\"string
\"},\"metricsBindAddr\":{\"format\":\"integer\",\"type\":\"string\"}},\"title\":
\"NodeAgent\",\"type\":\"object\"},\"Resources\":{\"additionalProperties\":false,
\"properties\":{\"limits\":{\"$ref\":\"#/definitions/Limits\"},\"requests\":
{\"$ref\":\"#/definitions/Limits\"}},\"title\":\"Resources\",\"type\":\"object
\"},\"Tolerations\":{\"additionalProperties\":false,\"items\":{\"type\":\"object
\"},\"type\":\"array\"},\"VpcCni\":{\"additionalProperties\":false,\"properties
\":{\"affinity\":{\"$ref\":\"#/definitions/Affinity\"},\"enableNetworkPolicy\":
{\"format\":\"boolean\",\"type\":\"string\"},\"enableWindowsIpam\":{\"format\":
\"boolean\",\"type\":\"string\"},\"eniConfig\":{\"$ref\":\"#/definitions/EniConfig
\"},\"env\":{\"$ref\":\"#/definitions/Env\"},\"init\":{\"$ref\":\"#/definitions/Init
\"},\"livenessProbeTimeoutSeconds\":{\"type\":\"integer\"},\"nodeAgent\":{\"$ref\":
\"#/definitions/NodeAgent\"},\"readinessProbeTimeoutSeconds\":{\"type\":\"integer
\"},\"resources\":{\"$ref\":\"#/definitions/Resources\"},\"tolerations\":{\"$ref
\":\"#/definitions/Tolerations\"}},\"title\":\"VpcCni\",\"type\":\"object\"}},
\"description\":\"vpc-cni\"}"
}
```

範例 2：建立或更新 Amazon coredns AddOns 時可用的組態選項

下列describe-addon-configuration範例會針對具有個別版本的 coredns 附加元件建立或更新附加元件時，傳回您使用的所有可用組態結構描述。

```
aws eks describe-addon-configuration \
    --addon-name coredns \
    --addon-version v1.8.7-eksbuild.4
```

輸出：

```
{
    "addonName": "coredns",
    "addonVersion": "v1.8.7-eksbuild.4",
    "configurationSchema": "{\"$ref\":\"#/definitions/Coredns\",\"$schema
\":\"http://json-schema.org/draft-06/schema#\",\"definitions\":{\"Coredns\":
{\"additionalProperties\":false,\"properties\":{\"computeType\":{\"type\":
\"string\"},\"corefile\":{\"description\":\"Entire corefile contents to use with
 installation\",\"type\":\"string\"},\"nodeSelector\":{\"additionalProperties\":
{\"type\":\"string\"},\"type\":\"object\"},\"replicaCount\":{\"type\":\"integer
\"},\"resources\":{\"$ref\":\"#/definitions/Resources\"}},\"title\":\"Coredns\",
\"type\":\"object\"},\"Limits\":{\"additionalProperties\":false,\"properties\":
{\"cpu\":{\"type\":\"string\"},\"memory\":{\"type\":\"string\"}},\"title\":\"Limits
\",\"type\":\"object\"},\"Resources\":{\"additionalProperties\":false,\"properties
\":{\"limits\":{\"$ref\":\"#/definitions/Limits\"},\"requests\":{\"$ref\":\"#/
definitions/Limits\"}},\"title\":\"Resources\",\"type\":\"object\"}}}"
}
```

如需詳細資訊，請參閱 Amazon EKS 中的建立或更新 Amazon EKS 叢集的 kubeconfig 檔案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAddonConfiguration。

describe-addon-versions

以下程式碼範例顯示如何使用 describe-addon-versions。

AWS CLI

範例 1：列出 EKS 叢集的所有可用附加元件

下列describe-addon-versions範例列出所有可用的 AWS 附加元件。

```
aws eks describe-addon-versions  \
    --query 'sort_by(addons  &owner)[].{publisher: publisher, owner: owner,
 addonName: addonName, type: type}' \
    --output table
```

輸出：

```
-----------------------------------------------------------------------------------------------
|                                                DescribeAddonVersions
                                |
+---------------------------------------------+------------------
+-----------------------+-------------------------+
```

```
|                  addonName                  |      owner       |       publisher
     |          type           |
+---------------------------------------------+------------------
+----------------------+-------------------------+
|  vpc-cni                                    |  aws             |  eks
     |  networking             |
|  snapshot-controller                        |  aws             |  eks
     |  storage                |
|  kube-proxy                                 |  aws             |  eks
     |  networking             |
|  eks-pod-identity-agent                     |  aws             |  eks
     |  security               |
|  coredns                                    |  aws             |  eks
     |  networking             |
|  aws-mountpoint-s3-csi-driver               |  aws             |  s3
     |  storage                |
|  aws-guardduty-agent                        |  aws             |  eks
     |  security               |
|  aws-efs-csi-driver                         |  aws             |  eks
     |  storage                |
|  aws-ebs-csi-driver                         |  aws             |  eks
     |  storage                |
|  amazon-cloudwatch-observability            |  aws             |  eks
     |  observability          |
|  adot                                       |  aws             |  eks
     |  observability          |
|  upwind-security_upwind-operator            |  aws-marketplace |  Upwind Security
     |  security               |
|  upbound_universal-crossplane               |  aws-marketplace |  upbound
     |  infra-management       |
|  tetrate-io_istio-distro                    |  aws-marketplace |  tetrate-io
     |  policy-management      |
|  teleport_teleport                          |  aws-marketplace |  teleport
     |  policy-management      |
|  stormforge_optimize-live                   |  aws-marketplace |  StormForge
     |  cost-management        |
|  splunk_splunk-otel-collector-chart         |  aws-marketplace |  Splunk
     |  monitoring             |
|  solo-io_istio-distro                       |  aws-marketplace |  Solo.io
     |  service-mesh           |
|  rafay-systems_rafay-operator               |  aws-marketplace |  rafay-systems
     |  kubernetes-management  |
|  new-relic_kubernetes-operator              |  aws-marketplace |  New Relic
     |  observability          |
```

```
|  netapp_trident-operator                    |  aws-marketplace |  NetApp Inc.
     |  storage                |
|  leaksignal_leakagent                       |  aws-marketplace |  leaksignal
     |  monitoring             |
|  kubecost_kubecost                          |  aws-marketplace |  kubecost
     |  cost-management        |
|  kong_konnect-ri                            |  aws-marketplace |  kong
     |  ingress-service-type   |
|  kasten_k10                                 |  aws-marketplace |  Kasten by Veeam
     |  data-protection        |
|  haproxy-technologies_kubernetes-ingress-ee |  aws-marketplace |  HAProxy
 Technologies |  ingress-controller     |
|  groundcover_agent                          |  aws-marketplace |  groundcover
     |  monitoring             |
|  grafana-labs_kubernetes-monitoring         |  aws-marketplace |  Grafana Labs
     |  monitoring             |
|  factorhouse_kpow                           |  aws-marketplace |  factorhouse
     |  monitoring             |
|  dynatrace_dynatrace-operator               |  aws-marketplace |  dynatrace
     |  monitoring             |
|  datree_engine-pro                          |  aws-marketplace |  datree
     |  policy-management      |
|  datadog_operator                           |  aws-marketplace |  Datadog
     |  monitoring             |
|  cribl_cribledge                            |  aws-marketplace |  Cribl
     |  observability          |
|  calyptia_fluent-bit                        |  aws-marketplace |  Calyptia Inc
     |  observability          |
|  accuknox_kubearmor                         |  aws-marketplace |  AccuKnox
     |  security               |
+---------------------------------------------+------------------
+----------------------+-------------------------+
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的管理 Amazon EKS 附加元件 - 建立附加元件。

範例 2：列出支援 EKS 之指定 Kubernetes 版本的所有可用附加元件

下列`describe-addon-versions`範例列出 EKS 支援之指定 Kubernetes 版本的所有可用附加元件。

```
aws eks describe-addon-versions  \
    --kubernetes-version=1.26 \
```

```
    --query 'sort_by(addons  &owner)[].{publisher: publisher, owner: owner,
 addonName: addonName, type: type}' \
    --output table
```

輸出：

```
-----------------------------------------------------------------------------------------------
|                                                DescribeAddonVersions
                                |
+---------------------------------------------+------------------
+-----------------------+------------------------+
|                 addonName                   |      owner       |       publisher
     |          type           |
+---------------------------------------------+------------------
+-----------------------+------------------------+
|  vpc-cni                                    |  aws             |  eks
     |  networking             |
|  snapshot-controller                        |  aws             |  eks
     |  storage                |
|  kube-proxy                                 |  aws             |  eks
     |  networking             |
|  eks-pod-identity-agent                     |  aws             |  eks
     |  security               |
|  coredns                                    |  aws             |  eks
     |  networking             |
|  aws-mountpoint-s3-csi-driver               |  aws             |  s3
     |  storage                |
|  aws-guardduty-agent                        |  aws             |  eks
     |  security               |
|  aws-efs-csi-driver                         |  aws             |  eks
     |  storage                |
|  aws-ebs-csi-driver                         |  aws             |  eks
     |  storage                |
|  amazon-cloudwatch-observability            |  aws             |  eks
     |  observability          |
|  adot                                       |  aws             |  eks
     |  observability          |
|  upwind-security_upwind-operator            |  aws-marketplace |  Upwind Security
     |  security               |
|  tetrate-io_istio-distro                    |  aws-marketplace |  tetrate-io
     |  policy-management      |
|  stormforge_optimize-live                   |  aws-marketplace |  StormForge
     |  cost-management        |
```

```
|  splunk_splunk-otel-collector-chart         |  aws-marketplace |  Splunk
     |  monitoring             |
|  solo-io_istio-distro                       |  aws-marketplace |  Solo.io
     |  service-mesh           |
|  rafay-systems_rafay-operator               |  aws-marketplace |  rafay-systems
     |  kubernetes-management  |
|  new-relic_kubernetes-operator              |  aws-marketplace |  New Relic
     |  observability          |
|  netapp_trident-operator                    |  aws-marketplace |  NetApp Inc.
     |  storage                |
|  leaksignal_leakagent                       |  aws-marketplace |  leaksignal
     |  monitoring             |
|  kubecost_kubecost                          |  aws-marketplace |  kubecost
     |  cost-management        |
|  kong_konnect-ri                            |  aws-marketplace |  kong
     |  ingress-service-type   |
|  haproxy-technologies_kubernetes-ingress-ee |  aws-marketplace |  HAProxy
 Technologies |  ingress-controller     |
|  groundcover_agent                          |  aws-marketplace |  groundcover
     |  monitoring             |
|  grafana-labs_kubernetes-monitoring         |  aws-marketplace |  Grafana Labs
     |  monitoring             |
|  dynatrace_dynatrace-operator               |  aws-marketplace |  dynatrace
     |  monitoring             |
|  datadog_operator                           |  aws-marketplace |  Datadog
     |  monitoring             |
|  cribl_cribledge                            |  aws-marketplace |  Cribl
     |  observability          |
|  calyptia_fluent-bit                        |  aws-marketplace |  Calyptia Inc
     |  observability          |
|  accuknox_kubearmor                         |  aws-marketplace |  AccuKnox
     |  security               |
+---------------------------------------------+------------------
+----------------------+-------------------------+
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的管理 Amazon EKS 附加元件 - 建立附加元件。

範例 3：列出支援 EKS 之指定 Kubernetes 版本的所有可用 vpc-cni 附加元件版本

下列`describe-addon-versions`範例列出 EKS 支援之指定 Kubernetes 版本的所有可用 vpc-cni 附加元件版本。

```
aws eks describe-addon-versions \
    --kubernetes-version=1.26 \
    --addon-name=vpc-cni \
    --query='addons[].addonVersions[].addonVersion'
```

輸出：

```
[
    "v1.18.0-eksbuild.1",
    "v1.17.1-eksbuild.1",
    "v1.16.4-eksbuild.2",
    "v1.16.3-eksbuild.2",
    "v1.16.2-eksbuild.1",
    "v1.16.0-eksbuild.1",
    "v1.15.5-eksbuild.1",
    "v1.15.4-eksbuild.1",
    "v1.15.3-eksbuild.1",
    "v1.15.1-eksbuild.1",
    "v1.15.0-eksbuild.2",
    "v1.14.1-eksbuild.1",
    "v1.14.0-eksbuild.3",
    "v1.13.4-eksbuild.1",
    "v1.13.3-eksbuild.1",
    "v1.13.2-eksbuild.1",
    "v1.13.0-eksbuild.1",
    "v1.12.6-eksbuild.2",
    "v1.12.6-eksbuild.1",
    "v1.12.5-eksbuild.2",
    "v1.12.0-eksbuild.2"
]
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的管理 Amazon EKS 附加元件 - 建立附加元件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAddonVersions。

`describe-addon`

以下程式碼範例顯示如何使用 `describe-addon`。

AWS CLI

描述在您的 Amazon EKS 叢集中主動執行 EKS 附加元件

下列describe-addon範例會在 Amazon EKS 叢集中主動執行 EKS 附加元件。

```
aws eks describe-addon \
    --cluster-name my-eks-cluster \
    --addon-name vpc-cni
```

輸出：

```
{
    "addon": {
        "addonName": "vpc-cni",
        "clusterName": "my-eks-cluster",
        "status": "ACTIVE",
        "addonVersion": "v1.16.4-eksbuild.2",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/vpc-
cni/0ec71efc-98dd-3203-60b0-4b939b2a5e5f",
        "createdAt": "2024-03-14T13:18:45.417000-04:00",
        "modifiedAt": "2024-03-14T13:18:49.557000-04:00",
        "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-
cluster-addon-vpc-cni-Role1-YfakrqOC1UTm",
        "tags": {
            "eks-addon-key-3": "value-3",
            "eks-addon-key-4": "value-4"
        },
        "configurationValues": "resources:\n    limits:\n        cpu: '100m'\nenv:\n
    AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAddon。

describe-cluster

以下程式碼範例顯示如何使用 describe-cluster。

AWS CLI

描述在您的 Amazon EKS 叢集中主動執行 EKS 附加元件

下列describe-cluster範例會在 Amazon EKS 叢集中主動執行 EKS 附加元件。

```
aws eks describe-cluster \
    --name my-eks-cluster
```

輸出：

```
{
    "cluster": {
        "name": "my-eks-cluster",
        "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster",
        "createdAt": "2024-03-14T11:31:44.348000-04:00",
        "version": "1.26",
        "endpoint": "https://JSA79429HJDASKJDJ8223829MNDNASW.yl4.us-
east-2.eks.amazonaws.com",
        "roleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-cluster-
ServiceRole-zMF6CBakwwbW",
        "resourcesVpcConfig": {
            "subnetIds": [
                "subnet-0fb75d2d8401716e7",
                "subnet-02184492f67a3d0f9",
                "subnet-04098063527aab776",
                "subnet-0e2907431c9988b72",
                "subnet-04ad87f71c6e5ab4d",
                "subnet-09d912bb63ef21b9a"
            ],
            "securityGroupIds": [
                "sg-0c1327f6270afbb36"
            ],
            "clusterSecurityGroupId": "sg-01c84d09d70f39a7f",
            "vpcId": "vpc-0012b8e1cc0abb17d",
            "endpointPublicAccess": true,
            "endpointPrivateAccess": true,
            "publicAccessCidrs": [
                "22.19.18.2/32"
            ]
        },
        "kubernetesNetworkConfig": {
            "serviceIpv4Cidr": "10.100.0.0/16",
```

```
            "ipFamily": "ipv4"
        },
        "logging": {
            "clusterLogging": [
                {
                    "types": [
                        "api",
                        "audit",
                        "authenticator",
                        "controllerManager",
                        "scheduler"
                    ],
                    "enabled": true
                }
            ]
        },
        "identity": {
            "oidc": {
                "issuer": "https://oidc.eks.us-east-2.amazonaws.com/id/
JSA79429HJDASKJDJ8223829MNDNASW"
            }
        },
        "status": "ACTIVE",
        "certificateAuthority": {
            "data": "CA_DATA_STRING..."
        },
        "platformVersion": "eks.14",
        "tags": {
            "aws:cloudformation:stack-name": "eksctl-my-eks-cluster-cluster",
            "alpha.eksctl.io/cluster-name": "my-eks-cluster",
            "karpenter.sh/discovery": "my-eks-cluster",
            "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-
east-2:111122223333:stack/eksctl-my-eks-cluster-cluster/e752ea00-e217-11ee-
beae-0a9599c8c7ed",
            "auto-delete": "no",
            "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-eks-cluster",
            "EKS-Cluster-Name": "my-eks-cluster",
            "alpha.eksctl.io/cluster-oidc-enabled": "true",
            "aws:cloudformation:logical-id": "ControlPlane",
            "alpha.eksctl.io/eksctl-version": "0.173.0-dev
+a7ee89342.2024-03-01T03:40:57Z",
            "Name": "eksctl-my-eks-cluster-cluster/ControlPlane"
        },
        "health": {
```

```
            "issues": []
        },
        "accessConfig": {
            "authenticationMode": "API_AND_CONFIG_MAP"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCluster。

describe-fargate-profile

以下程式碼範例顯示如何使用 describe-fargate-profile。

AWS CLI

描述 Fargate 設定檔

下列describe-fargate-profile範例說明 Fargate 設定檔。

```
aws eks describe-fargate-profile \
    --cluster-name my-eks-cluster \
    --fargate-profile-name my-fargate-profile
```

輸出：

```
{
    "fargateProfile": {
        "fargateProfileName": "my-fargate-profile",
        "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-
eks-cluster/my-fargate-profile/96c766ce-43d2-f9c9-954c-647334391198",
        "clusterName": "my-eks-cluster",
        "createdAt": "2024-04-11T10:42:52.486000-04:00",
        "podExecutionRoleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-
cluster-farga-FargatePodExecutionRole-1htfAaJdJUEO",
        "subnets": [
            "subnet-09d912bb63ef21b9a",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-0e2907431c9988b72"
        ],
        "selectors": [
            {
                "namespace": "prod*",
```

```
                "labels": {
                    "labelname*?": "*value1"
                }
            },
            {
                "namespace": "*dev*",
                "labels": {
                    "labelname*?": "*value*"
                }
            }
        ],
        "status": "ACTIVE",
        "tags": {
            "eks-fargate-profile-key-2": "value-2",
            "eks-fargate-profile-key-1": "value-1"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFargateProfile。

describe-identity-provider-config

以下程式碼範例顯示如何使用 describe-identity-provider-config。

AWS CLI

描述與 Amazon EKS 叢集相關聯的身分提供者組態

下列describe-identity-provider-config範例說明與您的 Amazon EKS 叢集相關聯的身分提供者組態。

```
aws eks describe-identity-provider-config \
    --cluster-name my-eks-cluster \
    --identity-provider-config type=oidc,name=my-identity-provider
```

輸出：

```
{
    "identityProviderConfig": {
        "oidc": {
            "identityProviderConfigName": "my-identity-provider",
```

```
            "identityProviderConfigArn": "arn:aws:eks:us-
east-2:111122223333:identityproviderconfig/my-eks-cluster/oidc/my-identity-
provider/8ac76722-78e4-cec1-ed76-d49eea058622",
            "clusterName": "my-eks-cluster",
            "issuerUrl": "https://oidc.eks.us-east-2.amazonaws.com/
id/38D6A4619A0A69E342B113ED7F1A7652",
            "clientId": "kubernetes",
            "usernameClaim": "email",
            "usernamePrefix": "my-username-prefix",
            "groupsClaim": "my-claim",
            "groupsPrefix": "my-groups-prefix",
            "requiredClaims": {
                "Claim1": "value1",
                "Claim2": "value2"
            },
            "tags": {
                "env": "dev"
            },
            "status": "ACTIVE"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的從 OpenID Connect 身分提供者驗證叢集的使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeIdentityProviderConfig。

describe-insight

以下程式碼範例顯示如何使用 describe-insight。

AWS CLI

使用 EKS 叢集的 ID 取得洞見的詳細資訊

下列describe-insight範例會傳回使用叢集名稱和洞見 ID 所指定洞見的詳細資訊。

```
aws eks describe-insight \
    --cluster-name eks-customer \
    --id 38ea7a64-a14f-4e0e-95c7-8dbcab3c3623
```

輸出：

```
{
    "insight": {
        "id": "38ea7a64-a14f-4e0e-95c7-8dbcab3c3623",
        "name": "Kubelet version skew",
        "category": "UPGRADE_READINESS",
        "kubernetesVersion": "1.33",
        "lastRefreshTime": "2025-05-24T11:22:50-05:00",
        "lastTransitionTime": "2025-05-24T11:22:50-05:00",
        "description": "Checks for kubelet versions of worker nodes in the cluster
 to see if upgrade would cause noncompliance with supported Kubernetes kubelet
 version skew policy.",
        "insightStatus": {
            "status": "PASSING",
            "reason": "Node kubelet versions match the cluster control plane
 version."
        },
        "recommendation": "Upgrade your worker nodes to match the Kubernetes version
 of your cluster control plane.",
        "additionalInfo": {
            "Kubelet version skew policy": "https://kubernetes.io/releases/version-
skew-policy/#kubelet",
            "Updating a managed node group": "https://docs.aws.amazon.com/eks/
latest/userguide/update-managed-node-group.html"
        },
        "resources": [],
        "categorySpecificSummary": {
            "deprecationDetails": []
        }
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的檢視叢集洞察。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInsight。

describe-nodegroup

以下程式碼範例顯示如何使用 `describe-nodegroup`。

AWS CLI

描述 Amazon EKS 叢集的受管節點群組

下列describe-nodegroup範例說明 Amazon EKS 叢集的受管節點群組。

```
aws eks describe-nodegroup \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup
```

輸出：

```
{
    "nodegroup": {
        "nodegroupName": "my-eks-nodegroup",
        "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-
cluster/my-eks-nodegroup/a8c75f2f-df78-a72f-4063-4b69af3de5b1",
        "clusterName": "my-eks-cluster",
        "version": "1.26",
        "releaseVersion": "1.26.12-20240329",
        "createdAt": "2024-04-08T11:42:10.555000-04:00",
        "modifiedAt": "2024-04-08T11:44:12.402000-04:00",
        "status": "ACTIVE",
        "capacityType": "ON_DEMAND",
        "scalingConfig": {
            "minSize": 1,
            "maxSize": 3,
            "desiredSize": 1
        },
        "instanceTypes": [
            "t3.medium"
        ],
        "subnets": [
            "subnet-0e2907431c9988b72",
            "subnet-04ad87f71c6e5ab4d",
            "subnet-09d912bb63ef21b9a"
        ],
        "amiType": "AL2_x86_64",
        "nodeRole": "arn:aws:iam::111122223333:role/role-name",
        "labels": {},
        "resources": {
            "autoScalingGroups": [
                {
                    "name": "eks-my-eks-nodegroup-a8c75f2f-df78-
a72f-4063-4b69af3de5b1"
                }
            ]
```

```
        },
        "diskSize": 20,
        "health": {
            "issues": []
        },
        "updateConfig": {
            "maxUnavailable": 1
        },
        "tags": {}
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeNodegroup。

describe-pod-identity-association

以下程式碼範例顯示如何使用 describe-pod-identity-association。

AWS CLI

提供 Pod Identity 關聯的詳細資訊

下列describe-pod-identity-association範例說明 EKS 叢集中的 Pod Identity 關聯。

```
aws eks describe-pod-identity-association \
    --cluster-name eks-customer \
    --association-id a-9njjin9gfghecgocd
```

輸出：

```
{
    "association": {
        "clusterName": "eks-customer",
        "namespace": "default",
        "serviceAccount": "default",
        "roleArn": "arn:aws:iam::111122223333:role/my-role",
        "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-9njjin9gfghecgocd",
        "associationId": "a-9njjin9gfghecgocd",
        "tags": {
            "Key2": "value2",
            "Key1": "value1"
```

```
        },
        "createdAt": "2025-05-24T19:52:14.135000-05:00",
        "modifiedAt": "2025-05-24T19:52:14.135000-05:00"
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的了解 EKS Pod Identity 如何授予 Pod 對 AWS 服務的存取權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePodIdentityAssociation。

`describe-update`

以下程式碼範例顯示如何使用 `describe-update`。

AWS CLI

範例 1：描述叢集的更新

下列`describe-update`範例說明名為 之叢集的更新。

```
aws eks describe-update \
    --name my-eks-cluster \
    --update-id 10bddb13-a71b-425a-b0a6-71cd03e59161
```

輸出：

```
{
    "update": {
        "id": "10bddb13-a71b-425a-b0a6-71cd03e59161",
        "status": "Successful",
        "type": "EndpointAccessUpdate",
        "params": [
            {
                "type": "EndpointPublicAccess",
                "value": "false"
            },
            {
                "type": "EndpointPrivateAccess",
                "value": "true"
            }
        ],
```

```
        "createdAt": "2024-03-14T10:01:26.297000-04:00",
        "errors": []
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的更新 Amazon EKS 叢集 Kubernetes 版本。

範例 2：描述叢集的更新

下列describe-update範例說明名為 之叢集的更新。

```
aws eks describe-update \
    --name my-eks-cluster \
    --update-id e4994991-4c0f-475a-a040-427e6da52966
```

輸出：

```
{
    "update": {
        "id": "e4994991-4c0f-475a-a040-427e6da52966",
        "status": "Successful",
        "type": "AssociateEncryptionConfig",
        "params": [
            {
                "type": "EncryptionConfig",
                "value": "[{\"resources\":[\"secrets\"],\"provider\":{\"keyArn\":
\"arn:aws:kms:region-code:account:key/key\"}}]"
            }
        ],
        "createdAt": "2024-03-14T11:01:26.297000-04:00",
        "errors": []
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的更新 Amazon EKS 叢集 Kubernetes 版本。

範例 3：描述叢集的更新

下列describe-update範例說明名為 之叢集的更新。

```
aws eks describe-update \
    --name my-eks-cluster \
    --update-id b5f0ba18-9a87-4450-b5a0-825e6e84496f
```

輸出：

```
{
    "update": {
        "id": "b5f0ba18-9a87-4450-b5a0-825e6e84496f",
        "status": "Successful",
        "type": "VersionUpdate",
        "params": [
            {
                "type": "Version",
                "value": "1.29"
            },
            {
                "type": "PlatformVersion",
                "value": "eks.1"
            }
        ],
        "createdAt": "2024-03-14T12:05:26.297000-04:00",
        "errors": []
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的更新 Amazon EKS 叢集 Kubernetes 版本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeUpdate。

disassociate-access-policy

以下程式碼範例顯示如何使用 disassociate-access-policy。

AWS CLI

取消存取政策與存取項目的關聯

以下內容會disassociate-access-policy移除與存取項目相關聯的存取政策。

```
aws eks disassociate-access-policy \
```

```
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/Admin \
    --policy-arn arn:aws:eks::aws:cluster-access-policy/AmazonEKSEditPolicy
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的將存取政策與存取項目建立關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateAccessPolicy。

disassociate-identity-provider-config

以下程式碼範例顯示如何使用 disassociate-identity-provider-config。

AWS CLI

取消身分提供者與 Amazon EKS 叢集的關聯

下列disassociate-identity-provider-config範例會取消身分提供者與 Amazon EKS 叢集的關聯。

```
aws eks disassociate-identity-provider-config \
    --cluster-name my-eks-cluster \
    --identity-provider-config 'type=oidc,name=my-identity-provider'
```

輸出：

```
{
    "update": {
        "id": "5f78d14e-c57b-4857-a3e4-cf664ae20949",
        "status": "InProgress",
        "type": "DisassociateIdentityProviderConfig",
        "params": [
            {
                "type": "IdentityProviderConfig",
                "value": "[]"
            }
        ],
        "createdAt": "2024-04-11T13:53:43.314000-04:00",
        "errors": []
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的從 OpenID Connect 身分提供者驗證叢集的使用者 - 取消 OIDC 身分提供者與叢集的關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateIdentityProviderConfig。

get-token

以下程式碼範例顯示如何使用 get-token。

AWS CLI

範例 1：取得名為 `my-eks-cluster` 之 Amazon EKS 叢集的身分驗證字符

下列get-token範例取得名為 my-eks-cluster 之 Amazon EKS 叢集的身分驗證字符。

```
aws eks get-token \
    --cluster-name my-eks-cluster
```

輸出：

```
{
    "kind": "ExecCredential",
    "apiVersion": "client.authentication.k8s.io/v1beta1",
    "spec": {},
    "status": {
        "expirationTimestamp": "2024-04-11T20:59:56Z",
        "token": "k8s-aws-v1.EXAMPLE_TOKEN_DATA_STRING..."
    }
}
```

範例 2：在簽署權杖時擔任登入資料的此 roleARN，以取得名為 `my-eks-cluster` 之 Amazon EKS 叢集的身分驗證權杖

下列get-token範例會透過在簽署權杖時為登入資料擔任此 roleARN，取得名為 my-eks-cluster 之 Amazon EKS 叢集的身分驗證權杖。

```
aws eks get-token \
    --cluster-name my-eks-cluster \
    --role-arn arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-cluster-ServiceRole-j1k7AfTIQtnM
```

輸出：

```
{
    "kind": "ExecCredential",
    "apiVersion": "client.authentication.k8s.io/v1beta1",
    "spec": {},
    "status": {
        "expirationTimestamp": "2024-04-11T21:05:26Z",
        "token": "k8s-aws-v1.EXAMPLE_TOKEN_DATA_STRING..."
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetToken。

`list-access-entries`

以下程式碼範例顯示如何使用 `list-access-entries`。

AWS CLI

列出 EKS 叢集的存取項目

以下會`list-access-entries`傳回與 EKS 叢集 相關聯的存取項目清單`eks-customer`。

```
aws eks list-access-entries \
    --cluster-name eks-customer
```

輸出：

```
{
    "accessEntries": [
        "arn:aws:iam::111122223333:role/Admin",
        "arn:aws:iam::111122223333:role/admin-test-ip",
        "arn:aws:iam::111122223333:role/assume-worker-node-role",
        "arn:aws:iam::111122223333:user/eks-admin-user"
    ]
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的使用 EKS 存取項目授予 IAM 使用者對 Kubernetes 的存取權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAccessEntries。

list-access-policies

以下程式碼範例顯示如何使用 list-access-policies。

AWS CLI

列出所有可用的存取政策

此list-access-policies範例會傳回所有可用存取政策的清單。

```
aws eks list-access-policies
```

輸出：

```
{
    "accessPolicies": [
        {
            "name": "AmazonEKSAdminPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSAdminPolicy"
        },
        {
            "name": "AmazonEKSAdminViewPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSAdminViewPolicy"
        },
        {
            "name": "AmazonEKSAutoNodePolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSAutoNodePolicy"
        },
        {
            "name": "AmazonEKSBlockStorageClusterPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSBlockStorageClusterPolicy"
        },
        {
            "name": "AmazonEKSBlockStoragePolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSBlockStoragePolicy"
        },
        {
            "name": "AmazonEKSClusterAdminPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSClusterAdminPolicy"
        },
        {
```

```
            "name": "AmazonEKSComputeClusterPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSComputeClusterPolicy"
        },
        {
            "name": "AmazonEKSComputePolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSComputePolicy"
        },
        {
            "name": "AmazonEKSEditPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSEditPolicy"
        },
        {
            "name": "AmazonEKSHybridPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSHybridPolicy"
        },
        {
            "name": "AmazonEKSLoadBalancingClusterPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSLoadBalancingClusterPolicy"
        },
        {
            "name": "AmazonEKSLoadBalancingPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSLoadBalancingPolicy"
        },
        {
            "name": "AmazonEKSNetworkingClusterPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSNetworkingClusterPolicy"
        },
        {
            "name": "AmazonEKSNetworkingPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSNetworkingPolicy"
        },
        {
            "name": "AmazonEKSViewPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSViewPolicy"
        },
        {
            "name": "AmazonEMRJobPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEMRJobPolicy"
        },
```

```
        {
            "name": "AmazonSagemakerHyperpodClusterPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonSagemakerHyperpodClusterPolicy"
        },
        {
            "name": "AmazonSagemakerHyperpodControllerPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonSagemakerHyperpodControllerPolicy"
        },
        {
            "name": "AmazonSagemakerHyperpodSystemNamespacePolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonSagemakerHyperpodSystemNamespacePolicy"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的將存取政策與存取項目建立關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAccessPolicies。

`list-addons`

以下程式碼範例顯示如何使用 `list-addons`。

AWS CLI

列出 Amazon EKS 叢集中名為 `my-eks-cluster` 的所有已安裝附加元件

下列`list-addons`範例列出 Amazon EKS 叢集中名為 my-eks-cluster 的所有已安裝附加元件。

```
aws eks list-addons \
    --cluster-name my-eks-cluster
```

輸出：

```
{
    "addons": [
        "kube-proxy",
        "vpc-cni"
    ]
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAddons。

`list-associated-access-policies`

以下程式碼範例顯示如何使用 `list-associated-access-policies`。

AWS CLI

列出與存取項目相關聯的存取政策

下列`list-associated-access-policies`範例會傳回與 EKS 叢集中的存取項目相關聯的存取政策清單。

```
aws eks list-associated-access-policies \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/Admin
```

輸出：

```
{
    "associatedAccessPolicies": [
        {
            "policyArn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSAdminPolicy",
            "accessScope": {
                "type": "cluster",
                "namespaces": []
            },
            "associatedAt": "2025-05-24T17:26:22.935000-05:00",
            "modifiedAt": "2025-05-24T17:26:22.935000-05:00"
        }
    ],
    "clusterName": "eks-customer",
    "principalArn": "arn:aws:iam::111122223333:role/Admin"
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的使用 EKS 存取項目授予 IAM 使用者對 Kubernetes 的存取權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAssociatedAccessPolicies。

list-clusters

以下程式碼範例顯示如何使用 list-clusters。

AWS CLI

若要列出 Amazon EKS 叢集中名為 `my-eks-cluster` 的所有已安裝附加元件

下列list-clusters範例列出 Amazon EKS 叢集中名為 my-eks-cluster 的所有已安裝附加元件。

```
aws eks list-clusters
```

輸出：

```
{
    "clusters": [
        "prod",
        "qa",
        "stage",
        "my-eks-cluster"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListClusters。

list-fargate-profiles

以下程式碼範例顯示如何使用 list-fargate-profiles。

AWS CLI

若要列出 Amazon EKS 叢集中名為 `my-eks-cluster` 的所有 Fargate 設定檔

下列list-fargate-profiles範例列出 Amazon EKS 叢集中名為 my-eks-cluster 的所有 Fargate 設定檔。

```
aws eks list-fargate-profiles \
    --cluster-name my-eks-cluster
```

輸出：

```
{
    "fargateProfileNames": [
        "my-fargate-profile"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFargateProfiles。

`list-identity-provider-configs`

以下程式碼範例顯示如何使用 `list-identity-provider-configs`。

AWS CLI

列出與 Amazon EKS 叢集相關聯的身分提供者

下列`list-identity-provider-configs`範例列出與 Amazon EKS 叢集相關聯的身分提供者。

```
aws eks list-identity-provider-configs \
    --cluster-name my-eks-cluster
```

輸出：

```
{
    "identityProviderConfigs": [
        {
            "type": "oidc",
            "name": "my-identity-provider"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的從 OpenID Connect 身分提供者驗證叢集的使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListIdentityProviderConfigs。

`list-insights`

以下程式碼範例顯示如何使用 `list-insights`。

AWS CLI

列出指定叢集的所有洞見

下列list-insights範例會傳回針對指定叢集檢查的所有洞見清單。

```
aws eks list-insights \
--cluster-name eks-customer
```

輸出：

```
{
    "insights": [
        {
            "id": "38ea7a64-a14f-4e0e-95c7-8dbcab3c3616",
            "name": "Kubelet version skew",
            "category": "UPGRADE_READINESS",
            "kubernetesVersion": "1.33",
            "lastRefreshTime": "2025-05-24T11:22:50-05:00",
            "lastTransitionTime": "2025-05-24T11:22:50-05:00",
            "description": "Checks for kubelet versions of worker nodes in the
 cluster to see if upgrade would cause noncompliance with supported Kubernetes
 kubelet version skew policy.",
            "insightStatus": {
                "status": "PASSING",
                "reason": "Node kubelet versions match the cluster control plane
 version."
            }
        },
        {
            "id": "9cd91472-f99c-45a9-b7d7-54d4900dee23",
            "name": "EKS add-on version compatibility",
            "category": "UPGRADE_READINESS",
            "kubernetesVersion": "1.33",
            "lastRefreshTime": "2025-05-24T11:22:59-05:00",
            "lastTransitionTime": "2025-05-24T11:22:50-05:00",
            "description": "Checks version of installed EKS add-ons to ensure they
 are compatible with the next version of Kubernetes. ",
            "insightStatus": {
                "status": "PASSING",
                "reason": "All installed EKS add-on versions are compatible with
 next Kubernetes version."
            }
```

```
        },
        {
            "id": "0deb269d-b1e1-458c-a2b4-7a57f940c875",
            "name": "Cluster health issues",
            "category": "UPGRADE_READINESS",
            "kubernetesVersion": "1.33",
            "lastRefreshTime": "2025-05-24T11:22:59-05:00",
            "lastTransitionTime": "2025-05-24T11:22:50-05:00",
            "description": "Checks for any cluster health issues that prevent
 successful upgrade to the next Kubernetes version on EKS.",
            "insightStatus": {
                "status": "PASSING",
                "reason": "No cluster health issues detected."
            }
        },
        {
            "id": "214fa274-344f-420b-812a-5049ce72c9ww",
            "name": "kube-proxy version skew",
            "category": "UPGRADE_READINESS",
            "kubernetesVersion": "1.33",
            "lastRefreshTime": "2025-05-24T11:22:50-05:00",
            "lastTransitionTime": "2025-05-24T11:22:50-05:00",
            "description": "Checks version of kube-proxy in cluster to see if
 upgrade would cause noncompliance with supported Kubernetes kube-proxy version skew
 policy.",
            "insightStatus": {
                "status": "PASSING",
                "reason": "kube-proxy versions match the cluster control plane
 version."
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的檢視叢集洞察。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListInsights。

`list-nodegroups`

以下程式碼範例顯示如何使用 `list-nodegroups`。

AWS CLI

列出 Amazon EKS 叢集中的所有節點群組

下列list-nodegroups範例列出 Amazon EKS 叢集中的所有節點群組。

```
aws eks list-nodegroups \
    --cluster-name my-eks-cluster
```

輸出：

```
{
    "nodegroups": [
        "my-eks-managed-node-group",
        "my-eks-nodegroup"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListNodegroups。

list-pod-identity-associations

以下程式碼範例顯示如何使用 list-pod-identity-associations。

AWS CLI

範例 1：列出 EKS 叢集中的 Pod Identity 關聯

以下會list-pod-identity-associations傳回eks-customer與在所有命名空間和服務帳戶中名為 的 EKS 叢集相關聯的 Pod Identity 關聯清單。

```
aws eks list-pod-identity-associations \
    --cluster-name eks-customer
```

輸出：

```
{
    "associations": [
        {
            "clusterName": "eks-customer",
            "namespace": "default",
```

```
            "serviceAccount": "default",
            "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-9njjin9gfghecgocd",
            "associationId": "a-9njjin9gfghecgocd"
        },
        {
            "clusterName": "eks-customer",
            "namespace": "kube-system",
            "serviceAccount": "eks-customer",
            "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-dvtacahdvjn01ffbc",
            "associationId": "a-dvtacahdvjn01ffbc"
        },
        {
            "clusterName": "eks-customer",
            "namespace": "kube-system",
            "serviceAccount": "coredns",
            "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-yrpsdroc4ei7k6xps",
            "associationId": "a-yrpsdroc4ei7k6xps"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的了解 EKS Pod Identity 如何授予 Pod 對 AWS 服務的存取權。

範例 2：根據命名空間和服務帳戶列出 EKS 叢集中的 Pod Identity 關聯

以下內容會根據命名空間和服務帳戶，`list-pod-identity-associations`傳回 EKS 叢集中的 Pod Identity 關聯清單。

```
aws eks list-pod-identity-associations \
    --cluster-name eks-customer \
    --namespace kube-system \
    --service-account eks-customer
```

輸出：

```
{
    "associations": [
        {
```

```
            "clusterName": "eks-customer",
            "namespace": "kube-system",
            "serviceAccount": "eks-customer",
            "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-dvtacahdvjn01ffbc",
            "associationId": "a-dvtacahdvjn01ffbc"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的了解 EKS Pod Identity 如何授予 Pod 對 AWS 服務的存取權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPodIdentityAssociations。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

範例 1：列出 Amazon EKS 叢集 ARN 的所有標籤

下列`list-tags-for-resource`範例列出 Amazon EKS 叢集 ARN 的所有標籤。

```
aws eks list-tags-for-resource \
    --resource-arn arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster
```

輸出：

```
{
    "tags": {
        "aws:cloudformation:stack-name": "eksctl-my-eks-cluster-cluster",
        "alpha.eksctl.io/cluster-name": "my-eks-cluster",
        "karpenter.sh/discovery": "my-eks-cluster",
        "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-
east-2:111122223333:stack/eksctl-my-eks-cluster-cluster/e752ea00-e217-11ee-
beae-0a9599c8c7ed",
        "auto-delete": "no",
        "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-eks-cluster",
        "EKS-Cluster-Name": "my-eks-cluster",
        "alpha.eksctl.io/cluster-oidc-enabled": "true",
        "aws:cloudformation:logical-id": "ControlPlane",
```

```
        "alpha.eksctl.io/eksctl-version": "0.173.0-dev
+a7ee89342.2024-03-01T03:40:57Z",
        "Name": "eksctl-my-eks-cluster-cluster/ControlPlane"
    }
}
```

範例 2：列出 Amazon EKS 節點群組 ARN 的所有標籤

下列`list-tags-for-resource`範例列出 Amazon EKS 節點群組 ARN 的所有標籤。

```
aws eks list-tags-for-resource \
    --resource-arn arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-cluster/my-
eks-managed-node-group/60c71ed2-2cfb-020f-a5f4-ad32477f198c
```

輸出：

```
{
    "tags": {
        "aws:cloudformation:stack-name": "eksctl-my-eks-cluster-nodegroup-my-eks-
managed-node-group",
        "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-
east-2:111122223333:stack/eksctl-my-eks-cluster-nodegroup-my-eks-managed-node-group/
eaa20310-e219-11ee-b851-0ab9ad8228ff",
        "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-eks-cluster",
        "EKS-Cluster-Name": "my-eks-cluster",
        "alpha.eksctl.io/nodegroup-type": "managed",
        "NodeGroup Name 1": "my-eks-managed-node-group",
        "k8s.io/cluster-autoscaler/enabled": "true",
        "nodegroup-role": "worker",
        "alpha.eksctl.io/cluster-name": "my-eks-cluster",
        "alpha.eksctl.io/nodegroup-name": "my-eks-managed-node-group",
        "karpenter.sh/discovery": "my-eks-cluster",
        "NodeGroup Name 2": "AmazonLinux-Linux-Managed-NG-v1-26-v1",
        "auto-delete": "no",
        "k8s.io/cluster-autoscaler/my-eks-cluster": "owned",
        "aws:cloudformation:logical-id": "ManagedNodeGroup",
        "alpha.eksctl.io/eksctl-version": "0.173.0-dev
+a7ee89342.2024-03-01T03:40:57Z"
    }
}
```

範例 3：列出 Amazon EKS Fargate 分析 ARNe 上的所有標籤

下列list-tags-for-resource範例列出 Amazon EKS Fargate 設定檔 ARN 的所有標籤。

```
aws eks list-tags-for-resource  \
    --resource-arn arn:aws:eks:us-east-2:111122223333:fargateprofile/my-eks-cluster/my-fargate-profile/d6c76780-e541-0725-c816-36754cab734b
```

輸出：

```
{
    "tags": {
        "eks-fargate-profile-key-2": "value-2",
        "eks-fargate-profile-key-1": "value-1"
    }
}
```

範例 4：列出 Amazon EKS 附加元件 ARN 的所有標籤

下列list-tags-for-resource範例列出 Amazon EKS 附加元件 ARN 的所有標籤。

```
aws eks list-tags-for-resource \
    --resource-arn arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/vpc-cni/0ec71efc-98dd-3203-60b0-4b939b2a5e5f
```

輸出：

```
{
    "tags": {
        "eks-addon-key-2": "value-2",
        "eks-addon-key-1": "value-1"
    }
}
```

範例 5：列出 Amazon EKS OIDC 身分提供者 ARN 的所有標籤

下列list-tags-for-resource範例列出 Amazon EKS OIDC 身分提供者 ARN 的所有標籤。

```
aws eks list-tags-for-resource \
    --resource-arn arn:aws:eks:us-east-2:111122223333:identityproviderconfig/my-eks-cluster/oidc/my-identity-provider/8ac76722-78e4-cec1-ed76-d49eea058622
```

輸出：

```
{
    "tags": {
        "my-identity-provider": "test"
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`list-update`

以下程式碼範例顯示如何使用 `list-update`。

AWS CLI

範例 1：列出與 Amazon EKS 叢集名稱相關聯的更新

下列`list-updates`範例列出 Amazon EKS 叢集名稱的所有更新 IDs。

```
aws eks list-updates \
    --name my-eks-cluster
```

輸出：

```
{
    "updateIds": [
        "5f78d14e-c57b-4857-a3e4-cf664ae20949",
        "760e5a3f-adad-48c7-88d3-7ac283c09c26",
        "cd4ec863-bc55-47d5-a377-3971502f529b",
        "f12657ce-e869-4f17-b158-a82ab8b7d937"
    ]
}
```

範例 2：列出 Amazon EKS 節點群組的所有更新 IDs

下列`list-updates`範例列出 Amazon EKS 節點群組的所有更新 IDs。

```
aws eks list-updates \
    --name my-eks-cluster \
    --nodegroup-name my-eks-managed-node-group
```

輸出：

```
{
    "updateIds": [
        "8c6c1bef-61fe-42ac-a242-89412387b8e7"
    ]
}
```

範例 3：列出 Amazon EKS 附加元件上的所有更新 IDs

下列`list-updates`範例列出 Amazon EKS 附加元件的所有更新 IDs。

```
aws eks list-updates \
    --name my-eks-cluster \
    --addon-name vpc-cni
```

輸出：

```
{
    "updateIds": [
        "9cdba8d4-79fb-3c83-afe8-00b508d33268"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListUpdate。

`list-updates`

以下程式碼範例顯示如何使用 `list-updates`。

AWS CLI

列出叢集的更新

此範例命令會列出`example`預設區域中名為 之叢集的目前更新。

命令：

```
aws eks list-updates --name example
```

輸出：

```
{
```

```
    "updateIds": [
        "10bddb13-a71b-425a-b0a6-71cd03e59161"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListUpdates。

register-cluster

以下程式碼範例顯示如何使用 register-cluster。

AWS CLI

範例 1：向 Amazon EKS 註冊外部 EKS_ANYWHERE Kubernetes 叢集

下列register-cluster範例會將外部 EKS_ANYWHERE Kubernetes 叢集註冊至 Amazon EKS。

```
aws eks register-cluster \
    --name my-eks-anywhere-cluster \
    --connector-config 'roleArn=arn:aws:iam::111122223333:role/
AmazonEKSConnectorAgentRole,provider=EKS_ANYWHERE'
```

輸出：

```
{
    "cluster": {
        "name": "my-eks-anywhere-cluster",
        "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-anywhere-cluster",
        "createdAt": "2024-04-12T12:38:37.561000-04:00",
        "status": "PENDING",
        "tags": {},
        "connectorConfig": {
            "activationId": "xxxxxxxxACTIVATION_IDxxxxxxxx",
            "activationCode": "xxxxxxxxACTIVATION_CODExxxxxxxx",
            "activationExpiry": "2024-04-15T12:38:37.082000-04:00",
            "provider": "EKS_ANYWHERE",
            "roleArn": "arn:aws:iam::111122223333:role/AmazonEKSConnectorAgentRole"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的連接外部叢集。

範例 2：向 Amazon EKS 註冊任何外部 Kubernetes 叢集

下列register-cluster範例會將外部 EKS_ANYWHERE Kubernetes 叢集註冊至 Amazon EKS。

```
aws eks register-cluster \
    --name my-eks-anywhere-cluster \
    --connector-config 'roleArn=arn:aws:iam::111122223333:role/
AmazonEKSConnectorAgentRole,provider=OTHER'
```

輸出：

```
{
    "cluster": {
        "name": "my-onprem-k8s-cluster",
        "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-onprem-k8s-cluster",
        "createdAt": "2024-04-12T12:42:10.861000-04:00",
        "status": "PENDING",
        "tags": {},
        "connectorConfig": {
            "activationId": "xxxxxxxxACTIVATION_IDxxxxxxxx",
            "activationCode": "xxxxxxxxACTIVATION_CODExxxxxxxx",
            "activationExpiry": "2024-04-15T12:42:10.339000-04:00",
            "provider": "OTHER",
            "roleArn": "arn:aws:iam::111122223333:role/AmazonEKSConnectorAgentRole"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的連接外部叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterCluster。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

範例 1：將指定的標籤新增至 Amazon EKS 叢集

下列tag-resource範例會將指定的標籤新增至 Amazon EKS 叢集。

```
aws eks tag-resource \
    --resource-arn arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster \
    --tag 'my-eks-cluster-test-1=test-value-1,my-eks-cluster-dev-1=dev-value-2'
```

此命令不會產生輸出。

範例 2：將指定的標籤新增至 Amazon EKS 節點群組

下列tag-resource範例會將指定的標籤新增至 Amazon EKS 節點群組。

```
aws eks tag-resource \
    --resource-arn arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-cluster/my-eks-managed-node-group/60c71ed2-2cfb-020f-a5f4-ad32477f198c \
    --tag 'my-eks-nodegroup-test-1=test-value-1,my-eks-nodegroup-dev-1=dev-value-2'
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

範例 1：從 Amazon EKS 叢集刪除指定的標籤

下列untag-resource範例會從 Amazon EKS 叢集刪除指定的標籤。

```
aws eks untag-resource \
    --resource-arn arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster \
    --tag-keys "my-eks-cluster-test-1" "my-eks-cluster-dev-1"
```

此命令不會產生輸出。

範例 2：從 Amazon EKS 節點群組刪除指定的標籤

下列untag-resource範例會從 Amazon EKS 節點群組刪除指定的標籤。

```
aws eks untag-resource \
    --resource-arn arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-cluster/my-eks-managed-node-group/60c71ed2-2cfb-020f-a5f4-ad32477f198c \
    --tag-keys "my-eks-nodegroup-test-1" "my-eks-nodegroup-dev-1"
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-access-entry

以下程式碼範例顯示如何使用 update-access-entry。

AWS CLI

更新 EKS 叢集的存取項目

以下透過新增 Kubernetes 群組 來update-access-entry更新 EKS 叢集的存取項目tester。

```
aws eks update-access-entry \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/Admin \
    --kubernetes-groups tester
```

輸出：

```
{
    "accessEntry": {
        "clusterName": "eks-customer",
        "principalArn": "arn:aws:iam::111122223333:role/Admin",
        "kubernetesGroups": [
            "tester"
        ],
        "accessEntryArn": "arn:aws:eks:us-west-2:111122223333:access-entry/eks-customer/role/111122223333/Admin/d2cb8183-d6ec-b82a-d967-eca21902a4b4",
        "createdAt": "2025-05-24T11:02:04.432000-05:00",
        "modifiedAt": "2025-05-24T17:08:01.608000-05:00",
        "tags": {},
        "username": "arn:aws:sts::111122223333:assumed-role/Admin/{{SessionName}}",
        "type": "STANDARD"
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的更新存取項目。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAccessEntry。

update-addon

以下程式碼範例顯示如何使用 update-addon。

AWS CLI

範例 1。使用服務帳戶角色 ARN 更新 Amazon EKS 附加元件

下列update-addon範例命令會使用服務帳戶角色 ARN 更新 Amazon EKS 附加元件。

```
aws eks update-addon \
    --cluster-name my-eks-cluster \
    --addon-name vpc-cni \
    --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-YfakrqOC1UTm
```

輸出：

```
{
    "update": {
        "id": "c00d2de2-c2e4-3d30-929e-46b8edec2ce4",
        "status": "InProgress",
        "type": "AddonUpdate",
        "params": [
            {
                "type": "ServiceAccountRoleArn",
                "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-YfakrqOC1UTm"
            }
        ],
        "updatedAt": "2024-04-12T16:04:55.614000-04:00",
        "errors": []
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的管理 Amazon EKS 附加元件 - 更新附加元件。

範例 2。使用特定附加元件版本更新 Amazon EKS 附加元件

下列update-addon範例命令會使用特定的附加元件版本更新 Amazon EKS 附加元件。

```
aws eks update-addon \
    --cluster-name my-eks-cluster \
    --addon-name vpc-cni \
    --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-YfakrqOC1UTm \
    --addon-version v1.16.4-eksbuild.2
```

輸出：

```
{
    "update": {
        "id": "f58dc0b0-2b18-34bd-bc6a-e4abc0011f36",
        "status": "InProgress",
        "type": "AddonUpdate",
        "params": [
            {
                "type": "AddonVersion",
                "value": "v1.16.4-eksbuild.2"
            },
            {
                "type": "ServiceAccountRoleArn",
                "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-YfakrqOC1UTm"
            }
        ],
        "createdAt": "2024-04-12T16:07:16.550000-04:00",
        "errors": []
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的管理 Amazon EKS 附加元件 - 更新附加元件。

範例 3. 使用自訂組態值更新 Amazon EKS 附加元件並解決衝突詳細資訊

下列update-addon範例命令會使用自訂組態值更新 Amazon EKS 附加元件，並解決衝突詳細資訊。

```
aws eks update-addon \
```

```
    --cluster-name my-eks-cluster \
    --addon-name vpc-cni \
    --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-YfakrqOC1UTm \
    --addon-version v1.16.4-eksbuild.2 \
    --configuration-values '{"resources": {"limits":{"cpu":"100m"}, "requests":{"cpu":"50m"}}}' \
    --resolve-conflicts PRESERVE
```

輸出：

```
{
    "update": {
        "id": "cd9f2173-a8d8-3004-a90f-032f14326520",
        "status": "InProgress",
        "type": "AddonUpdate",
        "params": [
            {
                "type": "AddonVersion",
                "value": "v1.16.4-eksbuild.2"
            },
            {
                "type": "ServiceAccountRoleArn",
                "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-YfakrqOC1UTm"
            },
            {
                "type": "ResolveConflicts",
                "value": "PRESERVE"
            },
            {
                "type": "ConfigurationValues",
                "value": "{\"resources\": {\"limits\":{\"cpu\":\"100m\"}, \"requests\":{\"cpu\":\"50m\"}}}"
            }
        ],
        "createdAt": "2024-04-12T16:16:27.363000-04:00",
        "errors": []
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的管理 Amazon EKS 附加元件 - 更新附加元件。

範例 4. 使用自訂 JSON 組態值檔案更新 Amazon EKS 附加元件

下列update-addon範例命令會使用自訂 JSON 組態值更新 Amazon EKS 附加元件，並解決衝突詳細資訊。

```
aws eks update-addon \
    --cluster-name my-eks-cluster \
    --addon-name vpc-cni \
    --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-YfakrqOC1UTm \
    --addon-version v1.17.1-eksbuild.1 \
    --configuration-values 'file://configuration-values.json' \
    --resolve-conflicts PRESERVE
```

configuration-values.json 的內容：

```
{
    "resources": {
        "limits": {
            "cpu": "100m"
        },
        "requests": {
            "cpu": "50m"
        }
    },
    "env": {
        "AWS_VPC_K8S_CNI_LOGLEVEL": "ERROR"
    }
}
```

輸出：

```
{
    "update": {
        "id": "6881a437-174f-346b-9a63-6e91763507cc",
        "status": "InProgress",
        "type": "AddonUpdate",
        "params": [
            {
                "type": "AddonVersion",
                "value": "v1.17.1-eksbuild.1"
            },
```

```
            {
                "type": "ServiceAccountRoleArn",
                "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-YfakrqOC1UTm"
            },
            {
                "type": "ResolveConflicts",
                "value": "PRESERVE"
            },
            {
                "type": "ConfigurationValues",
                "value": "{\n    \"resources\": {\n        \"limits\": {\n
  \"cpu\": \"100m\"\n        },\n        \"requests\": {\n            \"cpu\": \"50m
\"\n        }\n    },\n    \"env\": {\n        \"AWS_VPC_K8S_CNI_LOGLEVEL\": \"ERROR
\"\n    }\n}"
            }
        ],
        "createdAt": "2024-04-12T16:22:55.519000-04:00",
        "errors": []
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的管理 Amazon EKS 附加元件 - 更新附加元件。

範例 5. 使用自訂 YAML 組態值檔案更新 Amazon EKS 附加元件

下列update-addon範例命令會使用自訂 YAML 組態值更新 Amazon EKS 附加元件，並解決衝突詳細資訊。

```
aws eks update-addon \
    --cluster-name my-eks-cluster \
    --addon-name vpc-cni \
    --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-YfakrqOC1UTm \
    --addon-version v1.18.0-eksbuild.1 \
    --configuration-values 'file://configuration-values.yaml' \
    --resolve-conflicts PRESERVE
```

configuration-values.yaml 的內容：

```
resources:
```

```
    limits:
        cpu: '100m'
    requests:
        cpu: '50m'
env:
    AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'
```

輸出：

```
{
    "update": {
        "id": "a067a4c9-69d0-3769-ace9-d235c5b16701",
        "status": "InProgress",
        "type": "AddonUpdate",
        "params": [
            {
                "type": "AddonVersion",
                "value": "v1.18.0-eksbuild.1"
            },
            {
                "type": "ServiceAccountRoleArn",
                "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-YfakrqOC1UTm"
            },
            {
                "type": "ResolveConflicts",
                "value": "PRESERVE"
            },
            {
                "type": "ConfigurationValues",
                "value": "resources:\n    limits:\n        cpu: '100m'\n
 requests:\n        cpu: '50m'\nenv:\n    AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'"
            }
        ],
        "createdAt": "2024-04-12T16:25:07.212000-04:00",
        "errors": []
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的管理 Amazon EKS 附加元件 - 更新附加元件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAddon。

update-cluster-config

以下程式碼範例顯示如何使用 update-cluster-config。

AWS CLI

更新叢集端點存取

此範例命令會更新叢集，以停用端點公有存取並啟用私有端點存取。

命令：

```
aws eks update-cluster-config --name example \
--resources-vpc-config endpointPublicAccess=false,endpointPrivateAccess=true
```

輸出：

```
{
    "update": {
        "id": "ec883c93-2e9e-407c-a22f-8f6fa6e67d4f",
        "status": "InProgress",
        "type": "EndpointAccessUpdate",
        "params": [
            {
                "type": "EndpointPublicAccess",
                "value": "false"
            },
            {
                "type": "EndpointPrivateAccess",
                "value": "true"
            }
        ],
        "createdAt": 1565806986.506,
        "errors": []
    }
}
```

啟用叢集的記錄

此範例命令會啟用名為 之叢集的所有叢集控制平面記錄類型example。

命令：

```
aws eks update-cluster-config --name example \
--logging '{"clusterLogging":[{"types":
["api","audit","authenticator","controllerManager","scheduler"],"enabled":true}]}'
```

輸出：

```
{
    "update": {
        "id": "7551c64b-1d27-4b1e-9f8e-c45f056eb6fd",
        "status": "InProgress",
        "type": "LoggingUpdate",
        "params": [
            {
                "type": "ClusterLogging",
                "value": "{\"clusterLogging\":[{\"types\":[\"api\",\"audit\",
\"authenticator\",\"controllerManager\",\"scheduler\"],\"enabled\":true}]}"
            }
        ],
        "createdAt": 1565807210.37,
        "errors": []
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateClusterConfig。

update-cluster-version

以下程式碼範例顯示如何使用 update-cluster-version。

AWS CLI

將名為 `my-eks-cluster` 的 Amazon EKS 叢集更新為指定的 Kubernetes 版本

下列update-cluster-version範例會將 Amazon EKS 叢集更新為指定的 Kubernetes 版本。

```
aws eks update-cluster-version \
    --name my-eks-cluster \
    --kubernetes-version 1.27
```

輸出：

```
{
    "update": {
        "id": "e4091a28-ea14-48fd-a8c7-975aeb469e8a",
        "status": "InProgress",
        "type": "VersionUpdate",
        "params": [
            {
                "type": "Version",
                "value": "1.27"
            },
            {
                "type": "PlatformVersion",
                "value": "eks.16"
            }
        ],
        "createdAt": "2024-04-12T16:56:01.082000-04:00",
        "errors": []
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的更新 Amazon EKS 叢集 Kubernetes 版本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateClusterVersion。

update-kubeconfig

以下程式碼範例顯示如何使用 update-kubeconfig。

AWS CLI

範例 1：透過建立或更新 kubeconfig 來設定您的 kubectl，以便您可以連線到名為 `my-eks-cluster` 的 Amazon EKS 叢集

下列update-kubeconfig範例會透過建立或更新 kubeconfig 來設定您的 kubectl，以便您可以連線至名為 my-eks-cluster 的 Amazon EKS 叢集。

```
aws eks update-kubeconfig \
    --name my-eks-cluster
```

輸出：

```
Updated context arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster in /Users/
xxx/.kube/config
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的建立或更新 Amazon EKS 叢集的 kubeconfig 檔案。

範例 2：透過建立或更新 kubeconfig （使用 role-arn 選項來擔任叢集身分驗證的角色）來設定 kubectl，以便您可以連線到名為 `my-eks-cluster` 的 Amazon EKS 叢集

下列update-kubeconfig範例會透過建立或更新 kubeconfig （使用 role-arn 選項來擔任叢集身分驗證的角色）來設定您的 kubectl，以便您可以連線至名為 my-eks-cluster 的 Amazon EKS 叢集。

```
aws eks update-kubeconfig \
    --name my-eks-cluster \
    --role-arn arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-
cluster-ServiceRole-j1k7AfTIQtnM
```

輸出：

```
Updated context arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster in /Users/
xxx/.kube/config
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的建立或更新 Amazon EKS 叢集的 kubeconfig 檔案。

範例 3：建立或更新 kubeconfig 來設定您的 kubectl （使用 role-arn 選項來擔任叢集身分驗證的角色，以及自訂叢集別名和使用者別名），以便您可以連線至名為 `my-eks-cluster` 的 Amazon EKS 叢集

下列update-kubeconfig範例透過建立或更新 kubeconfig （使用 role-arn 選項來擔任叢集身分驗證的角色，以及自訂叢集別名和使用者別名）來設定您的 kubectl，以便您可以連線至名為 my-eks-cluster 的 Amazon EKS 叢集。

```
aws eks update-kubeconfig \
    --name my-eks-cluster \
    --role-arn arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-
cluster-ServiceRole-j1k7AfTIQtnM \
    --alias stage-eks-cluster \
```

```
    --user-alias john
```

輸出：

```
Updated context stage-eks-cluster in /Users/dubaria/.kube/config
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的建立或更新 Amazon EKS 叢集的 kubeconfig 檔案。

範例 4：列印 kubeconfig 檔案項目以供檢閱和設定 kubectl，以便您可以連線至名為 `my-eks-cluster` 的 Amazon EKS 叢集

下列update-kubeconfig範例透過建立或更新 kubeconfig（使用 role-arn 選項來擔任叢集身分驗證的角色，以及自訂叢集別名和使用者別名）來設定您的 kubectl，以便您可以連線至名為 my-eks-cluster 的 Amazon EKS 叢集。

```
aws eks update-kubeconfig \
    --name my-eks-cluster \
    --role-arn arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-cluster-ServiceRole-j1k7AfTIQtnM \
    --alias stage-eks-cluster \
    --user-alias john \
    --verbose
```

輸出：

```
Updated context stage-eks-cluster in /Users/dubaria/.kube/config
Entries:

context:
cluster: arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster
user: john
name: stage-eks-cluster

name: john
user:
exec:
    apiVersion: client.authentication.k8s.io/v1beta1
    args:
    - --region
    - us-east-2
    - eks
```

```
    - get-token
    - --cluster-name
    - my-eks-cluster
    - --output
    - json
    - --role
    - arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-cluster-
ServiceRole-j1k7AfTIQtnM
    command: aws

cluster:
certificate-authority-data: xxx_CA_DATA_xxx
server: https://DALSJ343KE23J3RN45653DSKJTT647TYD.yl4.us-east-2.eks.amazonaws.com
name: arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的建立或更新 Amazon EKS 叢集的 kubeconfig 檔案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateKubeconfig。

update-nodegroup-config

以下程式碼範例顯示如何使用 update-nodegroup-config。

AWS CLI

範例 1：更新受管節點群組，將 Amazon EKS 叢集的新標籤和污點新增至 EKS 工作者節點

下列update-nodegroup-config範例會更新受管節點群組，將新標籤和污點新增至 Amazon EKS 叢集的 EKS 工作者節點。

```
aws eks update-nodegroup-config \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --labels 'addOrUpdateLabels={my-eks-nodegroup-label-1=value-1,my-eks-nodegroup-
label-2=value-2}' \
    --taints 'addOrUpdateTaints=[{key=taint-key-1,value=taint-
value-1,effect=NO_EXECUTE}]'
```

輸出：

```
{
```

```
    "update": {
        "id": "e66d21d3-bd8b-3ad1-a5aa-b196dc08c7c1",
        "status": "InProgress",
        "type": "ConfigUpdate",
        "params": [
            {
                "type": "LabelsToAdd",
                "value": "{\"my-eks-nodegroup-label-2\":\"value-2\",\"my-eks-
nodegroup-label-1\":\"value-1\"}"
            },
            {
                "type": "TaintsToAdd",
                "value": "[{\"effect\":\"NO_EXECUTE\",\"value\":\"taint-value-1\",
\"key\":\"taint-key-1\"}]"
            }
        ],
        "createdAt": "2024-04-08T12:05:19.161000-04:00",
        "errors": []
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的更新受管節點群組。

範例 2：更新受管節點群組以移除 Amazon EKS 叢集 EKS 工作者節點的標籤和污點

下列update-nodegroup-config範例會更新受管節點群組，以移除 Amazon EKS 叢集 EKS 工作者節點的標籤和污點。

```
aws eks update-nodegroup-config \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --labels 'removeLabels=my-eks-nodegroup-label-1, my-eks-nodegroup-label-2' \
    --taints 'removeTaints=[{key=taint-key-1,value=taint-
value-1,effect=NO_EXECUTE}]'
```

輸出：

```
{
    "update": {
        "id": "67a08692-9e59-3ace-a916-13929f44cec3",
        "status": "InProgress",
        "type": "ConfigUpdate",
        "params": [
```

```
            {
                "type": "LabelsToRemove",
                "value": "[\"my-eks-nodegroup-label-1\",\"my-eks-nodegroup-
label-2\"]"
            },
            {
                "type": "TaintsToRemove",
                "value": "[{\"effect\":\"NO_EXECUTE\",\"value\":\"taint-value-1\",
\"key\":\"taint-key-1\"}]"
            }
        ],
        "createdAt": "2024-04-08T12:17:31.817000-04:00",
        "errors": []
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的更新受管節點群組。

範例 3：更新受管節點群組以移除和新增 Amazon EKS 叢集 EKS 工作者節點的標籤和污點

下列update-nodegroup-config範例會更新受管節點群組，以移除和新增 Amazon EKS 叢集 EKS 工作者節點的標籤和污點。

```
aws eks update-nodegroup-config \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --labels 'addOrUpdateLabels={my-eks-nodegroup-new-label-1=new-value-1,my-eks-
nodegroup-new-label-2=new-value-2},removeLabels=my-eks-nodegroup-label-1, my-eks-
nodegroup-label-2' \
    --taints 'addOrUpdateTaints=[{key=taint-new-key-1,value=taint-new-
value-1,effect=PREFER_NO_SCHEDULE}],removeTaints=[{key=taint-key-1,value=taint-
value-1,effect=NO_EXECUTE}]'
```

輸出：

```
{
    "update": {
        "id": "4a9c8c45-6ac7-3115-be71-d6412a2339b7",
        "status": "InProgress",
        "type": "ConfigUpdate",
        "params": [
            {
                "type": "LabelsToAdd",
```

```
                "value": "{\"my-eks-nodegroup-new-label-1\":\"new-value-1\",\"my-
eks-nodegroup-new-label-2\":\"new-value-2\"}"
            },
            {
                "type": "LabelsToRemove",
                "value": "[\"my-eks-nodegroup-label-1\",\"my-eks-nodegroup-
label-2\"]"
            },
            {
                "type": "TaintsToAdd",
                "value": "[{\"effect\":\"PREFER_NO_SCHEDULE\",\"value\":\"taint-new-
value-1\",\"key\":\"taint-new-key-1\"}]"
            },
            {
                "type": "TaintsToRemove",
                "value": "[{\"effect\":\"NO_EXECUTE\",\"value\":\"taint-value-1\",
\"key\":\"taint-key-1\"}]"
            }
        ],
        "createdAt": "2024-04-08T12:30:55.486000-04:00",
        "errors": []
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的更新受管節點群組。

範例 4：更新受管節點群組以更新 Amazon EKS 叢集 EKS 工作者節點的 scaling-config 和 update-config

下列update-nodegroup-config範例會更新受管節點群組，以更新 Amazon EKS 叢集的 EKS 工作者節點的 scaling-config 和 update-config。

```
aws eks update-nodegroup-config \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --scaling-config minSize=1,maxSize=5,desiredSize=2 \
    --update-config maxUnavailable=2
```

輸出：

```
{
    "update": {
        "id": "a977160f-59bf-3023-805d-c9826e460aea",
```

```
        "status": "InProgress",
        "type": "ConfigUpdate",
        "params": [
            {
                "type": "MinSize",
                "value": "1"
            },
            {
                "type": "MaxSize",
                "value": "5"
            },
            {
                "type": "DesiredSize",
                "value": "2"
            },
            {
                "type": "MaxUnavailable",
                "value": "2"
            }
        ],
        "createdAt": "2024-04-08T12:35:17.036000-04:00",
        "errors": []
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的更新受管節點群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateNodegroupConfig。

update-nodegroup-version

以下程式碼範例顯示如何使用 update-nodegroup-version。

AWS CLI

範例 1：更新 Amazon EKS 受管節點群組的 Kubernetes 版本或 AMI 版本

下列update-nodegroup-version範例會將 Amazon EKS 受管節點群組的 Kubernetes 版本或 AMI 版本更新為 Kubernetes 叢集的最新可用版本。

```
aws eks update-nodegroup-version \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
```

```
    --no-force
```

輸出：

```
{
    "update": {
        "id": "a94ebfc3-6bf8-307a-89e6-7dbaa36421f7",
        "status": "InProgress",
        "type": "VersionUpdate",
        "params": [
            {
                "type": "Version",
                "value": "1.26"
            },
            {
                "type": "ReleaseVersion",
                "value": "1.26.12-20240329"
            }
        ],
        "createdAt": "2024-04-08T13:16:00.724000-04:00",
        "errors": []
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的更新受管節點群組。

範例 2：更新 Amazon EKS 受管節點群組的 Kubernetes 版本或 AMI 版本

下列update-nodegroup-version範例會將 Amazon EKS 受管節點群組的 Kubernetes 版本或 AMI 版本更新為指定的 AMI 發行版本。

```
aws eks update-nodegroup-version \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --kubernetes-version '1.26' \
    --release-version '1.26.12-20240307' \
    --no-force
```

輸出：

```
{
    "update": {
```

```
        "id": "4db06fe1-088d-336b-bdcd-3fdb94995fb7",
        "status": "InProgress",
        "type": "VersionUpdate",
        "params": [
            {
                "type": "Version",
                "value": "1.26"
            },
            {
                "type": "ReleaseVersion",
                "value": "1.26.12-20240307"
            }
        ],
        "createdAt": "2024-04-08T13:13:58.595000-04:00",
        "errors": []
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的更新受管節點群組 - <https://docs.aws.amazon.com/eks/latest/userguide/update-managed-node-group.html>`。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateNodegroupVersion。

update-pod-identity-association

以下程式碼範例顯示如何使用 update-pod-identity-association。

AWS CLI

更新 EKS Pod 身分關聯

下列update-pod-identity-association範例會將關聯 ID 的關聯 IAM 角色從 變更為 arn:aws:iam::111122223333:role/my-role，以更新 EKS Pod 身分arn:aws:iam::111122223333:role/s3-role關聯a-9njjin9gfghecgocd。此 API 僅允許更新相關聯的 IAM 角色。

```
aws eks update-pod-identity-association \
    --cluster-name eks-customer \
    --association-id a-9njjin9gfghecgocd \
    --role-arn arn:aws:iam::111122223333:role/s3-role
```

輸出：

```
{
    "association": {
        "clusterName": "eks-customer",
        "namespace": "default",
        "serviceAccount": "default",
        "roleArn": "arn:aws:iam::111122223333:role/s3-role",
        "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-9njjin9gfghecgocd",
        "associationId": "a-9njjin9gfghecgocd",
        "tags": {
            "Key2": "value2",
            "Key1": "value1"
        },
        "createdAt": "2025-05-24T19:52:14.135000-05:00",
        "modifiedAt": "2025-05-25T21:01:53.120000-05:00"
    }
}
```

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的了解 EKS Pod Identity 如何授予 Pod 對 AWS 服務的存取權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePodIdentityAssociation。

使用 的 Elastic Beanstalk 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Elastic Beanstalk 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

abort-environment-update

以下程式碼範例顯示如何使用 abort-environment-update。

AWS CLI

中止部署

下列命令會中止名為 之環境的執行中應用程式版本部署my-env：

```
aws elasticbeanstalk abort-environment-update --environment-name my-env
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AbortEnvironmentUpdate。

check-dns-availability

以下程式碼範例顯示如何使用 check-dns-availability。

AWS CLI

檢查 CNAME 的可用性

下列命令會檢查子網域 的可用性my-cname.elasticbeanstalk.com：

```
aws elasticbeanstalk check-dns-availability --cname-prefix my-cname
```

輸出：

```
{
    "Available": true,
    "FullyQualifiedCNAME": "my-cname.elasticbeanstalk.com"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CheckDnsAvailability。

create-application-version

以下程式碼範例顯示如何使用 create-application-version。

AWS CLI

建立新的應用程式版本

下列命令會建立名為 "MyApp" 之應用程式的新版本 "v1"：

```
aws elasticbeanstalk create-application-version --application-name MyApp --
version-label v1 --description MyAppv1 --source-bundle S3Bucket="amzn-s3-demo-
bucket",S3Key="sample.war" --auto-create-application
```

由於自動建立應用程式選項，如果應用程式不存在，則auto-create-application。原始碼套件是存放在名為 "amzn-s3-demo-bucket" 的 s3 儲存貯體中的 .war 檔案，其中包含 Apache Tomcat 範例應用程式。

輸出：

```
{
  "ApplicationVersion": {
      "ApplicationName": "MyApp",
      "VersionLabel": "v1",
      "Description": "MyAppv1",
      "DateCreated": "2015-02-03T23:01:25.412Z",
      "DateUpdated": "2015-02-03T23:01:25.412Z",
      "SourceBundle": {
          "S3Bucket": "amzn-s3-demo-bucket",
          "S3Key": "sample.war"
      }
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateApplicationVersion。

create-application

以下程式碼範例顯示如何使用 create-application。

AWS CLI

建立新的應用程式

下列命令會建立新的應用程式，名為 "MyApp"：

```
aws elasticbeanstalk create-application --application-name MyApp --description "my
 application"
```

create-application 命令只會設定應用程式的名稱和描述。若要上傳應用程式的原始碼，請使用 建立應用程式的初始版本create-application-version。 create-application-

version也有一個auto-create-application選項，可讓您在一個步驟中建立應用程式和應用程式版本。

輸出：

```
{
  "Application": {
      "ApplicationName": "MyApp",
      "ConfigurationTemplates": [],
      "DateUpdated": "2015-02-12T18:32:21.181Z",
      "Description": "my application",
      "DateCreated": "2015-02-12T18:32:21.181Z"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateApplication。

create-configuration-template

以下程式碼範例顯示如何使用 create-configuration-template。

AWS CLI

建立組態範本

下列命令my-app-v1會從套用至 ID 為 的環境的設定建立名為 的組態範本e-rpqsewtp2j：

```
aws elasticbeanstalk create-configuration-template --application-name my-app --
template-name my-app-v1 --environment-id e-rpqsewtp2j
```

輸出：

```
{
    "ApplicationName": "my-app",
    "TemplateName": "my-app-v1",
    "DateCreated": "2015-08-12T18:40:39Z",
    "DateUpdated": "2015-08-12T18:40:39Z",
    "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateConfigurationTemplate。

create-environment

以下程式碼範例顯示如何使用 create-environment。

AWS CLI

為應用程式建立新環境

下列命令會為名為 "my-app" 的 java 應用程式版本 "v1" 建立新的環境：

```
aws elasticbeanstalk create-environment --application-name my-app --environment-
name my-env --cname-prefix my-app --version-label v1 --solution-stack-name "64bit
 Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8"
```

輸出：

```
{
  "ApplicationName": "my-app",
  "EnvironmentName": "my-env",
  "VersionLabel": "v1",
  "Status": "Launching",
  "EnvironmentId": "e-izqpassy4h",
  "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8",
  "CNAME": "my-app.elasticbeanstalk.com",
  "Health": "Grey",
  "Tier": {
      "Type": "Standard",
      "Name": "WebServer",
      "Version": " "
  },
  "DateUpdated": "2015-02-03T23:04:54.479Z",
  "DateCreated": "2015-02-03T23:04:54.479Z"
}
```

v1 是先前使用 create-application-version 上傳的應用程式版本標籤。

指定 JSON 檔案以定義環境組態選項

下列create-environment命令指定myoptions.json應使用名為 的 JSON 檔案來覆寫從解決方案堆疊或組態範本取得的值：

```
aws elasticbeanstalk create-environment --environment-name sample-env --application-
name sampleapp --option-settings file://myoptions.json
```

myoptions.json 是定義數個設定的 JSON 物件：

```
[
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "Interval",
    "Value": "15"
  },
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "Timeout",
    "Value": "8"
  },
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "HealthyThreshold",
    "Value": "2"
  },
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "UnhealthyThreshold",
    "Value": "3"
  }
]
```

如需詳細資訊，請參閱 AWS Elastic Beanstalk 開發人員指南中的選項值。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateEnvironment。

create-storage-location

以下程式碼範例顯示如何使用 create-storage-location。

AWS CLI

建立儲存位置

下列命令會在 Amazon S3 中建立儲存位置：

```
aws elasticbeanstalk create-storage-location
```

輸出：

```
{
    "S3Bucket": "elasticbeanstalk-us-west-2-0123456789012"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateStorageLocation。

delete-application-version

以下程式碼範例顯示如何使用 delete-application-version。

AWS CLI

刪除應用程式版本

下列命令會刪除名為 之應用程式的應用程式版本22a0-stage-150819_182129my-app：

```
aws elasticbeanstalk delete-application-version --version-label 22a0-
stage-150819_182129 --application-name my-app
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteApplicationVersion。

delete-application

以下程式碼範例顯示如何使用 delete-application。

AWS CLI

如欲刪除應用程式

下列命令會刪除名為 的應用程式my-app：

```
aws elasticbeanstalk delete-application --application-name my-app
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteApplication。

delete-configuration-template

以下程式碼範例顯示如何使用 delete-configuration-template。

AWS CLI

刪除組態範本

下列命令會刪除名為 之應用程式名為 my-template的組態範本my-app：

```
aws elasticbeanstalk delete-configuration-template --template-name my-template --
application-name my-app
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteConfigurationTemplate。

delete-environment-configuration

以下程式碼範例顯示如何使用 delete-environment-configuration。

AWS CLI

刪除草稿組態

下列命令會刪除名為 之環境的草稿組態my-env：

```
aws elasticbeanstalk delete-environment-configuration --environment-name my-env --
application-name my-app
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteEnvironmentConfiguration。

describe-application-versions

以下程式碼範例顯示如何使用 describe-application-versions。

AWS CLI

檢視應用程式版本的相關資訊

下列命令會擷取標示為 的應用程式版本相關資訊v2：

```
aws elasticbeanstalk describe-application-versions --application-name my-app --
version-label "v2"
```

輸出：

```
{
    "ApplicationVersions": [
        {
            "ApplicationName": "my-app",
            "VersionLabel": "v2",
            "Description": "update cover page",
            "DateCreated": "2015-07-23T01:32:26.079Z",
            "DateUpdated": "2015-07-23T01:32:26.079Z",
            "SourceBundle": {
                "S3Bucket": "elasticbeanstalk-us-west-2-015321684451",
                "S3Key": "my-app/5026-stage-150723_224258.war"
            }
        },
      {
          "ApplicationName": "my-app",
          "VersionLabel": "v1",
          "Description": "initial version",
          "DateCreated": "2015-07-23T22:26:10.816Z",
          "DateUpdated": "2015-07-23T22:26:10.816Z",
          "SourceBundle": {
              "S3Bucket": "elasticbeanstalk-us-west-2-015321684451",
              "S3Key": "my-app/5026-stage-150723_222618.war"
          }
      }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeApplicationVersions。

describe-applications

以下程式碼範例顯示如何使用 describe-applications。

AWS CLI

檢視應用程式清單

下列命令會擷取目前區域中應用程式的相關資訊：

```
aws elasticbeanstalk describe-applications
```

輸出：

```
{
    "Applications": [
        {
            "ApplicationName": "ruby",
            "ConfigurationTemplates": [],
            "DateUpdated": "2015-08-13T21:05:44.376Z",
            "Versions": [
                "Sample Application"
            ],
            "DateCreated": "2015-08-13T21:05:44.376Z"
        },
        {
            "ApplicationName": "pythonsample",
            "Description": "Application created from the EB CLI using \"eb init\"",
            "Versions": [
                "Sample Application"
            ],
            "DateCreated": "2015-08-13T19:05:43.637Z",
            "ConfigurationTemplates": [],
            "DateUpdated": "2015-08-13T19:05:43.637Z"
        },
        {
            "ApplicationName": "nodejs-example",
            "ConfigurationTemplates": [],
            "DateUpdated": "2015-08-06T17:50:02.486Z",
            "Versions": [
                "add elasticache",
                "First Release"
            ],
            "DateCreated": "2015-08-06T17:50:02.486Z"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeApplications。

describe-configuration-options

以下程式碼範例顯示如何使用 describe-configuration-options。

AWS CLI

檢視環境的組態選項

下列命令會擷取名為 之環境所有可用組態選項的描述my-env：

```
aws elasticbeanstalk describe-configuration-options --environment-name my-env --
application-name my-app
```

輸出（縮寫）：

```
{
    "Options": [
        {
            "Name": "JVMOptions",
            "UserDefined": false,
            "DefaultValue": "Xms=256m,Xmx=256m,XX:MaxPermSize=64m,JVM Options=",
            "ChangeSeverity": "RestartApplicationServer",
            "Namespace": "aws:cloudformation:template:parameter",
            "ValueType": "KeyValueList"
        },
        {
            "Name": "Interval",
            "UserDefined": false,
            "DefaultValue": "30",
            "ChangeSeverity": "NoInterruption",
            "Namespace": "aws:elb:healthcheck",
            "MaxValue": 300,
            "MinValue": 5,
            "ValueType": "Scalar"
        },
        ...
        {
            "Name": "LowerThreshold",
            "UserDefined": false,
            "DefaultValue": "2000000",
            "ChangeSeverity": "NoInterruption",
            "Namespace": "aws:autoscaling:trigger",
            "MinValue": 0,
            "ValueType": "Scalar"
        },
        {
            "Name": "ListenerEnabled",
            "UserDefined": false,
            "DefaultValue": "true",
            "ChangeSeverity": "Unknown",
            "Namespace": "aws:elb:listener",
```

```
            "ValueType": "Boolean"
        }
    ]
}
```

可用的組態選項因平台和組態版本而異。如需命名空間和支援選項的詳細資訊，請參閱 AWS Elastic Beanstalk 開發人員指南中的選項值。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeConfigurationOptions。

describe-configuration-settings

以下程式碼範例顯示如何使用 describe-configuration-settings。

AWS CLI

檢視環境的組態設定

下列命令會擷取名為 之環境的組態設定my-env：

```
aws elasticbeanstalk describe-configuration-settings --environment-name my-env --
application-name my-app
```

輸出（縮寫）：

```
{
    "ConfigurationSettings": [
        {
            "ApplicationName": "my-app",
            "EnvironmentName": "my-env",
            "Description": "Environment created from the EB CLI using \"eb create
\"",
            "DeploymentStatus": "deployed",
            "DateCreated": "2015-08-13T19:16:25Z",
            "OptionSettings": [
                {
                    "OptionName": "Availability Zones",
                    "ResourceName": "AWSEBAutoScalingGroup",
                    "Namespace": "aws:autoscaling:asg",
                    "Value": "Any"
                },
                {
```

```
                    "OptionName": "Cooldown",
                    "ResourceName": "AWSEBAutoScalingGroup",
                    "Namespace": "aws:autoscaling:asg",
                    "Value": "360"
                },
                ...
                {
                    "OptionName": "ConnectionDrainingTimeout",
                    "ResourceName": "AWSEBLoadBalancer",
                    "Namespace": "aws:elb:policies",
                    "Value": "20"
                },
                {
                    "OptionName": "ConnectionSettingIdleTimeout",
                    "ResourceName": "AWSEBLoadBalancer",
                    "Namespace": "aws:elb:policies",
                    "Value": "60"
                }
            ],
            "DateUpdated": "2015-08-13T23:30:07Z",
            "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8
 Java 8"
        }
    ]
}
```

如需命名空間和支援選項的詳細資訊，請參閱 AWS Elastic Beanstalk 開發人員指南中的選項值。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeConfigurationSettings。

describe-environment-health

以下程式碼範例顯示如何使用 `describe-environment-health`。

AWS CLI

檢視環境運作狀態

下列命令會擷取名為 之環境的整體運作狀態資訊`my-env`：

```
aws elasticbeanstalk describe-environment-health --environment-name my-env --
attribute-names All
```

輸出：

```
{
    "Status": "Ready",
    "EnvironmentName": "my-env",
    "Color": "Green",
    "ApplicationMetrics": {
        "Duration": 10,
        "Latency": {
            "P99": 0.004,
            "P75": 0.002,
            "P90": 0.003,
            "P95": 0.004,
            "P85": 0.003,
            "P10": 0.001,
            "P999": 0.004,
            "P50": 0.001
        },
        "RequestCount": 45,
        "StatusCodes": {
            "Status3xx": 0,
            "Status2xx": 45,
            "Status5xx": 0,
            "Status4xx": 0
        }
    },
    "RefreshedAt": "2015-08-20T21:09:18Z",
    "HealthStatus": "Ok",
    "InstancesHealth": {
        "Info": 0,
        "Ok": 1,
        "Unknown": 0,
        "Severe": 0,
        "Warning": 0,
        "Degraded": 0,
        "NoData": 0,
        "Pending": 0
    },
    "Causes": []
}
```

運作狀態資訊僅適用於已啟用增強型運作狀態報告的環境。如需詳細資訊，請參閱 AWS Elastic Beanstalk 開發人員指南中的增強型運作狀態報告和監控。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEnvironmentHealth。

describe-environment-resources

以下程式碼範例顯示如何使用 describe-environment-resources。

AWS CLI

檢視您環境中 AWS 資源的相關資訊

下列命令會擷取名為 之環境中資源的相關資訊my-env：

```
aws elasticbeanstalk describe-environment-resources --environment-name my-env
```

輸出：

```
{
    "EnvironmentResources": {
        "EnvironmentName": "my-env",
        "AutoScalingGroups": [
            {
                "Name": "awseb-e-qu3fyyjyjs-stack-AWSEBAutoScalingGroup-
QSB2ZO88SXZT"
            }
        ],
        "Triggers": [],
        "LoadBalancers": [
            {
                "Name": "awseb-e-q-AWSEBLoa-1EEPZ0K98BIF0"
            }
        ],
        "Queues": [],
        "Instances": [
            {
                "Id": "i-0c91c786"
            }
        ],
        "LaunchConfigurations": [
            {
                "Name": "awseb-e-qu3fyyjyjs-stack-
AWSEBAutoScalingLaunchConfiguration-1UUVQIBC96TQ2"
            }
```

```
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEnvironmentResources。

describe-environments

以下程式碼範例顯示如何使用 describe-environments。

AWS CLI

檢視環境的相關資訊

下列命令會擷取名為 的環境相關資訊my-env：

```
aws elasticbeanstalk describe-environments --environment-names my-env
```

輸出：

```
{
    "Environments": [
        {
            "ApplicationName": "my-app",
            "EnvironmentName": "my-env",
            "VersionLabel": "7f58-stage-150812_025409",
            "Status": "Ready",
            "EnvironmentId": "e-rpqsewtp2j",
            "EndpointURL": "awseb-e-w-AWSEBLoa-1483140XB0Q4L-109QXY8121.us-
west-2.elb.amazonaws.com",
            "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8
 Java 8",
            "CNAME": "my-env.elasticbeanstalk.com",
            "Health": "Green",
            "AbortableOperationInProgress": false,
            "Tier": {
                "Version": " ",
                "Type": "Standard",
                "Name": "WebServer"
            },
            "DateUpdated": "2015-08-12T18:16:55.019Z",
            "DateCreated": "2015-08-07T20:48:49.599Z"
```

```
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEnvironments。

describe-events

以下程式碼範例顯示如何使用 describe-events。

AWS CLI

檢視環境的事件

下列命令會擷取名為 之環境的事件my-env：

```
aws elasticbeanstalk describe-events --environment-name my-env
```

輸出（縮寫）：

```
{
    "Events": [
        {
            "ApplicationName": "my-app",
            "EnvironmentName": "my-env",
            "Message": "Environment health has transitioned from Info to Ok.",
            "EventDate": "2015-08-20T07:06:53.535Z",
            "Severity": "INFO"
        },
        {
            "ApplicationName": "my-app",
            "EnvironmentName": "my-env",
            "Severity": "INFO",
            "RequestId": "b7f3960b-4709-11e5-ba1e-07e16200da41",
            "Message": "Environment update completed successfully.",
            "EventDate": "2015-08-20T07:06:02.049Z"
        },
        ...
        {
            "ApplicationName": "my-app",
            "EnvironmentName": "my-env",
            "Severity": "INFO",
```

```
            "RequestId": "ca8dfbf6-41ef-11e5-988b-651aa638f46b",
            "Message": "Using elasticbeanstalk-us-west-2-012445113685 as Amazon S3
 storage bucket for environment data.",
            "EventDate": "2015-08-13T19:16:27.561Z"
        },
        {
            "ApplicationName": "my-app",
            "EnvironmentName": "my-env",
            "Severity": "INFO",
            "RequestId": "cdfba8f6-41ef-11e5-988b-65638f41aa6b",
            "Message": "createEnvironment is starting.",
            "EventDate": "2015-08-13T19:16:26.581Z"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEvents。

describe-instances-health

以下程式碼範例顯示如何使用 `describe-instances-health`。

AWS CLI

檢視環境運作狀態

下列命令會擷取名為 之環境中執行個體的運作狀態資訊`my-env`：

```
aws elasticbeanstalk describe-instances-health --environment-name my-env --
attribute-names All
```

輸出：

```
{
    "InstanceHealthList": [
        {
            "InstanceId": "i-08691cc7",
            "ApplicationMetrics": {
                "Duration": 10,
                "Latency": {
                    "P99": 0.006,
                    "P75": 0.002,
```

```
                    "P90": 0.004,
                    "P95": 0.005,
                    "P85": 0.003,
                    "P10": 0.0,
                    "P999": 0.006,
                    "P50": 0.001
                },
                "RequestCount": 48,
                "StatusCodes": {
                    "Status3xx": 0,
                    "Status2xx": 47,
                    "Status5xx": 0,
                    "Status4xx": 1
                }
            },
            "System": {
                "LoadAverage": [
                    0.0,
                    0.02,
                    0.05
                ],
                "CPUUtilization": {
                    "SoftIRQ": 0.1,
                    "IOWait": 0.2,
                    "System": 0.3,
                    "Idle": 97.8,
                    "User": 1.5,
                    "IRQ": 0.0,
                    "Nice": 0.1
                }
            },
            "Color": "Green",
            "HealthStatus": "Ok",
            "LaunchedAt": "2015-08-13T19:17:09Z",
            "Causes": []
        }
    ],
    "RefreshedAt": "2015-08-20T21:09:08Z"
}
```

運作狀態資訊僅適用於已啟用增強型運作狀態報告的環境。如需詳細資訊，請參閱 AWS Elastic Beanstalk 開發人員指南中的增強型運作狀態報告和監控。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInstancesHealth。

list-available-solution-stacks

以下程式碼範例顯示如何使用 list-available-solution-stacks。

AWS CLI

檢視解決方案堆疊

下列命令會列出所有目前可用平台組態以及您過去使用過的任何 解決方案堆疊：

```
aws elasticbeanstalk list-available-solution-stacks
```

輸出（縮寫）：

```
{
    "SolutionStacks": [
        "64bit Amazon Linux 2015.03 v2.0.0 running Node.js",
        "64bit Amazon Linux 2015.03 v2.0.0 running PHP 5.6",
        "64bit Amazon Linux 2015.03 v2.0.0 running PHP 5.5",
        "64bit Amazon Linux 2015.03 v2.0.0 running PHP 5.4",
        "64bit Amazon Linux 2015.03 v2.0.0 running Python 3.4",
        "64bit Amazon Linux 2015.03 v2.0.0 running Python 2.7",
        "64bit Amazon Linux 2015.03 v2.0.0 running Python",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.2 (Puma)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.2 (Passenger Standalone)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.1 (Puma)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.1 (Passenger Standalone)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.0 (Puma)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.0 (Passenger Standalone)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 1.9.3",
        "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8",
        "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 7 Java 7",
        "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 7 Java 6",
        "64bit Windows Server Core 2012 R2 running IIS 8.5",
        "64bit Windows Server 2012 R2 running IIS 8.5",
        "64bit Windows Server 2012 running IIS 8",
        "64bit Windows Server 2008 R2 running IIS 7.5",
        "64bit Amazon Linux 2015.03 v2.0.0 running Docker 1.6.2",
        "64bit Amazon Linux 2015.03 v2.0.0 running Multi-container Docker 1.6.2
 (Generic)",
        "64bit Debian jessie v2.0.0 running GlassFish 4.1 Java 8 (Preconfigured -
 Docker)",
        "64bit Debian jessie v2.0.0 running GlassFish 4.0 Java 7 (Preconfigured -
 Docker)",
```

```
        "64bit Debian jessie v2.0.0 running Go 1.4 (Preconfigured - Docker)",
        "64bit Debian jessie v2.0.0 running Go 1.3 (Preconfigured - Docker)",
        "64bit Debian jessie v2.0.0 running Python 3.4 (Preconfigured - Docker)",
    ],
    "SolutionStackDetails": [
        {
            "PermittedFileTypes": [
                "zip"
            ],
            "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Node.js"
        },
        ...
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAvailableSolutionStacks。

rebuild-environment

以下程式碼範例顯示如何使用 rebuild-environment。

AWS CLI

重建環境

下列命令會在名為 的環境中終止並重新建立資源my-env：

```
aws elasticbeanstalk rebuild-environment --environment-name my-env
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RebuildEnvironment。

request-environment-info

以下程式碼範例顯示如何使用 request-environment-info。

AWS CLI

請求結尾日誌

下列命令會從名為 的環境請求日誌my-env：

```
aws elasticbeanstalk request-environment-info --environment-name my-env --info-
type tail
```

請求日誌之後，請使用 retrieve-environment-info 擷取其位置。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RequestEnvironmentInfo。

`restart-app-server`

以下程式碼範例顯示如何使用 `restart-app-server`。

AWS CLI

重新啟動應用程式伺服器

下列命令會在名為 的環境中重新啟動所有執行個體上的應用程式伺服器my-env：

```
aws elasticbeanstalk restart-app-server --environment-name my-env
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestartAppServer。

`retrieve-environment-info`

以下程式碼範例顯示如何使用 `retrieve-environment-info`。

AWS CLI

擷取結尾日誌

下列命令會從名為 的環境擷取日誌的連結my-env：

```
aws elasticbeanstalk retrieve-environment-info --environment-name my-env --info-
type tail
```

輸出：

```
{
    "EnvironmentInfo": [
        {
            "SampleTimestamp": "2015-08-20T22:23:17.703Z",
            "Message": "https://elasticbeanstalk-us-
west-2-0123456789012.s3.amazonaws.com/resources/environments/
```

```
logs/tail/e-fyqyju3yjs/i-09c1c867/TailLogs-1440109397703.out?
AWSAccessKeyId=AKGPT4J56IAJ2EUBL5CQ&Expires=1440195891&Signature=n
%2BEalOV6A2HIOx4Rcfb7LT16bBM%3D",
            "InfoType": "tail",
            "Ec2InstanceId": "i-09c1c867"
        }
    ]
}
```

在瀏覽器中檢視連結。在擷取之前，必須使用 request-environment-info 請求日誌。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RetrieveEnvironmentInfo。

swap-environment-cnames

以下程式碼範例顯示如何使用 swap-environment-cnames。

AWS CLI

交換環境 CNAMES

下列命令會交換兩個環境的指派子網域：

```
aws elasticbeanstalk swap-environment-cnames --source-environment-name my-env-blue
 --destination-environment-name my-env-green
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SwapEnvironmentCnames。

terminate-environment

以下程式碼範例顯示如何使用 terminate-environment。

AWS CLI

終止環境

下列命令會終止名為 的 Elastic Beanstalk 環境my-env：

```
aws elasticbeanstalk terminate-environment --environment-name my-env
```

輸出：

```
{
    "ApplicationName": "my-app",
    "EnvironmentName": "my-env",
    "Status": "Terminating",
    "EnvironmentId": "e-fh2eravpns",
    "EndpointURL": "awseb-e-f-AWSEBLoa-1I9XUMP4-8492WNUP202574.us-
west-2.elb.amazonaws.com",
    "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java
 8",
    "CNAME": "my-env.elasticbeanstalk.com",
    "Health": "Grey",
    "AbortableOperationInProgress": false,
    "Tier": {
        "Version": " ",
        "Type": "Standard",
        "Name": "WebServer"
    },
    "DateUpdated": "2015-08-12T19:05:54.744Z",
    "DateCreated": "2015-08-12T18:52:53.622Z"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TerminateEnvironment。

update-application-version

以下程式碼範例顯示如何使用 update-application-version。

AWS CLI

變更應用程式版本的描述

下列命令會更新名為 之應用程式版本的描述22a0-stage-150819_185942：

```
aws elasticbeanstalk update-application-version --version-label 22a0-
stage-150819_185942 --application-name my-app --description "new description"
```

輸出：

```
{
    "ApplicationVersion": {
        "ApplicationName": "my-app",
        "VersionLabel": "22a0-stage-150819_185942",
```

```
        "Description": "new description",
        "DateCreated": "2015-08-19T18:59:17.646Z",
        "DateUpdated": "2015-08-20T22:53:28.871Z",
        "SourceBundle": {
            "S3Bucket": "elasticbeanstalk-us-west-2-0123456789012",
            "S3Key": "my-app/22a0-stage-150819_185942.war"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateApplicationVersion。

update-application

以下程式碼範例顯示如何使用 update-application。

AWS CLI

變更應用程式的描述

下列命令會更新名為 之應用程式的描述my-app：

```
aws elasticbeanstalk update-application --application-name my-app --description "my
 Elastic Beanstalk application"
```

輸出：

```
{
    "Application": {
        "ApplicationName": "my-app",
        "Description": "my Elastic Beanstalk application",
        "Versions": [
            "2fba-stage-150819_234450",
            "bf07-stage-150820_214945",
            "93f8",
            "fd7c-stage-150820_000431",
            "22a0-stage-150819_185942"
        ],
        "DateCreated": "2015-08-13T19:15:50.449Z",
        "ConfigurationTemplates": [],
        "DateUpdated": "2015-08-20T22:34:56.195Z"
    }
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateApplication。

update-configuration-template

以下程式碼範例顯示如何使用 update-configuration-template。

AWS CLI

更新組態範本

下列命令ConfigDocument會從名為 的已儲存組態範本中移除設定的 CloudWatch 自訂運作狀態指標組態my-template：

```
aws elasticbeanstalk update-configuration-template --template-
name my-template --application-name my-app --options-to-
remove Namespace=aws:elasticbeanstalk:healthreporting:system,OptionName=ConfigDocument
```

輸出：

```
{
    "ApplicationName": "my-app",
    "TemplateName": "my-template",
    "DateCreated": "2015-08-20T22:39:31Z",
    "DateUpdated": "2015-08-20T22:43:11Z",
    "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8"
}
```

如需命名空間和支援選項的詳細資訊，請參閱 AWS Elastic Beanstalk 開發人員指南中的選項值。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateConfigurationTemplate。

update-environment

以下程式碼範例顯示如何使用 update-environment。

AWS CLI

將環境更新為新版本

下列命令會將名為 "my-env" 的環境更新為其所屬應用程式的版本 "v2"：

```
aws elasticbeanstalk update-environment --environment-name my-env --version-label v2
```

此命令要求「my-env」環境已存在，並屬於具有有效應用程式版本且標籤為「v2」的應用程式。

輸出：

```
{
  "ApplicationName": "my-app",
  "EnvironmentName": "my-env",
  "VersionLabel": "v2",
  "Status": "Updating",
  "EnvironmentId": "e-szqipays4h",
  "EndpointURL": "awseb-e-i-AWSEBLoa-1RDLX6TC9VUAO-0123456789.us-
west-2.elb.amazonaws.com",
  "SolutionStackName": "64bit Amazon Linux running Tomcat 7",
  "CNAME": "my-env.elasticbeanstalk.com",
  "Health": "Grey",
  "Tier": {
      "Version": " ",
      "Type": "Standard",
      "Name": "WebServer"
  },
  "DateUpdated": "2015-02-03T23:12:29.119Z",
  "DateCreated": "2015-02-03T23:04:54.453Z"
}
```

設定環境變數

下列命令會將 "my-env" 環境中的 "PARAM1" 變數值設定為 "ParamValue"：

```
aws elasticbeanstalk update-environment --environment-name my-env --option-
settings Namespace=aws:elasticbeanstalk:application:environment,OptionName=PARAM1,Value=Para
```

除了 變數的名稱和值之外，`option-settings` 參數還會採用命名空間。除了環境變數之外，Elastic Beanstalk 還支援多個 選項的命名空間。

從檔案設定選項設定

下列命令會從 檔案設定 `aws:elb:loadbalancer` 命名空間中的數個選項：

```
aws elasticbeanstalk update-environment --environment-name my-env --option-
settings file://options.json
```

options.json 是定義數個設定的 JSON 物件：

```
[
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "Interval",
    "Value": "15"
  },
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "Timeout",
    "Value": "8"
  },
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "HealthyThreshold",
    "Value": "2"
  },
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "UnhealthyThreshold",
    "Value": "3"
  }
]
```

輸出：

```
{
    "ApplicationName": "my-app",
    "EnvironmentName": "my-env",
    "VersionLabel": "7f58-stage-150812_025409",
    "Status": "Updating",
    "EnvironmentId": "e-wtp2rpqsej",
    "EndpointURL": "awseb-e-w-AWSEBLoa-14XB83101Q4L-104QXY80921.sa-
east-1.elb.amazonaws.com",
    "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java
 8",
    "CNAME": "my-env.elasticbeanstalk.com",
    "Health": "Grey",
    "AbortableOperationInProgress": true,
    "Tier": {
        "Version": " ",
        "Type": "Standard",
```

```
        "Name": "WebServer"
    },
    "DateUpdated": "2015-08-12T18:15:23.804Z",
    "DateCreated": "2015-08-07T20:48:49.599Z"
}
```

如需命名空間和支援選項的詳細資訊，請參閱 AWS Elastic Beanstalk 開發人員指南中的選項值。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateEnvironment。

validate-configuration-settings

以下程式碼範例顯示如何使用 validate-configuration-settings。

AWS CLI

驗證組態設定

下列命令會驗證 CloudWatch 自訂指標組態文件：

```
aws elasticbeanstalk validate-configuration-settings --application-name my-app --
environment-name my-env --option-settings file://options.json
```

options.json 是一種 JSON 文件，其中包含一或多個要驗證的組態設定：

```
[
    {
        "Namespace": "aws:elasticbeanstalk:healthreporting:system",
        "OptionName": "ConfigDocument",
        "Value": "{\"CloudWatchMetrics\": {\"Environment\":
 {\"ApplicationLatencyP99.9\": null,\"InstancesSevere\": 60,
\"ApplicationLatencyP90\": 60,\"ApplicationLatencyP99\": null,
\"ApplicationLatencyP95\": 60,\"InstancesUnknown\": 60,\"ApplicationLatencyP85\":
 60,\"InstancesInfo\": null,\"ApplicationRequests2xx\": null,\"InstancesDegraded
\": null,\"InstancesWarning\": 60,\"ApplicationLatencyP50\": 60,
\"ApplicationRequestsTotal\": null,\"InstancesNoData\": null,\"InstancesPending
\": 60,\"ApplicationLatencyP10\": null,\"ApplicationRequests5xx\": null,
\"ApplicationLatencyP75\": null,\"InstancesOk\": 60,\"ApplicationRequests3xx\":
 null,\"ApplicationRequests4xx\": null},\"Instance\": {\"ApplicationLatencyP99.9\":
 null,\"ApplicationLatencyP90\": 60,\"ApplicationLatencyP99\": null,
\"ApplicationLatencyP95\": null,\"ApplicationLatencyP85\": null,\"CPUUser\": 60,
\"ApplicationRequests2xx\": null,\"CPUIdle\": null,\"ApplicationLatencyP50\":
 null,\"ApplicationRequestsTotal\": 60,\"RootFilesystemUtil\": null,
```

```
\"LoadAverage1min\": null,\"CPUIrq\": null,\"CPUNice\": 60,\"CPUIowait\": 60,
\"ApplicationLatencyP10\": null,\"LoadAverage5min\": null,\"ApplicationRequests5xx
\": null,\"ApplicationLatencyP75\": 60,\"CPUSystem\": 60,\"ApplicationRequests3xx\":
 60,\"ApplicationRequests4xx\": null,\"InstanceHealth\": null,\"CPUSoftirq\": 60}},
\"Version\": 1}"
    }
]
```

如果您指定的選項對指定的環境有效，Elastic Beanstalk 會傳回空的訊息陣列：

```
{
    "Messages": []
}
```

如果驗證失敗，回應將包含錯誤的相關資訊：

```
{
    "Messages": [
        {
            "OptionName": "ConfigDocumet",
            "Message": "Invalid option specification (Namespace:
 'aws:elasticbeanstalk:healthreporting:system', OptionName: 'ConfigDocumet'):
 Unknown configuration setting.",
            "Namespace": "aws:elasticbeanstalk:healthreporting:system",
            "Severity": "error"
        }
    ]
}
```

如需命名空間和支援選項的詳細資訊，請參閱 AWS Elastic Beanstalk 開發人員指南中的選項值。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ValidateConfigurationSettings。

Elastic Load Balancing - 使用 的第 1 版範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Elastic Load Balancing - 第 1 版，來執行動作並實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

add-tags

以下程式碼範例顯示如何使用 add-tags。

AWS CLI

將標籤新增至負載平衡器

此範例會將標籤新增至指定的負載平衡器。

命令：

```
aws elb add-tags --load-balancer-name my-load-balancer --
tags "Key=project,Value=lima" "Key=department,Value=digital-media"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddTags。

apply-security-groups-to-load-balancer

以下程式碼範例顯示如何使用 apply-security-groups-to-load-balancer。

AWS CLI

將安全群組與 VPC 中的負載平衡器建立關聯

此範例會將安全群組與 VPC 中指定的負載平衡器建立關聯。

命令：

```
aws elb apply-security-groups-to-load-balancer --load-balancer-name my-load-balancer
 --security-groups sg-fc448899
```

輸出：

```
{
  "SecurityGroups": [
```

```
        "sg-fc448899"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ApplySecurityGroupsToLoadBalancer。

attach-load-balancer-to-subnets

以下程式碼範例顯示如何使用 attach-load-balancer-to-subnets。

AWS CLI

將子網路連接至負載平衡器

此範例會將指定的子網路新增至指定負載平衡器的一組已設定子網路。

命令：

```
aws elb attach-load-balancer-to-subnets --load-balancer-name my-load-balancer --
subnets subnet-0ecac448
```

輸出：

```
{
   "Subnets": [
       "subnet-15aaab61",
       "subnet-0ecac448"
   ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachLoadBalancerToSubnets。

configure-health-check

以下程式碼範例顯示如何使用 configure-health-check。

AWS CLI

指定後端 EC2 執行個體的運作狀態檢查設定

此範例會指定用來評估後端 EC2 執行個體運作狀態的運作狀態檢查設定。

命令：

```
aws elb configure-health-check --load-balancer-name my-load-balancer --health-
check Target=HTTP:80/
png,Interval=30,UnhealthyThreshold=2,HealthyThreshold=2,Timeout=3
```

輸出：

```
{
   "HealthCheck": {
       "HealthyThreshold": 2,
       "Interval": 30,
       "Target": "HTTP:80/png",
       "Timeout": 3,
       "UnhealthyThreshold": 2
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ConfigureHealthCheck。

create-app-cookie-stickiness-policy

以下程式碼範例顯示如何使用 create-app-cookie-stickiness-policy。

AWS CLI

為您的 HTTPS 負載平衡器產生黏性政策

此範例會產生黏性政策，遵循應用程式產生之 Cookie 的黏性工作階段生命週期。

命令：

```
aws elb create-app-cookie-stickiness-policy --load-balancer-name my-load-balancer --
policy-name my-app-cookie-policy --cookie-name my-app-cookie
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAppCookieStickinessPolicy。

create-lb-cookie-stickiness-policy

以下程式碼範例顯示如何使用 create-lb-cookie-stickiness-policy。

AWS CLI

為您的 HTTPS 負載平衡器產生以持續時間為基礎的黏性政策

此範例會產生黏性政策，其中包含由指定過期期間控制的黏性工作階段生命週期。

命令：

```
aws elb create-lb-cookie-stickiness-policy --load-balancer-name my-load-balancer --
policy-name my-duration-cookie-policy --cookie-expiration-period 60
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLbCookieStickinessPolicy。

create-load-balancer-listeners

以下程式碼範例顯示如何使用 create-load-balancer-listeners。

AWS CLI

為負載平衡器建立 HTTP 接聽程式

此範例會使用 HTTP 通訊協定，在連接埠 80 為您的負載平衡器建立接聽程式。

命令：

```
aws elb create-load-balancer-listeners --load-balancer-name my-load-balancer --
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"
```

為負載平衡器建立 HTTPS 接聽程式

此範例會使用 HTTPS 通訊協定，在連接埠 443 為您的負載平衡器建立接聽程式。

命令：

```
aws elb create-load-balancer-listeners --load-balancer-name my-load-balancer --
listeners "Protocol=HTTPS,LoadBalancerPort=443,InstanceProtocol=HTTP,InstancePort=80"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLoadBalancerListeners。

create-load-balancer-policy

以下程式碼範例顯示如何使用 create-load-balancer-policy。

AWS CLI

建立在負載平衡器上啟用 Proxy Protocol 的政策

此範例會建立在指定的負載平衡器上啟用 Proxy Protocol 的政策。

命令：

```
aws elb create-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-ProxyProtocol-policy --policy-type-name ProxyProtocolPolicyType --policy-attributes AttributeName=ProxyProtocol,AttributeValue=true
```

使用建議的安全政策建立 SSL 溝通政策

此範例會使用建議的安全政策，為指定的 HTTPS 負載平衡器建立 SSL 溝通政策。

命令：

```
aws elb create-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-SSLNegotiation-policy --policy-type-name SSLNegotiationPolicyType --policy-attributes AttributeName=Reference-Security-Policy,AttributeValue=ELBSecurityPolicy-2015-03
```

使用自訂安全政策建立 SSL 溝通政策

此範例透過啟用通訊協定和密碼，使用自訂安全政策為您的 HTTPS 負載平衡器建立 SSL 溝通政策。

命令：

```
aws elb create-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-SSLNegotiation-policy --policy-type-name SSLNegotiationPolicyType --policy-attributes AttributeName=Protocol-SSLv3,AttributeValue=true AttributeName=Protocol-TLSv1.1,AttributeValue=true AttributeName=DHE-RSA-AES256-SHA256,AttributeValue=true AttributeName=Server-Defined-Cipher-Order,AttributeValue=true
```

建立公有金鑰政策

此範例會建立公有金鑰政策。

命令：

```
aws elb create-load-balancer-policy --load-balancer-name my-load-balancer --
policy-name my-PublicKey-policy --policy-type-name PublicKeyPolicyType --policy-
attributes AttributeName=PublicKey,AttributeValue=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQ
+dS74kj//c6x7ROtusUaeQCTgIUkayttRDWchuqo1pHC1u
+n5xxXnBBe2ejbb2WRsKIQ5rXEeixsjFpFsojpSQKkzhVGI6mJVZBJDVKSHmswnwLBdofLhzvllpovBPTHe
+o4haAWvDBALJU0pkSI1FecPHcs2hwxf14zHoXy1e2k36A64nXW43wtfx5qcVSIxtCEOjnYRg7RPvybaGfQ
+v6Iaxb/+7J5kEvZhTFQId+bSiJImF1FSUT1W1xwzBZPUbcUkkXDj45vC2s3Z8E
+Lk7a3uZhvsQHLZnrfuWjBWGWvZ/MhZYgEXAMPLE
```

建立後端伺服器身分驗證政策

此範例會建立後端伺服器身分驗證政策，以使用公有金鑰政策在您的後端執行個體上啟用身分驗證。

命令：

```
aws elb create-load-balancer-policy --load-balancer-name my-
load-balancer --policy-name my-authentication-policy --policy-
type-name BackendServerAuthenticationPolicyType --policy-
attributes AttributeName=PublicKeyPolicyName,AttributeValue=my-PublicKey-policy
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLoadBalancerPolicy。

create-load-balancer

以下程式碼範例顯示如何使用 create-load-balancer。

AWS CLI

建立 HTTP 負載平衡器

此範例會在 VPC 中建立具有 HTTP 接聽程式的負載平衡器。

命令：

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"
 --subnets subnet-15aaab61 --security-groups sg-a61988c3
```

輸出：

```
{
```

```
    "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com"
}
```

此範例會在 EC2-Classic 中建立具有 HTTP 接聽程式的負載平衡器。

命令：

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"
 --availability-zones us-west-2a us-west-2b
```

輸出：

```
{
    "DNSName": "my-load-balancer-123456789.us-west-2.elb.amazonaws.com"
}
```

建立 HTTPS 負載平衡器

此範例會在 VPC 中建立具有 HTTPS 接聽程式的負載平衡器。

命令：

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80" "Protoco
certificate/my-server-cert" --subnets subnet-15aaab61 --security-groups sg-a61988c3
```

輸出：

```
{
    "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com"
}
```

此範例會在 EC2-Classic 中使用 HTTPS 接聽程式建立負載平衡器。

命令：

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80" "Protoco
certificate/my-server-cert" --availability-zones us-west-2a us-west-2b
```

輸出：

```
{
    "DNSName": "my-load-balancer-123456789.us-west-2.elb.amazonaws.com"
}
```

建立內部負載平衡器

此範例會在 VPC 中建立具有 HTTP 接聽程式的內部負載平衡器。

命令：

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"
 --scheme internal --subnets subnet-a85db0df --security-groups sg-a61988c3
```

輸出：

```
{
    "DNSName": "internal-my-load-balancer-123456789.us-west-2.elb.amazonaws.com"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLoadBalancer。

delete-load-balancer-listeners

以下程式碼範例顯示如何使用 delete-load-balancer-listeners。

AWS CLI

從負載平衡器刪除接聽程式

此範例會從指定的負載平衡器刪除指定連接埠的接聽程式。

命令：

```
aws elb delete-load-balancer-listeners --load-balancer-name my-load-balancer --load-
balancer-ports 80
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLoadBalancerListeners。

delete-load-balancer-policy

以下程式碼範例顯示如何使用 delete-load-balancer-policy。

AWS CLI

從負載平衡器刪除政策

此範例會從指定的負載平衡器刪除指定的政策。政策不得在任何接聽程式上啟用。

命令：

```
aws elb delete-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-duration-cookie-policy
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLoadBalancerPolicy。

delete-load-balancer

以下程式碼範例顯示如何使用 delete-load-balancer。

AWS CLI

刪除負載平衡器

此範例會刪除指定的負載平衡器。

命令：

```
aws elb delete-load-balancer --load-balancer-name my-load-balancer
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLoadBalancer。

deregister-instances-from-load-balancer

以下程式碼範例顯示如何使用 deregister-instances-from-load-balancer。

AWS CLI

從負載平衡器取消註冊執行個體

此範例會從指定的負載平衡器取消註冊指定的執行個體。

命令：

```
aws elb deregister-instances-from-load-balancer --load-balancer-name my-load-balancer --instances i-d6f6fae3
```

輸出：

```
{
    "Instances": [
        {
            "InstanceId": "i-207d9717"
        },
        {
            "InstanceId": "i-afefb49b"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterInstancesFromLoadBalancer。

describe-account-limits

以下程式碼範例顯示如何使用 describe-account-limits。

AWS CLI

描述您的 Classic Load Balancer 限制

下列describe-account-limits範例顯示您 AWS 帳戶 Classic Load Balancer 限制的詳細資訊。

```
aws elb describe-account-limits
```

輸出：

```
{
    "Limits": [
        {
            "Name": "classic-load-balancers",
            "Max": "20"
```

```
        },
        {
            "Name": "classic-listeners",
            "Max": "100"
        },
        {
            "Name": "classic-registered-instances",
            "Max": "1000"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAccountLimits。

describe-instance-health

以下程式碼範例顯示如何使用 describe-instance-health。

AWS CLI

描述負載平衡器執行個體的運作狀態

此範例說明指定負載平衡器之執行個體的運作狀態。

命令：

```
aws elb describe-instance-health --load-balancer-name my-load-balancer
```

輸出：

```
{
  "InstanceStates": [
      {
          "InstanceId": "i-207d9717",
          "ReasonCode": "N/A",
          "State": "InService",
          "Description": "N/A"
      },
      {
          "InstanceId": "i-afefb49b",
          "ReasonCode": "N/A",
          "State": "InService",
          "Description": "N/A"
```

```
        }
    ]
}
```

描述負載平衡器執行個體的運作狀態

此範例說明指定負載平衡器之指定執行個體的運作狀態。

命令：

```
aws elb describe-instance-health --load-balancer-name my-load-balancer --
instances i-7299c809
```

以下是正在註冊之執行個體的範例回應。

輸出：

```
{
  "InstanceStates": [
      {
          "InstanceId": "i-7299c809",
          "ReasonCode": "ELB",
          "State": "OutOfService",
          "Description": "Instance registration is still in progress."
    }
  ]
}
```

以下是狀況不良執行個體的範例回應。

輸出：

```
{
  "InstanceStates": [
      {
          "InstanceId": "i-7299c809",
          "ReasonCode": "Instance",
          "State": "OutOfService",
          "Description": "Instance has failed at least the UnhealthyThreshold number
 of health checks consecutively."
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInstanceHealth。

describe-load-balancer-attributes

以下程式碼範例顯示如何使用 describe-load-balancer-attributes。

AWS CLI

描述負載平衡器的屬性

此範例說明指定負載平衡器的屬性。

命令：

```
aws elb describe-load-balancer-attributes --load-balancer-name my-load-balancer
```

輸出：

```
{
  "LoadBalancerAttributes": {
      "ConnectionDraining": {
          "Enabled": false,
          "Timeout": 300
      },
      "CrossZoneLoadBalancing": {
          "Enabled": true
      },
      "ConnectionSettings": {
          "IdleTimeout": 30
      },
      "AccessLog": {
          "Enabled": false
    }
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLoadBalancerAttributes。

describe-load-balancer-policies

以下程式碼範例顯示如何使用 describe-load-balancer-policies。

AWS CLI

描述與負載平衡器相關聯的所有政策

此範例說明與指定負載平衡器相關聯的所有政策。

命令：

```
aws elb describe-load-balancer-policies --load-balancer-name my-load-balancer
```

輸出：

```
{
  "PolicyDescriptions": [
    {
      "PolicyAttributeDescriptions": [
        {
          "AttributeName": "ProxyProtocol",
          "AttributeValue": "true"
        }
      ],
      "PolicyName": "my-ProxyProtocol-policy",
      "PolicyTypeName": "ProxyProtocolPolicyType"
    },
    {
        "PolicyAttributeDescriptions": [
            {
                "AttributeName": "CookieName",
                "AttributeValue": "my-app-cookie"
            }
        ],
        "PolicyName": "my-app-cookie-policy",
        "PolicyTypeName": "AppCookieStickinessPolicyType"
    },
    {
      "PolicyAttributeDescriptions": [
        {
          "AttributeName": "CookieExpirationPeriod",
          "AttributeValue": "60"
        }
      ],
      "PolicyName": "my-duration-cookie-policy",
      "PolicyTypeName": "LBCookieStickinessPolicyType"
    },
```

```
    .
    .
    .
  ]
}
```

描述與負載平衡器相關聯的特定政策

此範例說明與指定負載平衡器相關聯的指定政策。

命令：

```
aws elb describe-load-balancer-policies --load-balancer-name my-load-balancer --
policy-name my-authentication-policy
```

輸出：

```
{
  "PolicyDescriptions": [
      {
          "PolicyAttributeDescriptions": [
              {
                  "AttributeName": "PublicKeyPolicyName",
                  "AttributeValue": "my-PublicKey-policy"
              }
          ],
          "PolicyName": "my-authentication-policy",
          "PolicyTypeName": "BackendServerAuthenticationPolicyType"
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLoadBalancerPolicies。

describe-load-balancer-policy-types

以下程式碼範例顯示如何使用 describe-load-balancer-policy-types。

AWS CLI

描述 Elastic Load Balancing 定義的負載平衡器政策類型

此範例說明您可以用來為負載平衡器建立政策組態的負載平衡器政策類型。

命令：

```
aws elb describe-load-balancer-policy-types
```

輸出：

```
{
  "PolicyTypeDescriptions": [
      {
          "PolicyAttributeTypeDescriptions": [
              {
                  "Cardinality": "ONE",
                  "AttributeName": "ProxyProtocol",
                  "AttributeType": "Boolean"
              }
          ],
          "PolicyTypeName": "ProxyProtocolPolicyType",
          "Description": "Policy that controls whether to include the IP address and
 port of the originating request for TCP messages. This policy operates on TCP/SSL
 listeners only"
      },
      {
          "PolicyAttributeTypeDescriptions": [
              {
                  "Cardinality": "ONE",
                  "AttributeName": "PublicKey",
                  "AttributeType": "String"
              }
          ],
          "PolicyTypeName": "PublicKeyPolicyType",
          "Description": "Policy containing a list of public keys to
 accept when authenticating the back-end server(s). This policy cannot be
 applied directly to back-end servers or listeners but must be part of a
 BackendServerAuthenticationPolicyType."
      },
      {
          "PolicyAttributeTypeDescriptions": [
              {
                  "Cardinality": "ONE",
                  "AttributeName": "CookieName",
                  "AttributeType": "String"
              }
          ],
```

```
        "PolicyTypeName": "AppCookieStickinessPolicyType",
        "Description": "Stickiness policy with session lifetimes controlled by the
lifetime of the application-generated cookie. This policy can be associated only
with HTTP/HTTPS listeners."
    },
    {
        "PolicyAttributeTypeDescriptions": [
            {
                "Cardinality": "ZERO_OR_ONE",
                "AttributeName": "CookieExpirationPeriod",
                "AttributeType": "Long"
            }
        ],
        "PolicyTypeName": "LBCookieStickinessPolicyType",
        "Description": "Stickiness policy with session lifetimes controlled by
the browser (user-agent) or a specified expiration period. This policy can be
associated only with HTTP/HTTPS listeners."
    },
    {
        "PolicyAttributeTypeDescriptions": [
            .
            .
            .
        ],
        "PolicyTypeName": "SSLNegotiationPolicyType",
        "Description": "Listener policy that defines the ciphers and protocols
that will be accepted by the load balancer. This policy can be associated only with
HTTPS/SSL listeners."
    },
    {
        "PolicyAttributeTypeDescriptions": [
            {
                "Cardinality": "ONE_OR_MORE",
                "AttributeName": "PublicKeyPolicyName",
                "AttributeType": "PolicyName"
            }
        ],
        "PolicyTypeName": "BackendServerAuthenticationPolicyType",
        "Description": "Policy that controls authentication to back-end server(s)
and contains one or more policies, such as an instance of a PublicKeyPolicyType.
This policy can be associated only with back-end servers that are using HTTPS/SSL."
    }
 ]
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLoadBalancerPolicyTypes。

describe-load-balancers

以下程式碼範例顯示如何使用 describe-load-balancers。

AWS CLI

描述您的負載平衡器

此範例說明所有負載平衡器。

命令：

```
aws elb describe-load-balancers
```

描述其中一個負載平衡器

此範例說明指定的負載平衡器。

命令：

```
aws elb describe-load-balancers --load-balancer-name my-load-balancer
```

下列範例回應適用於 VPC 中的 HTTPS 負載平衡器。

輸出：

```
{
  "LoadBalancerDescriptions": [
    {
      "Subnets": [
          "subnet-15aaab61"
      ],
      "CanonicalHostedZoneNameID": "Z3DZXE0EXAMPLE",
      "CanonicalHostedZoneName": "my-load-balancer-1234567890.us-
west-2.elb.amazonaws.com",
      "ListenerDescriptions": [
          {
              "Listener": {
                  "InstancePort": 80,
```

```
                    "LoadBalancerPort": 80,
                    "Protocol": "HTTP",
                    "InstanceProtocol": "HTTP"
                },
                "PolicyNames": []
            },
            {
                "Listener": {
                    "InstancePort": 443,
                    "SSLCertificateId": "arn:aws:iam::123456789012:server-certificate/
my-server-cert",
                    "LoadBalancerPort": 443,
                    "Protocol": "HTTPS",
                    "InstanceProtocol": "HTTPS"
                },
                "PolicyNames": [
                    "ELBSecurityPolicy-2015-03"
                ]
            }
        ],
        "HealthCheck": {
            "HealthyThreshold": 2,
            "Interval": 30,
            "Target": "HTTP:80/png",
            "Timeout": 3,
            "UnhealthyThreshold": 2
        },
        "VPCId": "vpc-a01106c2",
        "BackendServerDescriptions": [
            {
                "InstancePort": 80,
                "PolicyNames": [
                    "my-ProxyProtocol-policy"
                ]
            }
        ],
        "Instances": [
            {
                "InstanceId": "i-207d9717"
            },
            {
                "InstanceId": "i-afefb49b"
            }
        ],
```

```
      "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com",
      "SecurityGroups": [
          "sg-a61988c3"
      ],
      "Policies": {
          "LBCookieStickinessPolicies": [
              {
                  "PolicyName": "my-duration-cookie-policy",
                  "CookieExpirationPeriod": 60
              }
          ],
          "AppCookieStickinessPolicies": [],
          "OtherPolicies": [
              "my-PublicKey-policy",
              "my-authentication-policy",
              "my-SSLNegotiation-policy",
              "my-ProxyProtocol-policy",
              "ELBSecurityPolicy-2015-03"
          ]
      },
      "LoadBalancerName": "my-load-balancer",
      "CreatedTime": "2015-03-19T03:24:02.650Z",
      "AvailabilityZones": [
          "us-west-2a"
      ],
      "Scheme": "internet-facing",
      "SourceSecurityGroup": {
          "OwnerAlias": "123456789012",
          "GroupName": "my-elb-sg"
      }
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLoadBalancers。

describe-tags

以下程式碼範例顯示如何使用 describe-tags。

AWS CLI

描述指派給負載平衡器的標籤

此範例說明指派給指定負載平衡器的標籤。

命令：

```
aws elb describe-tags --load-balancer-name my-load-balancer
```

輸出：

```
{
  "TagDescriptions": [
      {
          "Tags": [
              {
                  "Value": "lima",
                  "Key": "project"
              },
              {
                  "Value": "digital-media",
                  "Key": "department"
              }
          ],
          "LoadBalancerName": "my-load-balancer"
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTags。

detach-load-balancer-from-subnets

以下程式碼範例顯示如何使用 detach-load-balancer-from-subnets。

AWS CLI

從子網路分離負載平衡器

此範例會將指定的負載平衡器從指定的子網路分離。

命令：

```
aws elb detach-load-balancer-from-subnets --load-balancer-name my-load-balancer --
subnets subnet-0ecac448
```

輸出：

```
{
   "Subnets": [
       "subnet-15aaab61"
   ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetachLoadBalancerFromSubnets。

disable-availability-zones-for-load-balancer

以下程式碼範例顯示如何使用 disable-availability-zones-for-load-balancer。

AWS CLI

停用負載平衡器的可用區域

此範例會從指定負載平衡器的一組可用區域中移除指定的可用區域。

命令：

```
aws elb disable-availability-zones-for-load-balancer --load-balancer-name my-load-balancer --availability-zones us-west-2a
```

輸出：

```
{
    "AvailabilityZones": [
        "us-west-2b"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableAvailabilityZonesForLoadBalancer。

enable-availability-zones-for-load-balancer

以下程式碼範例顯示如何使用 enable-availability-zones-for-load-balancer。

AWS CLI

啟用負載平衡器的可用區域

此範例會將指定的可用區域新增至指定的負載平衡器。

命令：

```
aws elb enable-availability-zones-for-load-balancer --load-balancer-name my-load-balancer --availability-zones us-west-2b
```

輸出：

```
{
    "AvailabilityZones": [
        "us-west-2a",
        "us-west-2b"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableAvailabilityZonesForLoadBalancer。

modify-load-balancer-attributes

以下程式碼範例顯示如何使用 modify-load-balancer-attributes。

AWS CLI

修改負載平衡器的屬性

此範例會修改指定負載平衡器的 CrossZoneLoadBalancing 屬性。

命令：

```
aws elb modify-load-balancer-attributes --load-balancer-name my-load-balancer --load-balancer-attributes "{\"CrossZoneLoadBalancing\":{\"Enabled\":true}}"
```

輸出：

```
{
```

```
    "LoadBalancerAttributes": {
        "CrossZoneLoadBalancing": {
            "Enabled": true
        }
    },
    "LoadBalancerName": "my-load-balancer"
}
```

此範例會修改指定負載平衡器的 ConnectionDraining 屬性。

命令：

```
aws elb modify-load-balancer-attributes --load-balancer-name my-load-balancer
 --load-balancer-attributes "{\"ConnectionDraining\":{\"Enabled\":true,\"Timeout
\":300}}"
```

輸出：

```
{
    "LoadBalancerAttributes": {
        "ConnectionDraining": {
            "Enabled": true,
            "Timeout": 300
        }
    },
    "LoadBalancerName": "my-load-balancer"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyLoadBalancerAttributes。

register-instances-with-load-balancer

以下程式碼範例顯示如何使用 register-instances-with-load-balancer。

AWS CLI

向負載平衡器註冊執行個體

此範例會向指定的負載平衡器註冊指定的執行個體。

命令：

```
aws elb register-instances-with-load-balancer --load-balancer-name my-load-balancer
 --instances i-d6f6fae3
```

輸出：

```
{
   "Instances": [
       {
           "InstanceId": "i-d6f6fae3"
       },
       {
           "InstanceId": "i-207d9717"
       },
       {
           "InstanceId": "i-afefb49b"
       }
   ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterInstancesWithLoadBalancer。

remove-tags

以下程式碼範例顯示如何使用 remove-tags。

AWS CLI

從負載平衡器移除標籤

此範例會從指定的負載平衡器移除標籤。

命令：

```
aws elb remove-tags --load-balancer-name my-load-balancer --tags project
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveTags。

set-load-balancer-listener-ssl-certificate

以下程式碼範例顯示如何使用 set-load-balancer-listener-ssl-certificate。

AWS CLI

更新 HTTPS 負載平衡器的 SSL 憑證

此範例會取代指定 HTTPS 負載平衡器的現有 SSL 憑證。

命令：

```
aws elb set-load-balancer-listener-ssl-certificate --load-balancer-
name my-load-balancer --load-balancer-port 443 --ssl-certificate-
id arn:aws:iam::123456789012:server-certificate/new-server-cert
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetLoadBalancerListenerSslCertificate。

set-load-balancer-policies-for-backend-server

以下程式碼範例顯示如何使用 set-load-balancer-policies-for-backend-server。

AWS CLI

取代與後端執行個體的連接埠相關聯的政策

此範例會取代目前與指定連接埠相關聯的政策。

命令：

```
aws elb set-load-balancer-policies-for-backend-server --load-balancer-name my-load-
balancer --instance-port 80 --policy-names my-ProxyProtocol-policy
```

移除目前與後端執行個體上的連接埠相關聯的所有政策

此範例會移除與指定連接埠相關聯的所有政策。

命令：

```
aws elb set-load-balancer-policies-for-backend-server --load-balancer-name my-load-
balancer --instance-port 80 --policy-names []
```

若要確認政策已移除，請使用 describe-load-balancer-policies命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetLoadBalancerPoliciesForBackendServer。

set-load-balancer-policies-of-listener

以下程式碼範例顯示如何使用 set-load-balancer-policies-of-listener。

AWS CLI

取代與接聽程式相關聯的政策

此範例會取代目前與指定接聽程式相關聯的政策。

命令：

```
aws elb set-load-balancer-policies-of-listener --load-balancer-name my-load-balancer
 --load-balancer-port 443 --policy-names my-SSLNegotiation-policy
```

移除與接聽程式相關聯的所有政策

此範例會移除目前與指定接聽程式相關聯的所有政策。

命令：

```
aws elb set-load-balancer-policies-of-listener --load-balancer-name my-load-balancer
 --load-balancer-port 443 --policy-names []
```

若要確認已從負載平衡器移除政策，請使用 describe-load-balancer-policies命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetLoadBalancerPoliciesOfListener。

Elastic Load Balancing - 使用 的第 2 版範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Elastic Load Balancing - 第 2 版，來執行動作並實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

add-listener-certificates

以下程式碼範例顯示如何使用 add-listener-certificates。

AWS CLI

將憑證新增至安全接聽程式

此範例會將指定的憑證新增至指定的安全接聽程式。

命令：

```
aws elbv2 add-listener-certificates --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 --certificates CertificateArn=arn:aws:acm:us-west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705
```

輸出：

```
{
  "Certificates": [
      {
          "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705",
          "IsDefault": false
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddListenerCertificates。

add-tags

以下程式碼範例顯示如何使用 add-tags。

AWS CLI

將標籤新增至負載平衡器

下列add-tags範例會將 project和 department標籤新增至指定的負載平衡器。

```
aws elbv2 add-tags \
    --resource-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
    --tags "Key=project,Value=lima" "Key=department,Value=digital-media"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddTags。

`create-listener`

以下程式碼範例顯示如何使用 `create-listener`。

AWS CLI

範例 1：建立 HTTP 接聽程式

下列`create-listener`範例會為指定的 Application Load Balancer 建立 HTTP 接聽程式，將請求轉送至指定的目標群組。

```
aws elbv2 create-listener \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
    --protocol HTTP \
    --port 80 \
    --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

如需詳細資訊，請參閱《Application Load Balancer 使用者指南》中的教學課程：使用 AWS CLI 建立 Application Load Balancer。

範例 2：建立 HTTPS 接聽程式

下列`create-listener`範例會為指定的 Application Load Balancer 建立 HTTPS 接聽程式，將請求轉送至指定的目標群組。您必須為 HTTPS 接聽程式指定 SSL 憑證。您可以使用 AWS Certificate Manager (ACM) 建立和管理憑證。或者，您可以使用 SSL/TLS 工具建立憑證、取得憑證授權單位 (CA) 簽署的憑證，並將憑證上傳至 AWS Identity and Access Management (IAM)。

```
aws elbv2 create-listener \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
    --protocol HTTPS \
    --port 443 \
```

```
    --certificates CertificateArn=arn:aws:acm:us-
west-2:123456789012:certificate/3dcb0a41-bd72-4774-9ad9-756919c40557 \
    --ssl-policy ELBSecurityPolicy-2016-08 \
    --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

如需詳細資訊，請參閱《Application Load Balancer 使用者指南》中的新增 HTTPS 接聽程式。

範例 3：建立 TCP 接聽程式

下列create-listener範例會為指定的 Network Load Balancer 建立 TCP 接聽程式，將請求轉送至指定的目標群組。

```
aws elbv2 create-listener \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/net/my-network-load-balancer/5d1b75f4f1cee11e \
    --protocol TCP \
    --port 80 \
    --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-tcp-targets/b6bba954d1361c78
```

如需詳細資訊，請參閱《Network Load Balancer 使用者指南》中的教學課程：使用 AWS CLI 建立 Network Load Balancer。

範例 4：建立 TLS 接聽程式

下列create-listener範例會為指定的 Network Load Balancer 建立 TLS 接聽程式，將請求轉送至指定的目標群組。您必須指定 TLS 接聽程式的 SSL 憑證。

```
aws elbv2 create-listener \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
    --protocol TLS \
    --port 443 \
    --certificates CertificateArn=arn:aws:acm:us-
west-2:123456789012:certificate/3dcb0a41-bd72-4774-9ad9-756919c40557 \
    --ssl-policy ELBSecurityPolicy-2016-08 \
    --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

如需詳細資訊，請參閱《Network Load Balancer 使用者指南》中的 Network Load Balancer 的 TLS 接聽程式。

範例 5：建立 UDP 接聽程式

下列create-listener範例會為指定的 Network Load Balancer 建立 UDP 接聽程式，將請求轉送至指定的目標群組。

```
aws elbv2 create-listener \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/net/my-network-load-balancer/5d1b75f4f1cee11e \
    --protocol UDP \
    --port 53 \
    --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-tcp-targets/b6bba954d1361c78
```

如需詳細資訊，請參閱《Network Load Balancer 使用者指南》中的教學課程：使用 AWS CLI 建立 Network Load Balancer。

範例 6：為指定的閘道和轉送建立接聽程式

下列create-listener範例會為指定的 Gateway Load Balancer 建立接聽程式，將請求轉送至指定的目標群組。

```
aws elbv2 create-listener \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
east-1:850631746142:loadbalancer/gwy/my-gateway-load-balancer/e0f9b3d5c7f7d3d6 \
    --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
east-1:850631746142:targetgroup/my-glb-targets/007ca469fae3bb1615
```

輸出：

```
{
    "Listeners": [
        {
            "ListenerArn": "arn:aws:elasticloadbalancing:us-
east-1:850631746142:listener/gwy/my-agw-lb-example2/e0f9b3d5c7f7d3d6/
afc127db15f925de",
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
east-1:850631746142:loadbalancer/gwy/my-agw-lb-example2/e0f9b3d5c7f7d3d6",
            "DefaultActions": [
                {
                    "Type": "forward",
                    "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
east-1:850631746142:targetgroup/test-tg-agw-2/007ca469fae3bb1615",
```

```
                    "ForwardConfig": {
                        "TargetGroups": [
                            {
                                "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
east-1:850631746142:targetgroup/test-tg-agw-2/007ca469fae3bb1615"
                            }
                        ]
                    }
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Gateway Load Balancer 使用者指南》中的使用 AWS CLI 開始使用 Gateway Load Balancer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateListener。

create-load-balancer

以下程式碼範例顯示如何使用 create-load-balancer。

AWS CLI

範例 1：建立面向網際網路的負載平衡器

下列create-load-balancer範例會建立面向網際網路的 Application Load Balancer，並啟用指定子網路的可用區域。

```
aws elbv2 create-load-balancer \
    --name my-load-balancer \
    --subnets subnet-b7d581c0 subnet-8360a9e7
```

輸出：

```
{
    "LoadBalancers": [
        {
            "Type": "application",
            "Scheme": "internet-facing",
            "IpAddressType": "ipv4",
            "VpcId": "vpc-3ac0fb5f",
```

```
            "AvailabilityZones": [
                {
                    "ZoneName": "us-west-2a",
                    "SubnetId": "subnet-8360a9e7"
                },
                {
                    "ZoneName": "us-west-2b",
                    "SubnetId": "subnet-b7d581c0"
                }
            ],
            "CreatedTime": "2017-08-25T21:26:12.920Z",
            "CanonicalHostedZoneId": "Z2P70J7EXAMPLE",
            "DNSName": "my-load-balancer-424835706.us-west-2.elb.amazonaws.com",
            "SecurityGroups": [
                "sg-5943793c"
            ],
            "LoadBalancerName": "my-load-balancer",
            "State": {
                "Code": "provisioning"
            },
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188"
        }
    ]
}
```

如需詳細資訊，請參閱《Application Load Balancer 使用者指南》中的教學課程：使用 AWS CLI 建立 Application Load Balancer。

範例 2：建立內部負載平衡器

下列`create-load-balancer`範例會建立內部 Application Load Balancer，並啟用指定子網路的可用區域。

```
aws elbv2 create-load-balancer \
    --name my-internal-load-balancer \
    --scheme internal \
    --subnets subnet-b7d581c0 subnet-8360a9e7
```

輸出：

```
{
    "LoadBalancers": [
```

```
        {
            "Type": "application",
            "Scheme": "internal",
            "IpAddressType": "ipv4",
            "VpcId": "vpc-3ac0fb5f",
            "AvailabilityZones": [
                {
                    "ZoneName": "us-west-2a",
                    "SubnetId": "subnet-8360a9e7"
                },
                {
                    "ZoneName": "us-west-2b",
                    "SubnetId": "subnet-b7d581c0"
                }
            ],
            "CreatedTime": "2016-03-25T21:29:48.850Z",
            "CanonicalHostedZoneId": "Z2P70J7EXAMPLE",
            "DNSName": "internal-my-internal-load-balancer-1529930873.us-
west-2.elb.amazonaws.com",
            "SecurityGroups": [
                "sg-5943793c"
            ],
            "LoadBalancerName": "my-internal-load-balancer",
            "State": {
                "Code": "provisioning"
            },
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-internal-load-balancer/5b49b8d4303115c2"
        }
    ]
}
```

如需詳細資訊，請參閱《Application Load Balancer 使用者指南》中的教學課程：使用 AWS CLI 建立 Application Load Balancer。

範例 3：建立 Network Load Balancer

下列create-load-balancer範例會建立面向網際網路的 Network Load Balancer，並啟用指定子網路的可用區域。它使用子網路映射，將指定的彈性 IP 地址與可用區域的負載平衡器節點所使用的網路介面建立關聯。

```
aws elbv2 create-load-balancer \
    --name my-network-load-balancer \
```

```
    --type network \
    --subnet-mappings SubnetId=subnet-b7d581c0,AllocationId=eipalloc-64d5890a
```

輸出：

```
{
    "LoadBalancers": [
        {
            "Type": "network",
            "Scheme": "internet-facing",
            "IpAddressType": "ipv4",
            "VpcId": "vpc-3ac0fb5f",
            "AvailabilityZones": [
                {
                    "LoadBalancerAddresses": [
                        {
                            "IpAddress": "35.161.207.171",
                            "AllocationId": "eipalloc-64d5890a"
                        }
                    ],
                    "ZoneName": "us-west-2b",
                    "SubnetId": "subnet-5264e837"
                }
            ],
            "CreatedTime": "2017-10-15T22:41:25.657Z",
            "CanonicalHostedZoneId": "Z2P70J7EXAMPLE",
            "DNSName": "my-network-load-balancer-5d1b75f4f1cee11e.elb.us-
west-2.amazonaws.com",
            "LoadBalancerName": "my-network-load-balancer",
            "State": {
                "Code": "provisioning"
            },
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/net/my-network-load-balancer/5d1b75f4f1cee11e"
        }
    ]
}
```

如需詳細資訊，請參閱《Network Load Balancer 使用者指南》中的教學課程：使用 AWS CLI 建立 Network Load Balancer。

範例 4：建立閘道Load Balancer

下列create-load-balancer範例會建立 Gateway Load Balancer，並啟用指定子網路的可用區域。

```
aws elbv2 create-load-balancer \
    --name my-gateway-load-balancer \
    --type gateway \
    --subnets subnet-dc83f691 subnet-a62583f9
```

輸出：

```
{
    "LoadBalancers": [
        {
            "Type": "gateway",
            "VpcId": "vpc-838475fe",
            "AvailabilityZones": [
                {
                    "ZoneName": "us-east-1b",
                    "SubnetId": "subnet-a62583f9"
                },
            {
                    "ZoneName": "us-east-1a",
                    "SubnetId": "subnet-dc83f691"
                }
            ],
            "CreatedTime": "2021-07-14T19:33:43.324000+00:00",
            "LoadBalancerName": "my-gateway-load-balancer",
            "State": {
                "Code": "provisioning"
            },
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
east-1:850631746142:loadbalancer/gwy/my-gateway-load-balancer/dfbb5a7d32cdee79"
        }
    ]
}
```

如需詳細資訊，請參閱《Gateway Load Balancer 使用者指南》中的使用 AWS CLI 開始使用 Gateway Load Balancer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLoadBalancer。

create-rule

以下程式碼範例顯示如何使用 create-rule。

AWS CLI

範例 1：使用路徑條件和轉送動作建立規則

如果 URL 包含指定的模式，以下create-rule範例會建立將請求轉送至指定目標群組的規則。

```
aws elbv2 create-rule \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \
    --priority 5 \
    --conditions file://conditions-pattern.json
    --actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

conditions-pattern.json 的內容：

```
[
    {
        "Field": "path-pattern",
        "PathPatternConfig": {
            "Values": ["/images/*"]
        }
    }
]
```

範例 2：使用主機條件和固定回應建立規則

如果主機標頭中的主機名稱符合指定的主機名稱，以下create-rule範例會建立提供固定回應的規則。

```
aws elbv2 create-rule \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \
    --priority 10 \
    --conditions file://conditions-host.json \
    --actions file://actions-fixed-response.json
```

conditions-host.json 的內容

```
[
  {
      "Field": "host-header",
      "HostHeaderConfig": {
          "Values": ["*.example.com"]
      }
  }
]
```

actions-fixed-response.json 的內容

```
[
    {
        "Type": "fixed-response",
        "FixedResponseConfig": {
            "MessageBody": "Hello world",
            "StatusCode": "200",
            "ContentType": "text/plain"
        }
    }
]
```

範例 3：使用來源 IP 地址條件、驗證動作和轉送動作建立規則

下列create-rule範例會建立規則，在來源 IP 地址符合指定的 IP 地址時驗證使用者，並在身分驗證成功時將請求轉送至指定的目標群組。

```
aws elbv2 create-rule \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \
    --priority 20 \
    --conditions file://conditions-source-ip.json \
    --actions file://actions-authenticate.json
```

conditions-source-ip.json 的內容

```
[
    {
        "Field": "source-ip",
        "SourceIpConfig": {
            "Values": ["192.0.2.0/24", "198.51.100.10/32"]
        }
```

```
    }
]
```

actions-authenticate.json 的內容

```
[
    {
        "Type": "authenticate-oidc",
        "AuthenticateOidcConfig": {
            "Issuer": "https://idp-issuer.com",
            "AuthorizationEndpoint": "https://authorization-endpoint.com",
            "TokenEndpoint": "https://token-endpoint.com",
            "UserInfoEndpoint": "https://user-info-endpoint.com",
            "ClientId": "abcdefghijklmnopqrstuvwxyz123456789",
            "ClientSecret": "1234567890123456789012345678 90",
            "SessionCookieName": "my-cookie",
            "SessionTimeout": 3600,
            "Scope": "email",
            "AuthenticationRequestExtraParams": {
                "display": "page",
                "prompt": "login"
            },
            "OnUnauthenticatedRequest": "deny"
        },
        "Order": 1
    },
    {
        "Type": "forward",
        "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
east-1:880185128111:targetgroup/cli-test/642a97ecb0e0f26b",
        "Order": 2
    }
]
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRule。

create-target-group

以下程式碼範例顯示如何使用 create-target-group。

AWS CLI

範例 1：建立 Application Load Balancer 的目標群組

下列create-target-group範例會為 Application Load Balancer 建立目標群組，您可以在其中依執行個體 ID 註冊目標（目標類型為 instance)。此目標群組使用 HTTP 通訊協定、連接埠 80，以及 HTTP 目標群組的預設運作狀態檢查設定。

```
aws elbv2 create-target-group \
    --name my-targets \
    --protocol HTTP \
    --port 80 \
    --target-type instance \
    --vpc-id vpc-3ac0fb5f
```

輸出：

```
{
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
            "TargetGroupName": "my-targets",
            "Protocol": "HTTP",
            "Port": 80,
            "VpcId": "vpc-3ac0fb5f",
            "HealthCheckProtocol": "HTTP",
            "HealthCheckPort": "traffic-port",
            "HealthCheckEnabled": true,
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckTimeoutSeconds": 5,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "HealthCheckPath": "/",
            "Matcher": {
                "HttpCode": "200"
            },
            "TargetType": "instance",
            "ProtocolVersion": "HTTP1",
            "IpAddressType": "ipv4"
        }
    ]
}
```

如需詳細資訊，請參閱《Application Load Balancer 使用者指南》中的建立目標群組。

範例 2：建立目標群組，將流量從 Application Load Balancer 路由到 Lambda 函數

下列create-target-group範例會為 Application Load Balancer 建立目標群組，其中目標為 Lambda 函數（目標類型為 lambda)。預設會停用此目標群組的運作狀態檢查。

```
aws elbv2 create-target-group \
    --name my-lambda-target \
    --target-type lambda
```

輸出：

```
{
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-lambda-target/a3003e085dbb8ddc",
            "TargetGroupName": "my-lambda-target",
            "HealthCheckEnabled": false,
            "HealthCheckIntervalSeconds": 35,
            "HealthCheckTimeoutSeconds": 30,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "HealthCheckPath": "/",
            "Matcher": {
                "HttpCode": "200"
            },
            "TargetType": "lambda",
            "IpAddressType": "ipv4"
        }
    ]
}
```

如需詳細資訊，請參閱 Application Load Balancer 使用者指南的 Lambda 函數做為目標。

範例 3：建立 Network Load Balancer 的目標群組

下列create-target-group範例會為 Network Load Balancer 建立目標群組，您可以在其中依 IP 地址註冊目標（目標類型為 ip)。此目標群組使用 TCP 通訊協定、連接埠 80，以及 TCP 目標群組的預設運作狀態檢查設定。

```
aws elbv2 create-target-group \
    --name my-ip-targets \
```

```
    --protocol TCP \
    --port 80 \
    --target-type ip \
    --vpc-id vpc-3ac0fb5f
```

輸出：

```
{
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-ip-targets/b6bba954d1361c78",
            "TargetGroupName": "my-ip-targets",
            "Protocol": "TCP",
            "Port": 80,
            "VpcId": "vpc-3ac0fb5f",
            "HealthCheckEnabled": true,
            "HealthCheckProtocol": "TCP",
            "HealthCheckPort": "traffic-port",
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckTimeoutSeconds": 10,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "TargetType": "ip",
            "IpAddressType": "ipv4"
        }
    ]
}
```

如需詳細資訊，請參閱《Network Load Balancer 使用者指南》中的建立目標群組。

範例 4：建立目標群組，將流量從 Network Load Balancer 路由到 Application Load Balancer

下列`create-target-group`範例會建立 Network Load Balancer 的目標群組，您可以將 Application Load Balancer 註冊為目標（目標類型為 `alb`)。

aws elbv2 create-target-group --name my-alb-target --protocol TCP --port 80 --target-type alb --vpc-id vpc-3ac0fb5f

輸出：

```
{
    "TargetGroups": [
```

```
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-alb-target/a3003e085dbb8ddc",
            "TargetGroupName": "my-alb-target",
            "Protocol": "TCP",
            "Port": 80,
            "VpcId": "vpc-838475fe",
            "HealthCheckProtocol": "HTTP",
            "HealthCheckPort": "traffic-port",
            "HealthCheckEnabled": true,
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckTimeoutSeconds": 6,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "HealthCheckPath": "/",
            "Matcher": {
                "HttpCode": "200-399"
            },
            "TargetType": "alb",
            "IpAddressType": "ipv4"
        }
    ]
}
```

如需詳細資訊，請參閱《Network Load Balancer 使用者指南》中的使用 Application Load Balancer 建立目標群組做為目標。

範例 5：建立閘道Load Balancer的目標群組

下列create-target-group範例會為 Gateway Load Balancer 建立目標群組，其中目標為執行個體，而目標群組通訊協定為 GENEVE。

```
aws elbv2 create-target-group \
    --name my-glb-targetgroup \
    --protocol GENEVE \
    --port 6081 \
    --target-type instance \
    --vpc-id vpc-838475fe
```

輸出：

```
{
```

```
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-glb-targetgroup/00c3d57eacd6f40b6f",
            "TargetGroupName": "my-glb-targetgroup",
            "Protocol": "GENEVE",
            "Port": 6081,
            "VpcId": "vpc-838475fe",
            "HealthCheckProtocol": "TCP",
            "HealthCheckPort": "80",
            "HealthCheckEnabled": true,
            "HealthCheckIntervalSeconds": 10,
            "HealthCheckTimeoutSeconds": 5,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "TargetType": "instance"
        }
    ]
}
```

如需詳細資訊，請參閱《閘道Load Balancer使用者指南》中的建立目標群組 <https://docs.aws.amazon.com/elasticloadbalancing/latest/gateway/create-target-group.html>`__。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTargetGroup。

delete-listener

以下程式碼範例顯示如何使用 delete-listener。

AWS CLI

刪除接聽程式

下列delete-listener範例會刪除指定的接聽程式。

```
aws elbv2 delete-listener \
    --listener-arn arn:aws:elasticloadbalancing:ua-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteListener。

delete-load-balancer

以下程式碼範例顯示如何使用 delete-load-balancer。

AWS CLI

刪除負載平衡器

下列delete-load-balancer範例會刪除指定的負載平衡器。

```
aws elbv2 delete-load-balancer \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLoadBalancer。

delete-rule

以下程式碼範例顯示如何使用 delete-rule。

AWS CLI

刪除規則

下列delete-rule範例會刪除指定的規則。

```
aws elbv2 delete-rule \
    --rule-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/
app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/1291d13826f405c3
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteRule。

delete-target-group

以下程式碼範例顯示如何使用 delete-target-group。

AWS CLI

刪除目標群組

下列delete-target-group範例會刪除指定的目標群組。

```
aws elbv2 delete-target-group \
```

```
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Application Load Balancer 指南中的刪除Application Load Balancer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTargetGroup。

deregister-targets

以下程式碼範例顯示如何使用 deregister-targets。

AWS CLI

範例 1：從目標群組取消註冊目標

下列deregister-targets範例會從指定的目標群組中移除指定的執行個體。

```
aws elbv2 deregister-targets \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 \
    --targets Id=i-1234567890abcdef0
```

範例 2：取消註冊使用連接埠覆寫註冊的目標

下列deregister-targets範例會從使用連接埠覆寫註冊的目標群組中移除執行個體。

```
aws elbv2 deregister-targets \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-internal-targets/3bb63f11dfb0faf9 \
    --targets Id=i-1234567890abcdef0,Port=80 Id=i-1234567890abcdef0,Port=766
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterTargets。

describe-account-limits

以下程式碼範例顯示如何使用 describe-account-limits。

AWS CLI

描述您的 Elastic Load Balancing 限制

下列describe-account-limits範例顯示目前區域中 AWS 您帳戶的 Elastic Load Balancing 限制。

```
aws elbv2 describe-account-limits
```

輸出：

```
{
    "Limits": [
        {
            "Name": "target-groups",
            "Max": "3000"
        },
        {
            "Name": "targets-per-application-load-balancer",
            "Max": "1000"
        },
        {
            "Name": "listeners-per-application-load-balancer",
            "Max": "50"
        },
        {
            "Name": "rules-per-application-load-balancer",
            "Max": "100"
        },
        {
            "Name": "network-load-balancers",
            "Max": "50"
        },
        {
            "Name": "targets-per-network-load-balancer",
            "Max": "3000"
        },
        {
            "Name": "targets-per-availability-zone-per-network-load-balancer",
            "Max": "500"
        },
        {
            "Name": "listeners-per-network-load-balancer",
            "Max": "50"
        },
        {
            "Name": "condition-values-per-alb-rule",
```

```
            "Max": "5"
        },
        {
            "Name": "condition-wildcards-per-alb-rule",
            "Max": "5"
        },
        {
            "Name": "target-groups-per-application-load-balancer",
            "Max": "100"
        },
        {
            "Name": "target-groups-per-action-on-application-load-balancer",
            "Max": "5"
        },
        {
            "Name": "target-groups-per-action-on-network-load-balancer",
            "Max": "1"
        },
        {
            "Name": "certificates-per-application-load-balancer",
            "Max": "25"
        },
        {
            "Name": "certificates-per-network-load-balancer",
            "Max": "25"
        },
        {
            "Name": "targets-per-target-group",
            "Max": "1000"
        },
        {
            "Name": "target-id-registrations-per-application-load-balancer",
            "Max": "1000"
        },
        {
            "Name": "network-load-balancer-enis-per-vpc",
            "Max": "1200"
        },
        {
            "Name": "application-load-balancers",
            "Max": "50"
        },
        {
            "Name": "gateway-load-balancers",
```

```
            "Max": "100"
        },
        {
            "Name": "gateway-load-balancers-per-vpc",
            "Max": "100"
        },
        {
            "Name": "geneve-target-groups",
            "Max": "100"
        },
        {
            "Name": "targets-per-availability-zone-per-gateway-load-balancer",
            "Max": "300"
        }
    ]
}
```

如需詳細資訊，請參閱《 AWS 一般參考》中的配額。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAccountLimits。

describe-listener-certificates

以下程式碼範例顯示如何使用 describe-listener-certificates。

AWS CLI

描述安全接聽程式的憑證

此範例說明指定安全接聽程式的憑證。

命令：

```
aws elbv2 describe-listener-certificates --listener-
arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-
balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

輸出：

```
{
  "Certificates": [
      {
```

```
            "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705",
            "IsDefault": false
        },
        {
            "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/3dcb0a41-bd72-4774-9ad9-756919c40557",
            "IsDefault": false
        },
        {
            "CertificateArn": "arn:aws:acm:us-west-2:123456789012:certificate/
fe59da96-6f58-4a22-8eed-6d0d50477e1d",
            "IsDefault": true
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeListenerCertificates。

describe-listeners

以下程式碼範例顯示如何使用 describe-listeners。

AWS CLI

描述接聽程式

此範例說明指定的接聽程式。

命令：

```
aws elbv2 describe-listeners --listener-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

輸出：

```
{
    "Listeners": [
        {
            "Port": 80,
            "Protocol": "HTTP",
            "DefaultActions": [
```

```
            {
                "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
                "Type": "forward"
            }
        ],
        "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
        "ListenerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2"
    }
  ]
}
```

描述負載平衡器的接聽程式

此範例說明指定負載平衡器的接聽程式。

命令：

```
aws elbv2 describe-listeners --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

輸出：

```
{
  "Listeners": [
      {
          "Port": 443,
          "Protocol": "HTTPS",
          "DefaultActions": [
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
                  "Type": "forward"
              }
          ],
          "SslPolicy": "ELBSecurityPolicy-2015-05",
          "Certificates": [
              {
                  "CertificateArn": "arn:aws:iam::123456789012:server-certificate/
my-server-cert"
```

```
            }
        ],
        "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
        "ListenerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/0467ef3c8400ae65"
    },
    {
        "Port": 80,
        "Protocol": "HTTP",
        "DefaultActions": [
            {
                "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
                "Type": "forward"
            }
        ],
        "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
        "ListenerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2"
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeListeners。

describe-load-balancer-attributes

以下程式碼範例顯示如何使用 `describe-load-balancer-attributes`。

AWS CLI

描述負載平衡器屬性

下列`describe-load-balancer-attributes`範例顯示指定負載平衡器的屬性。

```
aws elbv2 describe-load-balancer-attributes \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

下列範例輸出顯示 Application Load Balancer 的屬性。

```
{
    "Attributes": [
        {
            "Value": "false",
            "Key": "access_logs.s3.enabled"
        },
        {
            "Value": "",
            "Key": "access_logs.s3.bucket"
        },
        {
            "Value": "",
            "Key": "access_logs.s3.prefix"
        },
        {
            "Value": "60",
            "Key": "idle_timeout.timeout_seconds"
        },
        {
            "Value": "false",
            "Key": "deletion_protection.enabled"
        },
        {
            "Value": "true",
            "Key": "routing.http2.enabled"
        }
    ]
}
```

下列範例輸出包含 Network Load Balancer 的屬性。

```
{
    "Attributes": [
        {
            "Value": "false",
            "Key": "access_logs.s3.enabled"
        },
        {
            "Value": "",
            "Key": "access_logs.s3.bucket"
        },
        {
            "Value": "",
```

```
            "Key": "access_logs.s3.prefix"
        },
        {
            "Value": "false",
            "Key": "deletion_protection.enabled"
        },
        {
            "Value": "false",
            "Key": "load_balancing.cross_zone.enabled"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLoadBalancerAttributes。

`describe-load-balancers`

以下程式碼範例顯示如何使用 `describe-load-balancers`。

AWS CLI

描述負載平衡器

此範例說明指定的負載平衡器。

命令：

```
aws elbv2 describe-load-balancers --load-balancer-
arns arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-
balancer/50dc6c495c0c9188
```

輸出：

```
{
  "LoadBalancers": [
      {
          "Type": "application",
          "Scheme": "internet-facing",
          "IpAddressType": "ipv4",
          "VpcId": "vpc-3ac0fb5f",
          "AvailabilityZones": [
              {
```

```
                "ZoneName": "us-west-2a",
                "SubnetId": "subnet-8360a9e7"
            },
            {
                "ZoneName": "us-west-2b",
                "SubnetId": "subnet-b7d581c0"
            }
        ],
        "CreatedTime": "2016-03-25T21:26:12.920Z",
        "CanonicalHostedZoneId": "Z2P70J7EXAMPLE",
        "DNSName": "my-load-balancer-424835706.us-west-2.elb.amazonaws.com",
        "SecurityGroups": [
            "sg-5943793c"
        ],
        "LoadBalancerName": "my-load-balancer",
        "State": {
            "Code": "active"
        },
        "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188"
      }
  ]
}
```

描述所有負載平衡器

此範例說明所有負載平衡器。

命令：

```
aws elbv2 describe-load-balancers
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLoadBalancers。

describe-rules

以下程式碼範例顯示如何使用 describe-rules。

AWS CLI

範例 1：描述規則

下列describe-rules範例顯示指定規則的詳細資訊。

```
aws elbv2 describe-rules \
    --rule-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/
app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/9683b2d02a6cabee
```

範例 2：描述接聽程式的規則

下列describe-rules範例顯示指定接聽程式之規則的詳細資訊。輸出包含預設規則以及您新增的任何其他規則。

```
aws elbv2 describe-rules \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRules。

describe-ssl-policies

以下程式碼範例顯示如何使用 describe-ssl-policies。

AWS CLI

範例 1：依負載平衡器類型列出用於 SSL 溝通的政策

下列describe-ssl-policies範例顯示可用於與 Application Load Balancer 進行 SSL 溝通的政策名稱。此範例使用 --query 參數來僅顯示政策的名稱。

```
aws elbv2 describe-ssl-policies \
    --load-balancer-type application \
    --query SslPolicies[*].Name
```

輸出：

```
[
    "ELBSecurityPolicy-2016-08",
    "ELBSecurityPolicy-TLS13-1-2-2021-06",
    "ELBSecurityPolicy-TLS13-1-2-Res-2021-06",
    "ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06",
    "ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06",
    "ELBSecurityPolicy-TLS13-1-1-2021-06",
    "ELBSecurityPolicy-TLS13-1-0-2021-06",
```

```
    "ELBSecurityPolicy-TLS13-1-3-2021-06",
    "ELBSecurityPolicy-TLS-1-2-2017-01",
    "ELBSecurityPolicy-TLS-1-1-2017-01",
    "ELBSecurityPolicy-TLS-1-2-Ext-2018-06",
    "ELBSecurityPolicy-FS-2018-06",
    "ELBSecurityPolicy-2015-05",
    "ELBSecurityPolicy-TLS-1-0-2015-04",
    "ELBSecurityPolicy-FS-1-2-Res-2019-08",
    "ELBSecurityPolicy-FS-1-1-2019-08",
    "ELBSecurityPolicy-FS-1-2-2019-08",
    "ELBSecurityPolicy-FS-1-2-Res-2020-10"
]
```

範例 2：列出支援特定通訊協定的政策

下列describe-ssl-policies範例顯示支援 TLS 1.3 通訊協定的政策名稱。此範例使用 --query 參數來僅顯示政策的名稱。

```
aws elbv2 describe-ssl-policies \
    --load-balancer-type application \
    --query SslPolicies[?contains(SslProtocols,'TLSv1.3')].Name
```

輸出：

```
[
    "ELBSecurityPolicy-TLS13-1-2-2021-06",
    "ELBSecurityPolicy-TLS13-1-2-Res-2021-06",
    "ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06",
    "ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06",
    "ELBSecurityPolicy-TLS13-1-1-2021-06",
    "ELBSecurityPolicy-TLS13-1-0-2021-06",
    "ELBSecurityPolicy-TLS13-1-3-2021-06"
]
```

範例 3：顯示政策的密碼

下列describe-ssl-policies範例顯示指定政策的密碼名稱。此範例使用 --query 參數僅顯示密碼名稱。清單中的第一個密碼具有優先順序 1，其餘的密碼會依優先順序排列。

```
aws elbv2 describe-ssl-policies \
    --names ELBSecurityPolicy-TLS13-1-2-2021-06 \
```

```
    --query SslPolicies[*].Ciphers[*].Name
```

輸出：

```
[
    "TLS_AES_128_GCM_SHA256",
    "TLS_AES_256_GCM_SHA384",
    "TLS_CHACHA20_POLY1305_SHA256",
    "ECDHE-ECDSA-AES128-GCM-SHA256",
    "ECDHE-RSA-AES128-GCM-SHA256",
    "ECDHE-ECDSA-AES128-SHA256",
    "ECDHE-RSA-AES128-SHA256",
    "ECDHE-ECDSA-AES256-GCM-SHA384",
    "ECDHE-RSA-AES256-GCM-SHA384",
    "ECDHE-ECDSA-AES256-SHA384",
    "ECDHE-RSA-AES256-SHA384"
]
```

如需詳細資訊，請參閱《Application Load Balancer 使用者指南》中的安全政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSslPolicies。

describe-tags

以下程式碼範例顯示如何使用 describe-tags。

AWS CLI

描述指派給負載平衡器的標籤

此範例說明指派給指定負載平衡器的標籤。

命令：

```
aws elbv2 describe-tags --resource-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

輸出：

```
{
  "TagDescriptions": [
      {
```

```
            "ResourceArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
            "Tags": [
                {
                    "Value": "lima",
                    "Key": "project"
                },
                {
                    "Value": "digital-media",
                    "Key": "department"
                }
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTags。

describe-target-group-attributes

以下程式碼範例顯示如何使用 describe-target-group-attributes。

AWS CLI

描述目標群組屬性

下列describe-target-group-attributes範例顯示指定目標群組的屬性。

```
aws elbv2 describe-target-group-attributes \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

如果通訊協定為 HTTP 或 HTTPS，且目標類型為 instance或 ，則輸出會包含 屬性ip。

```
{
    "Attributes": [
        {
            "Value": "false",
            "Key": "stickiness.enabled"
        },
        {
            "Value": "300",
```

```
            "Key": "deregistration_delay.timeout_seconds"
        },
        {
            "Value": "lb_cookie",
            "Key": "stickiness.type"
        },
        {
            "Value": "86400",
            "Key": "stickiness.lb_cookie.duration_seconds"
        },
        {
            "Value": "0",
            "Key": "slow_start.duration_seconds"
        }
    ]
}
```

如果通訊協定為 HTTP 或 HTTPS，且目標類型為 ，則下列輸出會包含屬性lambda。

```
{
    "Attributes": [
        {
            "Value": "false",
            "Key": "lambda.multi_value_headers.enabled"
        }
    ]
}
```

如果通訊協定為 TCP、TLS、UDP 或 TCP_UDP，則以下輸出包含屬性。

```
{
    "Attributes": [
        {
            "Value": "false",
            "Key": "proxy_protocol_v2.enabled"
        },
        {
            "Value": "300",
            "Key": "deregistration_delay.timeout_seconds"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTargetGroupAttributes。

describe-target-groups

以下程式碼範例顯示如何使用 describe-target-groups。

AWS CLI

範例 1：描述目標群組

下列describe-target-groups範例顯示指定目標群組的詳細資訊。

```
aws elbv2 describe-target-groups \
    --target-group-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

輸出：

```
{
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
            "TargetGroupName": "my-targets",
            "Protocol": "HTTP",
            "Port": 80,
            "VpcId": "vpc-3ac0fb5f",
            "HealthCheckProtocol": "HTTP",
            "HealthCheckPort": "traffic-port",
            "HealthCheckEnabled": true,
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckTimeoutSeconds": 5,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "HealthCheckPath": "/",
            "Matcher": {
                "HttpCode": "200"
            },
            "LoadBalancerArns": [
                "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/
app/my-load-balancer/50dc6c495c0c9188"
            ],
```

```
            "TargetType": "instance",
            "ProtocolVersion": "HTTP1",
            "IpAddressType": "ipv4"
        }
    ]
}
```

範例 2：描述負載平衡器的所有目標群組

下列describe-target-groups範例顯示指定負載平衡器之所有目標群組的詳細資訊。此範例使用 --query 參數僅顯示目標群組名稱。

```
aws elbv2 describe-target-groups \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
    --query TargetGroups[*].TargetGroupName
```

輸出：

```
[
    "my-instance-targets",
    "my-ip-targets",
    "my-lambda-target"
]
```

如需詳細資訊，請參閱《應用程式負載平衡器指南》中的目標群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTargetGroups。

describe-target-health

以下程式碼範例顯示如何使用 describe-target-health。

AWS CLI

範例 1：描述目標群組目標的運作狀態

下列describe-target-health範例顯示指定目標群組目標的運作狀態詳細資訊。這些目標運作狀態良好。

```
aws elbv2 describe-target-health \
```

```
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

輸出：

```
{
    "TargetHealthDescriptions": [
        {
            "HealthCheckPort": "80",
            "Target": {
                "Id": "i-ceddcd4d",
                "Port": 80
            },
            "TargetHealth": {
                "State": "healthy"
            }
        },
        {
            "HealthCheckPort": "80",
            "Target": {
                "Id": "i-0f76fade",
                "Port": 80
            },
            "TargetHealth": {
                "State": "healthy"
            }
        }
    ]
}
```

範例 2：描述目標的運作狀態

下列`describe-target-health`範例顯示指定目標的運作狀態詳細資訊。此目標運作狀態良好。

```
aws elbv2 describe-target-health \
    --targets Id=i-0f76fade,Port=80 \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

輸出：

```
{
```

```
    "TargetHealthDescriptions": [
        {
            "HealthCheckPort": "80",
            "Target": {
                "Id": "i-0f76fade",
                "Port": 80
            },
            "TargetHealth": {
                "State": "healthy"
            }
        }
    ]
}
```

下列範例輸出適用於在接聽程式的 動作中未指定目標群組的目標。此目標無法接收來自負載平衡器的流量。

```
{
    "TargetHealthDescriptions": [
    {
        "HealthCheckPort": "80",
        "Target": {
            "Id": "i-0f76fade",
            "Port": 80
        },
            "TargetHealth": {
                "State": "unused",
                "Reason": "Target.NotInUse",
                "Description": "Target group is not configured to receive traffic
 from the load balancer"
            }
        }
    ]
}
```

下列範例輸出適用於目標，其目標群組僅在接聽程式的 動作中指定。目標仍在註冊中。

```
{
    "TargetHealthDescriptions": [
        {
            "HealthCheckPort": "80",
            "Target": {
                "Id": "i-0f76fade",
```

```
                "Port": 80
            },
            "TargetHealth": {
                "State": "initial",
                "Reason": "Elb.RegistrationInProgress",
                "Description": "Target registration is in progress"
            }
        }
    ]
}
```

下列範例輸出適用於運作狀態不佳的目標。

```
{
    "TargetHealthDescriptions": [
        {
            "HealthCheckPort": "80",
            "Target": {
                "Id": "i-0f76fade",
                "Port": 80
            },
            "TargetHealth": {
                "State": "unhealthy",
                "Reason": "Target.Timeout",
                "Description": "Connection to target timed out"
            }
        }
    ]
}
```

下列範例輸出適用於 Lambda 函數的目標，且運作狀態檢查已停用。

```
{
    "TargetHealthDescriptions": [
        {
            "Target": {
                "Id": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
                "AvailabilityZone": "all",
            },
            "TargetHealth": {
                "State": "unavailable",
                "Reason": "Target.HealthCheckDisabled",
                "Description": "Health checks are not enabled for this target"
```

```
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTargetHealth。

modify-listener

以下程式碼範例顯示如何使用 modify-listener。

AWS CLI

範例 1：將預設動作變更為轉送動作

下列modify-listener範例會將預設動作變更為指定接聽程式forward的動作。

```
aws elbv2 modify-listener \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \
    --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-new-targets/2453ed029918f21f
```

輸出：

```
{
    "Listeners": [
        {
            "ListenerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/
f2f7dc8efc522ab2",
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
            "Protocol": "HTTP",
            "Port": 80,
            "DefaultActions": [
                {
                    "Type": "forward",
                    "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-new-targets/2453ed029918f21f"
                }
            ]
```

```
        }
    ]
}
```

範例 2：將預設動作變更為重新導向動作

下列modify-listener範例會將預設動作變更為指定接聽程式redirect的動作。

```
aws elbv2 modify-listener \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \
    --default-actions Type=redirect,
 RedirectConfig='{Protocol=HTTPS,StatusCode=HTTP_302}'
```

輸出：

```
{
    "Listeners": [
        {
            "ListenerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/
f2f7dc8efc522ab2",
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
            "Protocol": "HTTP",
            "Port": 80,
            "DefaultActions": [
                {
                    "Type": "redirect",
                    "RedirectConfig": {
                        "Protocol": "HTTPS",
                        "Port": "#{port}",
                        "Host": "#{host}",
                        "Path": "/#{path}",
                        "Query": "#{query}",
                        "StatusCode": "HTTP_302",
                    }
                }
            ]
        }
    ]
}
```

範例 3：變更伺服器憑證

下列modify-listener範例會變更指定 HTTPS 接聽程式的伺服器憑證。

```
aws elbv2 modify-listener \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/0467ef3c8400ae65 \
    --certificates CertificateArn=arn:aws:iam::123456789012:server-certificate/my-new-server-cert
```

輸出：

```
{
    "Listeners": [
        {
            "ListenerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/0467ef3c8400ae65",
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
            "Protocol": "HTTPS",
            "Port": 443,
            "DefaultActions": [
                {
                    "Type": "forward",
                    "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067"
                }
            ],
            "SslPolicy": "ELBSecurityPolicy-2015-05",
            "Certificates": [
                {
                    "CertificateArn": "arn:aws:iam::123456789012:server-certificate/my-new-server-cert"
                }
            ],
        }
    ]
}
```

如需詳細資訊，請參閱 Application Load Balancer 使用者指南中的接聽程式規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyListener。

modify-load-balancer-attributes

以下程式碼範例顯示如何使用 modify-load-balancer-attributes。

AWS CLI

啟用刪除保護

此範例會啟用指定負載平衡器的刪除保護。

命令：

```
aws elbv2 modify-load-balancer-attributes --load-balancer-
arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-
balancer/50dc6c495c0c9188 --attributes Key=deletion_protection.enabled,Value=true
```

輸出：

```
{
  "Attributes": [
      {
          "Value": "true",
          "Key": "deletion_protection.enabled"
      },
      {
          "Value": "false",
          "Key": "access_logs.s3.enabled"
      },
      {
          "Value": "60",
          "Key": "idle_timeout.timeout_seconds"
      },
      {
          "Value": "",
          "Key": "access_logs.s3.prefix"
      },
      {
          "Value": "",
          "Key": "access_logs.s3.bucket"
      }
  ]
}
```

變更閒置逾時

此範例會變更指定負載平衡器的閒置逾時值。

命令：

```
aws elbv2 modify-load-balancer-attributes --load-balancer-
arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-
balancer/50dc6c495c0c9188 --attributes Key=idle_timeout.timeout_seconds,Value=30
```

輸出：

```
{
  "Attributes": [
      {
          "Value": "30",
          "Key": "idle_timeout.timeout_seconds"
      },
      {
          "Value": "false",
          "Key": "access_logs.s3.enabled"
      },
      {
          "Value": "",
          "Key": "access_logs.s3.prefix"
      },
      {
          "Value": "true",
          "Key": "deletion_protection.enabled"
      },
      {
          "Value": "",
          "Key": "access_logs.s3.bucket"
      }
  ]
}
```

啟用存取日誌

此範例會啟用指定負載平衡器的存取日誌。請注意，S3 儲存貯體必須與負載平衡器位於相同的區域中，而且必須連接政策以授予 Elastic Load Balancing 服務的存取權。

命令：

```
aws elbv2 modify-load-balancer-attributes --load-balancer-
arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-
balancer/50dc6c495c0c9188 --
attributes Key=access_logs.s3.enabled,Value=true Key=access_logs.s3.bucket,Value=my-
loadbalancer-logs Key=access_logs.s3.prefix,Value=myapp
```

輸出：

```
{
  "Attributes": [
      {
          "Value": "true",
          "Key": "access_logs.s3.enabled"
      },
      {
          "Value": "my-load-balancer-logs",
          "Key": "access_logs.s3.bucket"
      },
      {
          "Value": "myapp",
          "Key": "access_logs.s3.prefix"
      },
      {
          "Value": "60",
          "Key": "idle_timeout.timeout_seconds"
      },
      {
          "Value": "false",
          "Key": "deletion_protection.enabled"
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyLoadBalancerAttributes。

modify-rule

以下程式碼範例顯示如何使用 modify-rule。

AWS CLI

修改規則

下列modify-rule範例會更新指定規則的動作和條件。

```
aws elbv2 modify-rule \
  --actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 \
  --conditions Field=path-pattern,Values='/images/*'
  --rule-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/9683b2d02a6cabee
```

輸出：

```
{
    "Rules": [
        {
            "Priority": "10",
            "Conditions": [
                {
                    "Field": "path-pattern",
                    "Values": [
                        "/images/*"
                    ]
                }
            ],
            "RuleArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener-rule/app/my-load-balancer/50dc6c495c0c9188/
f2f7dc8efc522ab2/9683b2d02a6cabee",
            "IsDefault": false,
            "Actions": [
                {
                    "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
                    "Type": "forward"
                }
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyRule。

modify-target-group-attributes

以下程式碼範例顯示如何使用 modify-target-group-attributes。

AWS CLI

修改取消註冊延遲逾時

此範例會將取消註冊延遲逾時設定為指定目標群組的指定值。

命令：

```
aws elbv2 modify-target-group-attributes --target-group-
arn arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067 --
attributes Key=deregistration_delay.timeout_seconds,Value=600
```

輸出：

```
{
  "Attributes": [
      {
          "Value": "false",
          "Key": "stickiness.enabled"
      },
      {
          "Value": "600",
          "Key": "deregistration_delay.timeout_seconds"
      },
      {
          "Value": "lb_cookie",
          "Key": "stickiness.type"
      },
      {
          "Value": "86400",
          "Key": "stickiness.lb_cookie.duration_seconds"
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyTargetGroupAttributes。

modify-target-group

以下程式碼範例顯示如何使用 modify-target-group。

AWS CLI

修改目標群組的運作狀態檢查組態

下列modify-target-group範例會變更用於評估指定目標群組之目標運作狀態的運作狀態檢查組態。請注意，由於 CLI 剖析逗號的方式，您必須使用單引號括住--matcher選項的範圍，而不是雙引號。

```
aws elbv2 modify-target-group \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-https-targets/2453ed029918f21f \
    --health-check-protocol HTTPS \
    --health-check-port 443 \
    --matcher HttpCode='200,299'
```

輸出：

```
{
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-https-targets/2453ed029918f21f",
            "TargetGroupName": "my-https-targets",
            "Protocol": "HTTPS",
            "Port": 443,
            "VpcId": "vpc-3ac0fb5f",
            "HealthCheckProtocol": "HTTPS",
            "HealthCheckPort": "443",
            "HealthCheckEnabled": true,
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckTimeoutSeconds": 5,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "Matcher": {
                "HttpCode": "200,299"
            },
            "LoadBalancerArns": [
                "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/
app/my-load-balancer/50dc6c495c0c9188"
```

```
            ],
            "TargetType": "instance",
            "ProtocolVersion": "HTTP1",
            "IpAddressType": "ipv4"
        }
    ]
}
```

如需詳細資訊，請參閱《應用程式負載平衡器指南》中的目標群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyTargetGroup。

register-targets

以下程式碼範例顯示如何使用 register-targets。

AWS CLI

範例 1：依執行個體 ID 向目標群組註冊目標

下列register-targets範例會向目標群組註冊指定的執行個體。目標群組的目標類型必須為 instance。

```
aws elbv2 register-targets \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 \
    --targets Id=i-1234567890abcdef0 Id=i-0abcdef1234567890
```

範例 2：使用連接埠覆寫向目標群組註冊目標

下列register-targets範例使用多個連接埠向目標群組註冊指定的執行個體。這可讓您在與目標群組中的目標相同的執行個體上註冊容器。

```
aws elbv2 register-targets \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-internal-targets/3bb63f11dfb0faf9 \
    --targets Id=i-0598c7d356eba48d7,Port=80 Id=i-0598c7d356eba48d7,Port=766
```

範例 3：依 IP 地址向目標群組註冊目標

下列register-targets範例會向目標群組註冊指定的 IP 地址。目標群組的目標類型必須為 ip。

```
aws elbv2 register-targets \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-tcp-ip-targets/8518e899d173178f \
    --targets Id=10.0.1.15 Id=10.0.1.23
```

範例 4：將 Lambda 函數註冊為目標

下列register-targets範例會向目標群組註冊指定的 IP 地址。目標群組的目標類型必須為lambda。您必須授予 Elastic Load Balancing 調用 Lambda 函數的許可。

```
aws elbv2 register-targets \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-tcp-ip-targets/8518e899d173178f \
    --targets Id=arn:aws:lambda:us-west-2:123456789012:function:my-function
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterTargets。

remove-listener-certificates

以下程式碼範例顯示如何使用 remove-listener-certificates。

AWS CLI

從安全接聽程式移除憑證

此範例會從指定的安全接聽程式中移除指定的憑證。

命令：

```
aws elbv2 remove-listener-certificates --listener-
arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/
app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 --
certificates CertificateArn=arn:aws:acm:us-west-2:123456789012:certificate/5cc54884-
f4a3-4072-80be-05b9ba72f705
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveListenerCertificates。

remove-tags

以下程式碼範例顯示如何使用 remove-tags。

AWS CLI

從負載平衡器移除標籤

下列remove-tags範例會從指定的負載平衡器移除 project和 department標籤。

```
aws elbv2 remove-tags \
    --resource-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
    --tag-keys project department
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveTags。

set-ip-address-type

以下程式碼範例顯示如何使用 set-ip-address-type。

AWS CLI

設定負載平衡器的地址類型

此範例會將指定負載平衡器的地址類型設定為 dualstack。負載平衡器子網路必須具有相關聯的 IPv6 CIDR 區塊。

命令：

```
aws elbv2 set-ip-address-type --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 --ip-address-
type dualstack
```

輸出：

```
{
    "IpAddressType": "dualstack"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetIpAddressType。

set-rule-priorities

以下程式碼範例顯示如何使用 set-rule-priorities。

AWS CLI

設定規則優先順序

此範例會設定指定規則的優先順序。

命令：

```
aws elbv2 set-rule-priorities --rule-
priorities RuleArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-
rule/app/my-load-balancer/50dc6c495c0c9188/
f2f7dc8efc522ab2/1291d13826f405c3,Priority=5
```

輸出：

```
{
  "Rules": [
      {
          "Priority": "5",
          "Conditions": [
              {
                  "Field": "path-pattern",
                  "Values": [
                      "/img/*"
                  ]
              }
          ],
          "RuleArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-
rule/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/1291d13826f405c3",
          "IsDefault": false,
          "Actions": [
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
                  "Type": "forward"
              }
          ]
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetRulePriorities。

set-security-groups

以下程式碼範例顯示如何使用 set-security-groups。

AWS CLI

將安全群組與負載平衡器建立關聯

此範例會將指定的安全群組與指定的負載平衡器建立關聯。

命令：

```
aws elbv2 set-security-groups --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 --security-groups sg-5943793c
```

輸出：

```
{
  "SecurityGroupIds": [
      "sg-5943793c"
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetSecurityGroups。

set-subnets

以下程式碼範例顯示如何使用 set-subnets。

AWS CLI

啟用負載平衡器的可用區域

此範例會啟用指定負載平衡器之指定子網路的可用區域。

命令：

```
aws elbv2 set-subnets --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 --subnets subnet-8360a9e7 subnet-b7d581c0
```

輸出：

```
{
  "AvailabilityZones": [
      {
          "SubnetId": "subnet-8360a9e7",
          "ZoneName": "us-west-2a"
      },
      {
          "SubnetId": "subnet-b7d581c0",
          "ZoneName": "us-west-2b"
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetSubnets。

使用 的 Elastic Transcoder 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Elastic Transcoder 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`cancel-job`

以下程式碼範例顯示如何使用 `cancel-job`。

AWS CLI

取消 ElasticTranscoder 的任務

這會取消 ElasticTranscoder 的指定任務。

命令：

```
aws elastictranscoder cancel-job --id 3333333333333-abcde3
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelJob。

create-job

以下程式碼範例顯示如何使用 create-job。

AWS CLI

為 ElasticTranscoder 建立任務

下列create-job範例會建立 ElasticTranscoder 的任務。

```
aws elastictranscoder create-job \
    --pipeline-id 1111111111111-abcde1 \
    --inputs file://inputs.json \
    --outputs file://outputs.json \
    --output-key-prefix "recipes/" \
    --user-metadata file://user-metadata.json
```

inputs.json 的內容：

```
[{
    "Key":"ETS_example_file.mp4",
    "FrameRate":"auto",
    "Resolution":"auto",
    "AspectRatio":"auto",
    "Interlaced":"auto",
    "Container":"mp4"
}]
```

output.json 的內容：

```
[
    {
        "Key":"webm/ETS_example_file-kindlefirehd.webm",
        "Rotate":"0",
        "PresetId":"1351620000001-100250"
    }
]
```

user-metadata.json 的內容：

```
{
    "Food type":"Italian",
    "Cook book":"recipe notebook"
}
```

輸出：

```
{
    "Job": {
        "Status": "Submitted",
        "Inputs": [
            {
                "Container": "mp4",
                "FrameRate": "auto",
                "Key": "ETS_example_file.mp4",
                "AspectRatio": "auto",
                "Resolution": "auto",
                "Interlaced": "auto"
            }
        ],
        "Playlists": [],
        "Outputs": [
            {
                "Status": "Submitted",
                "Rotate": "0",
                "PresetId": "1351620000001-100250",
                "Watermarks": [],
                "Key": "webm/ETS_example_file-kindlefirehd.webm",
                "Id": "1"
            }
        ],
        "PipelineId": "3333333333333-abcde3",
        "OutputKeyPrefix": "recipes/",
        "UserMetadata": {
            "Cook book": "recipe notebook",
            "Food type": "Italian"
        },
        "Output": {
            "Status": "Submitted",
            "Rotate": "0",
            "PresetId": "1351620000001-100250",
```

```
            "Watermarks": [],
            "Key": "webm/ETS_example_file-kindlefirehd.webm",
            "Id": "1"
        },
        "Timing": {
            "SubmitTimeMillis": 1533838012298
        },
        "Input": {
            "Container": "mp4",
            "FrameRate": "auto",
            "Key": "ETS_example_file.mp4",
            "AspectRatio": "auto",
            "Resolution": "auto",
            "Interlaced": "auto"
        },
        "Id": "1533838012294-example",
        "Arn": "arn:aws:elastictranscoder:us-west-2:123456789012:job/1533838012294-
example"
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateJob。

create-pipeline

以下程式碼範例顯示如何使用 create-pipeline。

AWS CLI

建立 ElasticTranscoder 的管道

下列create-pipeline範例會建立 ElasticTranscoder 的管道。

```
aws elastictranscoder create-pipeline \
    --name Default \
    --input-bucket salesoffice.example.com-source \
    --role arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role \
    --notifications Progressing="",Completed="",Warning="",Error=arn:aws:sns:us-
east-1:111222333444:ETS_Errors \
    --content-config file://content-config.json \
    --thumbnail-config file://thumbnail-config.json
```

content-config.json 的內容：

```
{
    "Bucket":"salesoffice.example.com-public-promos",
    "Permissions":[
        {
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
            "Access":[
            "FullControl"
            ]
        }
    ],
    "StorageClass":"Standard"
}
```

thumbnail-config.json 的內容：

```
{
    "Bucket":"salesoffice.example.com-public-promos-thumbnails",
    "Permissions":[
        {
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
            "Access":[
                "FullControl"
            ]
        }
    ],
    "StorageClass":"ReducedRedundancy"
}
```

輸出：

```
{
    "Pipeline": {
        "Status": "Active",
        "ContentConfig": {
            "Bucket": "salesoffice.example.com-public-promos",
            "StorageClass": "Standard",
            "Permissions": [
                {
                    "Access": [
                        "FullControl"
```

```
                    ],
                    "Grantee": "marketing-promos@example.com",
                    "GranteeType": "Email"
                }
            ]
        },
        "Name": "Default",
        "ThumbnailConfig": {
            "Bucket": "salesoffice.example.com-public-promos-thumbnails",
            "StorageClass": "ReducedRedundancy",
            "Permissions": [
                {
                    "Access": [
                        "FullControl"
                    ],
                    "Grantee": "marketing-promos@example.com",
                    "GranteeType": "Email"
                }
            ]
        },
        "Notifications": {
            "Completed": "",
            "Warning": "",
            "Progressing": "",
            "Error": "arn:aws:sns:us-east-1:123456789012:ETS_Errors"
        },
        "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
        "InputBucket": "salesoffice.example.com-source",
        "Id": "1533765810590-example",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/1533765810590-example"
    },
    "Warnings": [
        {
            "Message": "The SNS notification topic for Error events and the pipeline
 are in different regions, which increases processing time for jobs in the pipeline
 and can incur additional charges. To decrease processing time and prevent cross-
regional charges, use the same region for the SNS notification topic and the
 pipeline.",
            "Code": "6006"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePipeline。

create-preset

以下程式碼範例顯示如何使用 create-preset。

AWS CLI

建立 ElasticTranscoder 的預設集

下列create-preset範例會建立 ElasticTranscoder 的預設集。

```
aws elastictranscoder create-preset \
    --name DefaultPreset \
    --description "Use for published videos" \
    --container mp4 \
    --video file://video.json \
    --audio file://audio.json \
    --thumbnails file://thumbnails.json
```

video.json 的內容：

```
{
    "Codec":"H.264",
    "CodecOptions":{
        "Profile":"main",
        "Level":"2.2",
        "MaxReferenceFrames":"3",
        "MaxBitRate":"",
        "BufferSize":"",
        "InterlacedMode":"Progressive",
        "ColorSpaceConversionMode":"None"
    },
    "KeyframesMaxDist":"240",
    "FixedGOP":"false",
    "BitRate":"1600",
    "FrameRate":"auto",
    "MaxFrameRate":"30",
    "MaxWidth":"auto",
    "MaxHeight":"auto",
    "SizingPolicy":"Fit",
    "PaddingPolicy":"Pad",
    "DisplayAspectRatio":"auto",
```

```
    "Watermarks":[
        {
            "Id":"company logo",
            "MaxWidth":"20%",
            "MaxHeight":"20%",
            "SizingPolicy":"ShrinkToFit",
            "HorizontalAlign":"Right",
            "HorizontalOffset":"10px",
            "VerticalAlign":"Bottom",
            "VerticalOffset":"10px",
            "Opacity":"55.5",
            "Target":"Content"
        }
    ]
}
```

audio.json 的內容：

```
{
    "Codec":"AAC",
    "CodecOptions":{
        "Profile":"AAC-LC"
    },
    "SampleRate":"44100",
    "BitRate":"96",
    "Channels":"2"
}
```

thumbnails.json 的內容：

```
{
    "Format":"png",
    "Interval":"120",
    "MaxWidth":"auto",
    "MaxHeight":"auto",
    "SizingPolicy":"Fit",
    "PaddingPolicy":"Pad"
}
```

輸出：

```
{
```

```
    "Preset": {
        "Thumbnails": {
            "SizingPolicy": "Fit",
            "MaxWidth": "auto",
            "Format": "png",
            "PaddingPolicy": "Pad",
            "Interval": "120",
            "MaxHeight": "auto"
        },
        "Container": "mp4",
        "Description": "Use for published videos",
        "Video": {
            "SizingPolicy": "Fit",
            "MaxWidth": "auto",
            "PaddingPolicy": "Pad",
            "MaxFrameRate": "30",
            "FrameRate": "auto",
            "MaxHeight": "auto",
            "KeyframesMaxDist": "240",
            "FixedGOP": "false",
            "Codec": "H.264",
            "Watermarks": [
                {
                    "SizingPolicy": "ShrinkToFit",
                    "VerticalOffset": "10px",
                    "VerticalAlign": "Bottom",
                    "Target": "Content",
                    "MaxWidth": "20%",
                    "MaxHeight": "20%",
                    "HorizontalAlign": "Right",
                    "HorizontalOffset": "10px",
                    "Opacity": "55.5",
                    "Id": "company logo"
                }
            ],
            "CodecOptions": {
                "Profile": "main",
                "MaxBitRate": "32",
                "InterlacedMode": "Progressive",
                "Level": "2.2",
                "ColorSpaceConversionMode": "None",
                "MaxReferenceFrames": "3",
                "BufferSize": "5"
            },
```

```
            "BitRate": "1600",
            "DisplayAspectRatio": "auto"
        },
        "Audio": {
            "Channels": "2",
            "CodecOptions": {
                "Profile": "AAC-LC"
            },
            "SampleRate": "44100",
            "Codec": "AAC",
            "BitRate": "96"
        },
        "Type": "Custom",
        "Id": "1533765290724-example"
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/1533765290724-example",
        "Name": "DefaultPreset"
    },
    "Warning": ""
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePreset。

delete-pipeline

以下程式碼範例顯示如何使用 delete-pipeline。

AWS CLI

刪除指定的 ElasticTranscoder 管道

這會刪除指定的 ElasticTranscoder 管道。

命令：

```
aws elastictranscoder delete-pipeline --id 1111111111111-abcde1
```

輸出：

```
{
 "Success":"true"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePipeline。

`delete-preset`

以下程式碼範例顯示如何使用 `delete-preset`。

AWS CLI

刪除指定的 ElasticTranscoder 預設集

這會刪除指定的 ElasticTranscoder 預設集。

命令：

```
aws elastictranscoder delete-preset --id 5555555555555-abcde5
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePreset。

`list-jobs-by-pipeline`

以下程式碼範例顯示如何使用 `list-jobs-by-pipeline`。

AWS CLI

在指定的管道中擷取 ElasticTranscoder 任務清單

此範例會擷取指定管道中的 ElasticTranscoder 任務清單。

命令：

```
aws elastictranscoder list-jobs-by-pipeline --pipeline-id 1111111111111-abcde1
```

輸出：

```
{
   "Jobs": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListJobsByPipeline。

`list-jobs-by-status`

以下程式碼範例顯示如何使用 `list-jobs-by-status`。

AWS CLI

擷取狀態為完成的 ElasticTranscoder 任務清單

此範例會擷取狀態為完成的 ElasticTranscoder 任務清單。

命令：

```
aws elastictranscoder list-jobs-by-status --status Complete
```

輸出：

```
{
   "Jobs": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListJobsByStatus。

`list-pipelines`

以下程式碼範例顯示如何使用 `list-pipelines`。

AWS CLI

擷取 ElasticTranscoder 管道的清單

此範例會擷取 ElasticTranscoder 管道的清單。

命令：

```
aws elastictranscoder list-pipelines
```

輸出：

```
{
   "Pipelines": [
       {
           "Status": "Active",
           "ContentConfig": {
```

```
            "Bucket": "ets-example",
            "Permissions": []
        },
        "Name": "example-pipeline",
        "ThumbnailConfig": {
            "Bucket": "ets-example",
            "Permissions": []
        },
        "Notifications": {
            "Completed": "arn:aws:sns:us-west-2:123456789012:ets_example",
            "Warning": "",
            "Progressing": "",
            "Error": ""
        },
        "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
        "InputBucket": "ets-example",
        "OutputBucket": "ets-example",
        "Id": "3333333333333-abcde3",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde3"
    },
    {
        "Status": "Paused",
        "ContentConfig": {
            "Bucket": "ets-example",
            "Permissions": []
        },
        "Name": "example-php-test",
        "ThumbnailConfig": {
            "Bucket": "ets-example",
            "Permissions": []
        },
        "Notifications": {
            "Completed": "",
            "Warning": "",
            "Progressing": "",
            "Error": ""
        },
        "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
        "InputBucket": "ets-example",
        "OutputBucket": "ets-example",
        "Id": "3333333333333-abcde2",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde2"
```

```
        },
        {
            "Status": "Active",
            "ContentConfig": {
                "Bucket": "ets-west-output",
                "Permissions": []
            },
            "Name": "pipeline-west",
            "ThumbnailConfig": {
                "Bucket": "ets-west-output",
                "Permissions": []
            },
            "Notifications": {
                "Completed": "arn:aws:sns:us-west-2:123456789012:ets-notifications",
                "Warning": "",
                "Progressing": "",
                "Error": ""
            },
            "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
            "InputBucket": "ets-west-input",
            "OutputBucket": "ets-west-output",
            "Id": "3333333333333-abcde1",
            "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde1"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPipelines。

`list-presets`

以下程式碼範例顯示如何使用 `list-presets`。

AWS CLI

擷取 ElasticTranscoder 預設集的清單

此範例會擷取 ElasticTranscoder 預設集的清單。

命令：

```
aws elastictranscoder list-presets --max-items 2
```

輸出：

```
{
  "Presets": [
      {
          "Container": "mp4",
          "Name": "KindleFireHD-preset",
          "Video": {
              "Resolution": "1280x720",
              "FrameRate": "30",
              "KeyframesMaxDist": "90",
              "FixedGOP": "false",
              "Codec": "H.264",
              "Watermarks": [],
              "CodecOptions": {
                  "Profile": "main",
                  "MaxReferenceFrames": "3",
                  "ColorSpaceConversionMode": "None",
                  "InterlacedMode": "Progressive",
                  "Level": "4"
              },
              "AspectRatio": "16:9",
              "BitRate": "2200"
          },
          "Audio": {
              "Channels": "2",
              "CodecOptions": {
                  "Profile": "AAC-LC"
              },
              "SampleRate": "48000",
              "Codec": "AAC",
              "BitRate": "160"
          },
          "Type": "Custom",
          "Id": "3333333333333-abcde2",
          "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/3333333333333-abcde2",
          "Thumbnails": {
              "AspectRatio": "16:9",
              "Interval": "60",
              "Resolution": "192x108",
              "Format": "png"
          }
      },
```

```
      {
          "Thumbnails": {
              "AspectRatio": "16:9",
              "Interval": "60",
              "Resolution": "192x108",
              "Format": "png"
          },
          "Container": "mp4",
          "Description": "Custom preset for transcoding jobs",
          "Video": {
              "Resolution": "1280x720",
              "FrameRate": "30",
              "KeyframesMaxDist": "90",
              "FixedGOP": "false",
              "Codec": "H.264",
              "Watermarks": [],
              "CodecOptions": {
                  "Profile": "main",
                  "MaxReferenceFrames": "3",
                  "ColorSpaceConversionMode": "None",
                  "InterlacedMode": "Progressive",
                  "Level": "3.1"
              },
              "AspectRatio": "16:9",
              "BitRate": "2200"
          },
          "Audio": {
              "Channels": "2",
              "CodecOptions": {
                  "Profile": "AAC-LC"
              },
              "SampleRate": "44100",
              "Codec": "AAC",
              "BitRate": "160"
          },
          "Type": "Custom",
          "Id": "3333333333333-abcde3",
          "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/3333333333333-abcde3",
          "Name": "Roman's Preset"
      }
  ],
  "NextToken": "eyJQYWdlVG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPresets。

read-job

以下程式碼範例顯示如何使用 read-job。

AWS CLI

擷取 ElasticTranscoder 任務

此範例會擷取指定的 ElasticTranscoder 任務。

命令：

```
aws elastictranscoder read-job --id 1533838012294-example
```

輸出：

```
{
   "Job": {
       "Status": "Progressing",
       "Inputs": [
           {
               "Container": "mp4",
               "FrameRate": "auto",
               "Key": "ETS_example_file.mp4",
               "AspectRatio": "auto",
               "Resolution": "auto",
               "Interlaced": "auto"
           }
       ],
       "Playlists": [],
       "Outputs": [
           {
               "Status": "Progressing",
               "Rotate": "0",
               "PresetId": "1351620000001-100250",
               "Watermarks": [],
               "Key": "webm/ETS_example_file-kindlefirehd.webm",
               "Id": "1"
```

```
            }
        ],
        "PipelineId": "3333333333333-abcde3",
        "OutputKeyPrefix": "recipes/",
        "UserMetadata": {
            "Cook book": "recipe notebook",
            "Food type": "Italian"
        },
        "Output": {
            "Status": "Progressing",
            "Rotate": "0",
            "PresetId": "1351620000001-100250",
            "Watermarks": [],
            "Key": "webm/ETS_example_file-kindlefirehd.webm",
            "Id": "1"
        },
        "Timing": {
            "SubmitTimeMillis": 1533838012298,
            "StartTimeMillis": 1533838013786
        },
        "Input": {
            "Container": "mp4",
            "FrameRate": "auto",
            "Key": "ETS_example_file.mp4",
            "AspectRatio": "auto",
            "Resolution": "auto",
            "Interlaced": "auto"
        },
        "Id": "1533838012294-example",
        "Arn": "arn:aws:elastictranscoder:us-west-2:123456789012:job/1533838012294-
example"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReadJob。

read-pipeline

以下程式碼範例顯示如何使用 read-pipeline。

AWS CLI

擷取 ElasticTranscoder 管道

此範例會擷取指定的 ElasticTranscoder 管道。

命令：

```
aws elastictranscoder read-pipeline --id 3333333333333-abcde3
```

輸出：

```
{
   "Pipeline": {
       "Status": "Active",
       "ContentConfig": {
           "Bucket": "ets-example",
           "StorageClass": "Standard",
           "Permissions": [
               {
                   "Access": [
                       "FullControl"
                   ],
                   "Grantee": "marketing-promos@example.com",
                   "GranteeType": "Email"
               }
           ]
       },
       "Name": "Default",
       "ThumbnailConfig": {
           "Bucket": "ets-example",
           "StorageClass": "ReducedRedundancy",
           "Permissions": [
               {
                   "Access": [
                       "FullControl"
                   ],
                   "Grantee": "marketing-promos@example.com",
                   "GranteeType": "Email"
               }
           ]
       },
       "Notifications": {
           "Completed": "",
           "Warning": "",
           "Progressing": "",
           "Error": "arn:aws:sns:us-east-1:123456789012:ETS_Errors"
```

```
        },
        "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
        "InputBucket": "ets-example",
        "Id": "3333333333333-abcde3",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde3"
    },
    "Warnings": [
        {
            "Message": "The SNS notification topic for Error events and the pipeline
 are in different regions, which increases processing time for jobs in the pipeline
 and can incur additional charges. To decrease processing time and prevent cross-
regional charges, use the same region for the SNS notification topic and the
 pipeline.",
            "Code": "6006"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReadPipeline。

read-preset

以下程式碼範例顯示如何使用 read-preset。

AWS CLI

擷取 ElasticTranscoder 預設集

此範例會擷取指定的 ElasticTranscoder 預設集。

命令：

```
aws elastictranscoder read-preset --id 1351620000001-500020
```

輸出：

```
{
    "Preset": {
        "Thumbnails": {
            "SizingPolicy": "ShrinkToFit",
            "MaxWidth": "192",
            "Format": "png",
```

```
            "PaddingPolicy": "NoPad",
            "Interval": "300",
            "MaxHeight": "108"
        },
        "Container": "fmp4",
        "Description": "System preset: MPEG-Dash Video - 4.8M",
        "Video": {
            "SizingPolicy": "ShrinkToFit",
            "MaxWidth": "1280",
            "PaddingPolicy": "NoPad",
            "FrameRate": "30",
            "MaxHeight": "720",
            "KeyframesMaxDist": "60",
            "FixedGOP": "true",
            "Codec": "H.264",
            "Watermarks": [
                {
                    "SizingPolicy": "ShrinkToFit",
                    "VerticalOffset": "10%",
                    "VerticalAlign": "Top",
                    "Target": "Content",
                    "MaxWidth": "10%",
                    "MaxHeight": "10%",
                    "HorizontalAlign": "Left",
                    "HorizontalOffset": "10%",
                    "Opacity": "100",
                    "Id": "TopLeft"
                },
                {
                    "SizingPolicy": "ShrinkToFit",
                    "VerticalOffset": "10%",
                    "VerticalAlign": "Top",
                    "Target": "Content",
                    "MaxWidth": "10%",
                    "MaxHeight": "10%",
                    "HorizontalAlign": "Right",
                    "HorizontalOffset": "10%",
                    "Opacity": "100",
                    "Id": "TopRight"
                },
                {
                    "SizingPolicy": "ShrinkToFit",
                    "VerticalOffset": "10%",
                    "VerticalAlign": "Bottom",
```

```
                    "Target": "Content",
                    "MaxWidth": "10%",
                    "MaxHeight": "10%",
                    "HorizontalAlign": "Left",
                    "HorizontalOffset": "10%",
                    "Opacity": "100",
                    "Id": "BottomLeft"
                },
                {
                    "SizingPolicy": "ShrinkToFit",
                    "VerticalOffset": "10%",
                    "VerticalAlign": "Bottom",
                    "Target": "Content",
                    "MaxWidth": "10%",
                    "MaxHeight": "10%",
                    "HorizontalAlign": "Right",
                    "HorizontalOffset": "10%",
                    "Opacity": "100",
                    "Id": "BottomRight"
                }
            ],
            "CodecOptions": {
                "Profile": "main",
                "MaxBitRate": "4800",
                "InterlacedMode": "Progressive",
                "Level": "3.1",
                "ColorSpaceConversionMode": "None",
                "MaxReferenceFrames": "3",
                "BufferSize": "9600"
            },
            "BitRate": "4800",
            "DisplayAspectRatio": "auto"
        },
        "Type": "System",
        "Id": "1351620000001-500020",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/1351620000001-500020",
        "Name": "System preset: MPEG-Dash Video - 4.8M"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReadPreset。

update-pipeline-notifications

以下程式碼範例顯示如何使用 update-pipeline-notifications。

AWS CLI

更新 ElasticTranscoder 管道的通知

此範例會更新指定 ElasticTranscoder 管道的通知。

命令：

```
aws elastictranscoder update-pipeline-notifications --id 1111111111111-abcde1 --notifications Progressing=arn:aws:sns:us-west-2:0123456789012:my-topic,Completed=arn:aws:sns:us-west-2:0123456789012:my-topic,Warning=arn:aws:sns:us-west-2:0123456789012:my-topic,Error=arn:aws:sns:us-east-1:111222333444:ETS_Errors
```

輸出：

```
{
   "Pipeline": {
       "Status": "Active",
       "ContentConfig": {
           "Bucket": "ets-example",
           "StorageClass": "Standard",
           "Permissions": [
               {
                   "Access": [
                       "FullControl"
                   ],
                   "Grantee": "marketing-promos@example.com",
                   "GranteeType": "Email"
               }
           ]
       },
       "Name": "Default",
       "ThumbnailConfig": {
           "Bucket": "ets-example",
           "StorageClass": "ReducedRedundancy",
           "Permissions": [
               {
                   "Access": [
                       "FullControl"
```

```
                    ],
                    "Grantee": "marketing-promos@example.com",
                    "GranteeType": "Email"
                }
            ]
        },
        "Notifications": {
            "Completed": "arn:aws:sns:us-west-2:0123456789012:my-topic",
            "Warning": "arn:aws:sns:us-west-2:0123456789012:my-topic",
            "Progressing": "arn:aws:sns:us-west-2:0123456789012:my-topic",
            "Error": "arn:aws:sns:us-east-1:111222333444:ETS_Errors"
        },
        "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
        "InputBucket": "ets-example",
        "Id": "1111111111111-abcde1",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/1111111111111-abcde1"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePipelineNotifications。

update-pipeline-status

以下程式碼範例顯示如何使用 update-pipeline-status。

AWS CLI

更新 ElasticTranscoder 管道的狀態

此範例會更新指定 ElasticTranscoder 管道的狀態。

命令：

```
aws elastictranscoder update-pipeline-status --id 1111111111111-abcde1 --
status Paused
```

輸出：

```
{
    "Pipeline": {
```

```
        "Status": "Paused",
        "ContentConfig": {
            "Bucket": "ets-example",
            "StorageClass": "Standard",
            "Permissions": [
                {
                    "Access": [
                        "FullControl"
                    ],
                    "Grantee": "marketing-promos@example.com",
                    "GranteeType": "Email"
                }
            ]
        },
        "Name": "Default",
        "ThumbnailConfig": {
            "Bucket": "ets-example",
            "StorageClass": "ReducedRedundancy",
            "Permissions": [
                {
                    "Access": [
                        "FullControl"
                    ],
                    "Grantee": "marketing-promos@example.com",
                    "GranteeType": "Email"
                }
            ]
        },
        "Notifications": {
            "Completed": "",
            "Warning": "",
            "Progressing": "",
            "Error": "arn:aws:sns:us-east-1:803981987763:ETS_Errors"
        },
        "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
        "InputBucket": "ets-example",
        "Id": "1111111111111-abcde1",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/1111111111111-abcde1"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePipelineStatus。

update-pipeline

以下程式碼範例顯示如何使用 update-pipeline。

AWS CLI

更新 ElasticTranscoder 管道

下列update-pipeline範例會更新指定的 ElasticTranscoder 管道。

```
aws elastictranscoder update-pipeline \
    --id 1111111111111-abcde1
    --name DefaultExample \
    --input-bucket salesoffice.example.com-source \
    --role arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role \
    --notifications Progressing="",Completed="",Warning="",Error=arn:aws:sns:us-
east-1:111222333444:ETS_Errors \
    --content-config file://content-config.json \
    --thumbnail-config file://thumbnail-config.json
```

content-config.json 的內容：

```
{
    "Bucket":"salesoffice.example.com-public-promos",
    "Permissions":[
        {
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
            "Access":[
                "FullControl"
            ]
        }
    ],
    "StorageClass":"Standard"
}
```

thumbnail-config.json 的內容：

```
{
    "Bucket":"salesoffice.example.com-public-promos-thumbnails",
    "Permissions":[
        {
```

```
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
            "Access":[
                "FullControl"
            ]
        }
    ],
    "StorageClass":"ReducedRedundancy"
}
```

輸出：

```
{
    "Pipeline": {
        "Status": "Active",
        "ContentConfig": {
            "Bucket": "ets-example",
            "StorageClass": "Standard",
            "Permissions": [
                {
                    "Access": [
                        "FullControl"
                    ],
                    "Grantee": "marketing-promos@example.com",
                    "GranteeType": "Email"
                }
            ]
        },
        "Name": "DefaultExample",
        "ThumbnailConfig": {
            "Bucket": "ets-example",
            "StorageClass": "ReducedRedundancy",
            "Permissions": [
                {
                    "Access": [
                        "FullControl"
                    ],
                    "Grantee": "marketing-promos@example.com",
                    "GranteeType": "Email"
                }
            ]
        },
        "Notifications": {
```

```
            "Completed": "",
            "Warning": "",
            "Progressing": "",
            "Error": "arn:aws:sns:us-east-1:111222333444:ETS_Errors"
        },
        "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
        "InputBucket": "ets-example",
        "Id": "3333333333333-abcde3",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde3"
    },
    "Warnings": [
        {
            "Message": "The SNS notification topic for Error events and the pipeline
 are in different regions, which increases processing time for jobs in the pipeline
 and can incur additional charges. To decrease processing time and prevent cross-
regional charges, use the same region for the SNS notification topic and the
 pipeline.",
            "Code": "6006"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePipeline。

使用 的 ElastiCache 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 ElastiCache 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

add-tags-to-resource

以下程式碼範例顯示如何使用 add-tags-to-resource。

AWS CLI

將標籤新增至資源

下列add-tags-to-resource範例會將最多 10 個標籤、鍵/值對新增至叢集或快照資源。

```
aws elasticache add-tags-to-resource \
    --resource-name "arn:aws:elasticache:us-east-1:1234567890:cluster:my-mem-cluster" \
    --tags '{"20150202":15, "ElastiCache":"Service"}'
```

輸出：

```
{
    "TagList": [
        {
            "Value": "20150202",
            "Key": "APIVersion"
        },
        {
            "Value": "ElastiCache",
            "Key": "Service"
        }
    ]
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用成本分配標籤監控成本。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AddTagsToResource。

authorize-cache-security-group-ingress

以下程式碼範例顯示如何使用 authorize-cache-security-group-ingress。

AWS CLI

授權用於傳入的快取安全群組

下列authorize-cache-security-group-ingress範例允許網路傳入快取安全群組。

```
aws elasticache authorize-cache-security-group-ingress \
     --cache-security-group-name  "my-sec-grp" \
     --ec2-security-group-name "my-ec2-sec-grp" \
     --ec2-security-group-owner-id "1234567890"
```

命令不會產生輸出。

如需詳細資訊，請參閱 Elasticache 使用者指南中的 Amazon ElastiCache 中的自助式更新。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AuthorizeCacheSecurityGroupIngress。

batch-apply-update-action

以下程式碼範例顯示如何使用 batch-apply-update-action。

AWS CLI

套用服務更新

下列batch-apply-update-action範例會將服務更新套用至 Redis 叢集。

```
aws elasticache batch-apply-update-action \
    --service-update-name elc-xxxxx406-xxx \
    --replication-group-ids test-cluster
```

輸出：

```
{
    "ProcessedUpdateActions": [
        {
            "ReplicationGroupId": "pat-cluster",
            "ServiceUpdateName": "elc-xxxxx406-xxx",
            "UpdateActionStatus": "waiting-to-start"
        }
    ],
    "UnprocessedUpdateActions": []
}
```

如需詳細資訊，請參閱 Elasticache 使用者指南中的 Amazon ElastiCache 中的自助式更新。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchApplyUpdateAction。

batch-stop-update-action

以下程式碼範例顯示如何使用 batch-stop-update-action。

AWS CLI

停止服務更新

下列batch-stop-update-action範例會將服務更新套用至 Redis 叢集。

```
aws elasticache batch-stop-update-action \
    --service-update-name elc-xxxxx406-xxx \
    --replication-group-ids test-cluster
```

輸出：

```
{
    "ProcessedUpdateActions": [
        {
            "ReplicationGroupId": "pat-cluster",
            "ServiceUpdateName": "elc-xxxxx406-xxx",
            "UpdateActionStatus": "stopping"
        }
    ],
    "UnprocessedUpdateActions": []
}
```

如需詳細資訊，請參閱 Elasticache 使用者指南中的 Amazon ElastiCache 中的自助式更新。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchStopUpdateAction。

copy-snapshot

以下程式碼範例顯示如何使用 copy-snapshot。

AWS CLI

複製快照

下列copy-snapshot範例會複製現有的快照。

```
aws elasticache copy-snapshot \
    --source-snapshot-name "my-snapshot" \
    --target-snapshot-name "my-snapshot-copy"
```

輸出：

```
{
    "Snapshot":{
        "Engine": "redis",
        "CacheParameterGroupName": "default.redis3.2",
        "VpcId": "vpc-3820329f3",
        "CacheClusterId": "my-redis4",
        "SnapshotRetentionLimit": 7,
        "NumCacheNodes": 1,
        "SnapshotName": "my-snapshot-copy",
        "CacheClusterCreateTime": "2016-12-21T22:24:04.955Z",
        "AutoMinorVersionUpgrade": true,
        "PreferredAvailabilityZone": "us-east-1c",
        "SnapshotStatus": "creating",
        "SnapshotSource": "manual",
        "SnapshotWindow": "07:00-08:00",
        "EngineVersion": "3.2.4",
        "NodeSnapshots": [
            {
                "CacheSize": "3 MB",
                "SnapshotCreateTime": "2016-12-28T07:00:52Z",
                "CacheNodeId": "0001",
                "CacheNodeCreateTime": "2016-12-21T22:24:04.955Z"
            }
        ],
        "CacheSubnetGroupName": "default",
        "Port": 6379,
        "PreferredMaintenanceWindow": "tue:09:30-tue:10:30",
        "CacheNodeType": "cache.m3.large"
    }
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的匯出備份。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CopySnapshot。

create-cache-cluster

以下程式碼範例顯示如何使用 create-cache-cluster。

AWS CLI

建立快取叢集

下列create-cache-cluster範例使用 Redis 引擎建立快取叢集。

```
aws elasticache create-cache-cluster \
    --cache-cluster-id "cluster-test" \
    --engine redis \
    --cache-node-type cache.m5.large \
    --num-cache-nodes 1
```

輸出：

```
{
    "CacheCluster": {
        "CacheClusterId": "cluster-test",
        "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
        "CacheNodeType": "cache.m5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.5",
        "CacheClusterStatus": "creating",
        "NumCacheNodes": 1,
        "PreferredMaintenanceWindow": "sat:13:00-sat:14:00",
        "PendingModifiedValues": {},
        "CacheSecurityGroups": [],
        "CacheParameterGroup": {
            "CacheParameterGroupName": "default.redis5.0",
            "ParameterApplyStatus": "in-sync",
            "CacheNodeIdsToReboot": []
        },
        "CacheSubnetGroupName": "default",
        "AutoMinorVersionUpgrade": true,
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "06:30-07:30",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
```

```
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的建立叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCacheCluster。

create-cache-parameter-group

以下程式碼範例顯示如何使用 create-cache-parameter-group。

AWS CLI

建立快取參數群組

下列create-cache-parameter-group範例會建立新的 Amazon ElastiCache 快取參數群組。

```
aws elasticache create-cache-parameter-group \
    --cache-parameter-group-family "redis5.0" \
    --cache-parameter-group-name "mygroup" \
    --description "mygroup"
```

輸出：

```
{
    "CacheParameterGroup": {
        "CacheParameterGroupName": "mygroup",
        "CacheParameterGroupFamily": "redis5.0",
        "Description": "my group"
    }
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的建立參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCacheParameterGroup。

create-cache-subnet-group

以下程式碼範例顯示如何使用 create-cache-subnet-group。

AWS CLI

建立快取子網路群組

下列create-cache-subnet-group範例會建立新的快取子網路群組。

```
aws elasticache create-cache-subnet-group \
    --cache-subnet-group-name "mygroup" \
    --cache-subnet-group-description "my subnet group" \
    --subnet-ids "subnet-xxxxec4f"
```

輸出：

```
{
    "CacheSubnetGroup": {
        "CacheSubnetGroupName": "mygroup",
        "CacheSubnetGroupDescription": "my subnet group",
        "VpcId": "vpc-a3e97cdb",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-xxxxec4f",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2d"
                }
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的建立快取子網路群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCacheSubnetGroup。

create-global-replication-group

以下程式碼範例顯示如何使用 create-global-replication-group。

AWS CLI

建立全域複寫群組

下列create-global-replication-group範例會建立新的全域複寫群組。

```
aws elasticache create-global-replication-group \
    --global-replication-group-id-suffix my-global-replication-group \
    --primary-replication-group-id my-primary-cluster
```

輸出：

```
{
    "GlobalReplicationGroup": {
        "GlobalReplicationGroupId": "sgaui-my-global-replication-group",
        "GlobalReplicationGroupDescription": " ",
        "Status": "creating",
        "CacheNodeType": "cache.r5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.6",
        "Members": [
            {
                "ReplicationGroupId": "my-primary-cluster",
                "ReplicationGroupRegion": "us-west-2",
                "Role": "PRIMARY",
                "AutomaticFailover": "enabled",
                "Status": "associating"
            }
        ],
        "ClusterEnabled": true,
        "GlobalNodeGroups": [
            {
                "GlobalNodeGroupId": "sgaui-my-global-replication-group-0001",
                "Slots": "0-16383"
            }
        ],
        "AuthTokenEnabled": false,
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用全域資料存放區跨 AWS 區域複寫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateGlobalReplicationGroup。

create-replication-group

以下程式碼範例顯示如何使用 create-replication-group。

AWS CLI

建立複寫群組

下列create-replication-group範例會建立 Redis （停用叢集模式） 或 Redis （啟用叢集模式） 複寫群組。此操作僅適用於 Redis。

```
aws elasticache create-replication-group \
    --replication-group-id "mygroup" \
    --replication-group-description "my group" \
    --engine "redis" \
    --cache-node-type "cache.m5.large"
```

輸出：

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "mygroup",
        "Description": "my group",
        "Status": "creating",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "mygroup-001"
        ],
        "AutomaticFailover": "disabled",
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "06:00-07:00",
        "ClusterEnabled": false,
        "CacheNodeType": "cache.m5.large",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的建立 Redis 複寫群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateReplicationGroup。

create-snapshot

以下程式碼範例顯示如何使用 create-snapshot。

AWS CLI

建立快照

下列create-snapshot範例會使用 Redis 引擎建立快照。

```
aws elasticache create-snapshot \
    --snapshot-name mysnapshot \
    --cache-cluster-id cluster-test
```

輸出：

```
{
    "Snapshot": {
        "SnapshotName": "mysnapshot",
        "CacheClusterId": "cluster-test",
        "SnapshotStatus": "creating",
        "SnapshotSource": "manual",
        "CacheNodeType": "cache.m5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.5",
        "NumCacheNodes": 1,
        "PreferredAvailabilityZone": "us-west-2b",
        "CacheClusterCreateTime": "2020-03-19T03:12:01.483Z",
        "PreferredMaintenanceWindow": "sat:13:00-sat:14:00",
        "Port": 6379,
        "CacheParameterGroupName": "default.redis5.0",
        "CacheSubnetGroupName": "default",
        "VpcId": "vpc-a3e97cdb",
        "AutoMinorVersionUpgrade": true,
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "06:30-07:30",
        "NodeSnapshots": [
            {
                "CacheNodeId": "0001",
                "CacheSize": "",
                "CacheNodeCreateTime": "2020-03-19T03:12:01.483Z"
            }
        ]
    }
}
```

如需詳細資訊，請參閱 Elasticache 使用者指南中的備份和還原 ElastiCache for Redis。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSnapshot。

create-user-group

以下程式碼範例顯示如何使用 create-user-group。

AWS CLI

建立使用者群組

下列create-user-group範例會建立新的使用者群組。

```
aws elasticache create-user-group \
    --user-group-id myusergroup \
    --engine redis \
    --user-ids default
```

輸出：

```
{
    "UserGroupId": "myusergroup",
    "Status": "creating",
    "Engine": "redis",
    "UserIds": [
        "default"
    ],
    "ReplicationGroups": [],
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用角色型存取控制 (RBAC) 驗證使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateUserGroup。

create-user

以下程式碼範例顯示如何使用 create-user。

AWS CLI

建立使用者

下列create-user範例會建立新的使用者。

```
aws elasticache create-user \
```

```
    --user-id user1 \
    --user-name myUser \
    --passwords mYnuUzrpAxXw2rdzx \
    --engine redis \
    --access-string "on ~app::* -@all +@read"
```

輸出：

```
{
    "UserId": "user2",
    "UserName": "myUser",
    "Status": "active",
    "Engine": "redis",
    "AccessString": "on ~app::* -@all +@read +@hash +@bitmap +@geo -setbit -bitfield
 -hset -hsetnx -hmset -hincrby -hincrbyfloat -hdel -bitop -geoadd -georadius -
georadiusbymember",
    "UserGroupIds": [],
    "Authentication": {
        "Type": "password",
        "PasswordCount": 1
    },
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user2"
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用角色型存取控制 (RBAC) 驗證使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateUser。

decrease-node-groups-in-global-replication-group

以下程式碼範例顯示如何使用 decrease-node-groups-in-global-replication-group。

AWS CLI

減少全域複寫群組中的節點群組數量

以下使用 Redis 引擎decrease-node-groups-in-global-replication-group減少節點群組計數。

```
aws elasticache decrease-node-groups-in-global-replication-group \
    --global-replication-group-id sgaui-test \
    --node-group-count 1 \
```

```
    --apply-immediately \
    --global-node-groups-to-retain sgaui-test-0003
```

輸出：

```
{
    "GlobalReplicationGroup":
    {
        "GlobalReplicationGroupId": "sgaui-test",
        "GlobalReplicationGroupDescription": "test",
        "Status": "modifying",
        "CacheNodeType": "cache.r5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.6",
        "Members": [
            {
                "ReplicationGroupId": "test-2",
                "ReplicationGroupRegion": "us-east-1",
                "Role": "SECONDARY",
                "AutomaticFailover": "enabled",
                "Status": "associated"
            },
            {
                "ReplicationGroupId": "test-1",
                "ReplicationGroupRegion": "us-west-2",
                "Role": "PRIMARY",
                "AutomaticFailover": "enabled",
                "Status": "associated"
            }
        ],
        "ClusterEnabled": true,
        "GlobalNodeGroups": [
            {
                "GlobalNodeGroupId": "sgaui-test-0001",
                "Slots": "0-449,1816-5461"
            },
            {
                "GlobalNodeGroupId": "sgaui-test-0002",
                "Slots": "6827-10922"
            },
            {
                "GlobalNodeGroupId": "sgaui-test-0003",
                "Slots": "10923-14052,15418-16383"
```

```
            },
            {
                "GlobalNodeGroupId": "sgaui-test-0004",
                "Slots": "450-1815,5462-6826,14053-15417"
            }
        ],
        "AuthTokenEnabled": false,
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用全域資料存放區跨 AWS 區域複寫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DecreaseNodeGroupsInGlobalReplicationGroup。

decrease-replica-count

以下程式碼範例顯示如何使用 decrease-replica-count。

AWS CLI

減少複本計數

下列decrease-replica-count範例會動態減少 Redis （停用叢集模式） 複寫群組中的複本數目，或 Redis （啟用叢集模式） 複寫群組的一或多個節點群組 （碎片） 中的複本節點數目。此操作會在沒有叢集停機時間的情況下執行。

```
aws elasticache decrease-replica-count \
    --replication-group-id my-cluster \
    --apply-immediately  \
    --new-replica-count 2
```

輸出：

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "my-cluster",
        "Description": " ",
        "Status": "modifying",
        "PendingModifiedValues": {},
        "MemberClusters": [
```

```
            "myrepliace",
            "my-cluster-001",
            "my-cluster-002",
            "my-cluster-003"
        ],
        "NodeGroups": [
            {
                "NodeGroupId": "0001",
                "Status": "modifying",
                "PrimaryEndpoint": {
                    "Address": "my-cluster.xxxxx.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "ReaderEndpoint": {
                    "Address": "my-cluster-
ro.xxxxx.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "myrepliace",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address":
 "myrepliace.xxxxx.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "replica"
                    },
                    {
                        "CacheClusterId": "my-cluster-001",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "my-
cluster-001.xxxxx.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "primary"
                    },
                    {
                        "CacheClusterId": "my-cluster-002",
                        "CacheNodeId": "0001",
```

```
                        "ReadEndpoint": {
                            "Address": "my-
cluster-002.xxxxx.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "replica"
                    },
                    {
                        "CacheClusterId": "my-cluster-003",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "my-
cluster-003.xxxxx.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "replica"
                    }
                ]
            }
        ],
        "AutomaticFailover": "disabled",
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "07:30-08:30",
        "ClusterEnabled": false,
        "CacheNodeType": "cache.r5.xlarge",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的變更複本數量。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DecreaseReplicaCount。

delete-cache-cluster

以下程式碼範例顯示如何使用 delete-cache-cluster。

AWS CLI

刪除快取叢集

下列delete-cache-cluster範例會刪除先前佈建的指定叢集。命令會刪除所有相關聯的快取節點、節點端點和叢集本身。當您從此操作收到成功回應時，Amazon ElastiCache 會立即開始刪除叢集；您無法取消或還原此操作。

此操作不適用於下列項目：

Redis （啟用叢集模式） clustersA 叢集是複寫groupA節點群組 （碎片） enabledA異地同步備份模式 來自 Redis （啟用叢集模式） 複寫groupA叢集 叢集未處於可用狀態

```
aws elasticache delete-cache-cluster \
    --cache-cluster-id "my-cluster-002"
```

輸出：

```
{
    "CacheCluster": {
        "CacheClusterId": "my-cluster-002",
        "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
        "CacheNodeType": "cache.r5.xlarge",
        "Engine": "redis",
        "EngineVersion": "5.0.5",
        "CacheClusterStatus": "deleting",
        "NumCacheNodes": 1,
        "PreferredAvailabilityZone": "us-west-2a",
        "CacheClusterCreateTime": "2019-11-26T03:35:04.546Z",
        "PreferredMaintenanceWindow": "mon:04:05-mon:05:05",
        "PendingModifiedValues": {},
        "NotificationConfiguration": {
            "TopicArn": "arn:aws:sns:us-west-x:xxxxxxx4152:My_Topic",
            "TopicStatus": "active"
        },
        "CacheSecurityGroups": [],
        "CacheParameterGroup": {
            "CacheParameterGroupName": "mygroup",
            "ParameterApplyStatus": "in-sync",
            "CacheNodeIdsToReboot": []
        },
        "CacheSubnetGroupName": "kxkxk",
        "AutoMinorVersionUpgrade": true,
        "SecurityGroups": [
            {
                "SecurityGroupId": "sg-xxxxxxxxxx9836",
```

```
                "Status": "active"
            },
            {
                "SecurityGroupId": "sg-xxxxxxxxxxxx7b",
                "Status": "active"
            }
        ],
        "ReplicationGroupId": "my-cluster",
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "07:30-08:30",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的刪除叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCacheCluster。

delete-cache-parameter-group

以下程式碼範例顯示如何使用 delete-cache-parameter-group。

AWS CLI

刪除快取參數群組

下列delete-cache-parameter-group範例會刪除指定的快取參數群組。如果快取參數群組與任何快取叢集相關聯，則無法刪除該快取參數群組。

```
aws elasticache delete-cache-parameter-group \
    --cache-parameter-group-name myparamgroup
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Elasticache 使用者指南》中的刪除參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCacheParameterGroup。

delete-cache-subnet-group

以下程式碼範例顯示如何使用 delete-cache-subnet-group。

AWS CLI

刪除快取子網路群組

下列delete-cache-subnet-group範例會刪除指定的快取子網路群組。如果快取子網路群組與任何叢集相關聯，則無法刪除該快取子網路群組。

```
aws elasticache delete-cache-subnet-group \
    --cache-subnet-group-name "mygroup"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Elasticache 使用者指南》中的刪除子網路群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCacheSubnetGroup。

delete-global-replication-group

以下程式碼範例顯示如何使用 delete-global-replication-group。

AWS CLI

刪除全域複寫群組

下列delete-global-replication-group範例會刪除新的全域複寫群組。

```
aws elasticache delete-global-replication-group \
    --global-replication-group-id my-global-replication-group \
    --retain-primary-replication-group
```

輸出：

```
{
    "GlobalReplicationGroup": {
        "GlobalReplicationGroupId": "sgaui-my-grg",
        "GlobalReplicationGroupDescription": "my-grg",
        "Status": "deleting",
        "CacheNodeType": "cache.r5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.6",
        "Members": [
            {
                "ReplicationGroupId": "my-cluster-grg",
```

```
                "ReplicationGroupRegion": "us-west-2",
                "Role": "PRIMARY",
                "AutomaticFailover": "enabled",
                "Status": "associated"
            }
        ],
        "ClusterEnabled": false,
        "AuthTokenEnabled": false,
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用全域資料存放區跨 AWS 區域複寫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteGlobalReplicationGroup。

delete-replication-group

以下程式碼範例顯示如何使用 delete-replication-group。

AWS CLI

刪除複寫群組

下列delete-replication-group範例會刪除現有的複寫群組。根據預設，此操作會刪除整個複寫群組，包括主要/主要和所有僅供讀取複本。如果複寫群組只有一個主要節點，您可以選擇性地刪除僅供讀取複本，同時透過設定 RetainPrimaryCluster=true 來保留主要節點。

當您從此操作收到成功回應時，Amazon ElastiCache 會立即開始刪除選取的資源；您無法取消或還原此操作。僅適用於 Redis。

```
aws elasticache delete-replication-group \
    --replication-group-id "mygroup"
```

輸出：

```
{
     "ReplicationGroup": {
         "ReplicationGroupId": "mygroup",
         "Description": "my group",
         "Status": "deleting",
```

```
            "PendingModifiedValues": {},
            "AutomaticFailover": "disabled",
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "06:00-07:00",
            "TransitEncryptionEnabled": false,
            "AtRestEncryptionEnabled": false
        }
    }
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteReplicationGroup。

delete-snapshot

以下程式碼範例顯示如何使用 delete-snapshot。

AWS CLI

刪除快照

下列delete-snapshot範例使用 Redis 引擎刪除快照。

```
aws elasticache delete-snapshot \
    --snapshot-name mysnapshot
```

輸出：

```
{
    "Snapshot": {
        "SnapshotName": "my-cluster-snapshot",
        "ReplicationGroupId": "mycluster",
        "ReplicationGroupDescription": "mycluster",
        "SnapshotStatus": "deleting",
        "SnapshotSource": "manual",
        "CacheNodeType": "cache.r5.xlarge",
        "Engine": "redis",
        "EngineVersion": "5.0.5",
        "PreferredMaintenanceWindow": "thu:12:00-thu:13:00",
        "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxxxxx152:My_Topic",
        "Port": 6379,
        "CacheParameterGroupName": "default.redis5.0.cluster.on",
        "CacheSubnetGroupName": "default",
        "VpcId": "vpc-a3e97cdb",
        "AutoMinorVersionUpgrade": true,
```

```
        "SnapshotRetentionLimit": 1,
        "SnapshotWindow": "13:00-14:00",
        "NumNodeGroups": 4,
        "AutomaticFailover": "enabled",
        "NodeSnapshots": [
            {
                "CacheClusterId": "mycluster-0002-003",
                "NodeGroupId": "0002",
                "CacheNodeId": "0001",
                "CacheSize": "6 MB",
                "CacheNodeCreateTime": "2020-06-18T00:05:44.719000+00:00",
                "SnapshotCreateTime": "2020-06-25T20:34:30+00:00"
            },
            {
                "CacheClusterId": "mycluster-0003-003",
                "NodeGroupId": "0003",
                "CacheNodeId": "0001",
                "CacheSize": "6 MB",
                "CacheNodeCreateTime": "2019-12-05T19:13:15.912000+00:00",
                "SnapshotCreateTime": "2020-06-25T20:34:30+00:00"
            },
            {
                "CacheClusterId": "mycluster-0004-002",
                "NodeGroupId": "0004",
                "CacheNodeId": "0001",
                "CacheSize": "6 MB",
                "CacheNodeCreateTime": "2019-12-09T19:44:34.324000+00:00",
                "SnapshotCreateTime": "2020-06-25T20:34:30+00:00"
            },
            {
                "CacheClusterId": "mycluster-0005-003",
                "NodeGroupId": "0005",
                "CacheNodeId": "0001",
                "CacheSize": "6 MB",
                "CacheNodeCreateTime": "2020-06-18T00:05:44.775000+00:00",
                "SnapshotCreateTime": "2020-06-25T20:34:30+00:00"
            }
        ]
    }
}
```

如需詳細資訊，請參閱 Elasticache 使用者指南中的備份和還原 ElastiCache for Redis。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteSnapshot。

delete-user-group

以下程式碼範例顯示如何使用 delete-user-group。

AWS CLI

刪除使用者群組

下列delete-user-group範例會刪除使用者群組。

```
aws elasticache delete-user-group \
    --user-group-id myusergroup
```

輸出：

```
{
    "UserGroupId": "myusergroup",
    "Status": "deleting",
    "Engine": "redis",
    "UserIds": [
        "default"
    ],
    "ReplicationGroups": [],
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用角色型存取控制 (RBAC) 驗證使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteUserGroup。

delete-user

以下程式碼範例顯示如何使用 delete-user。

AWS CLI

若要刪除使用者

下列delete-user範例會刪除使用者。

```
aws elasticache delete-user \
    --user-id user2
```

輸出：

```
{
    "UserId": "user1",
    "UserName": "myUser",
    "Status": "deleting",
    "Engine": "redis",
    "AccessString": "on ~* +@all",
    "UserGroupIds": [
        "myusergroup"
    ],
    "Authentication": {
        "Type": "password",
        "PasswordCount": 1
    },
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user1"
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用角色型存取控制 (RBAC) 驗證使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteUser。

describe-cache-clusters

以下程式碼範例顯示如何使用 describe-cache-clusters。

AWS CLI

描述快取叢集

下列describe-cache-clusters範例說明快取叢集。

```
aws elasticache describe-cache-clusters
```

輸出：

```
{
    "CacheClusters": [
    {
            "CacheClusterId": "my-cluster-003",
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/
elasticache/home#client-download:",
            "CacheNodeType": "cache.r5.large",
```

```
            "Engine": "redis",
            "EngineVersion": "5.0.5",
            "CacheClusterStatus": "available",
            "NumCacheNodes": 1,
            "PreferredAvailabilityZone": "us-west-2a",
            "CacheClusterCreateTime": "2019-11-26T01:22:52.396Z",
            "PreferredMaintenanceWindow": "mon:17:30-mon:18:30",
            "PendingModifiedValues": {},
            "NotificationConfiguration": {
                "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxxx152:My_Topic",
                "TopicStatus": "active"
            },
            "CacheSecurityGroups": [],
            "CacheParameterGroup": {
                "CacheParameterGroupName": "default.redis5.0",
                "ParameterApplyStatus": "in-sync",
                "CacheNodeIdsToReboot": []
            },
            "CacheSubnetGroupName": "kxkxk",
            "AutoMinorVersionUpgrade": true,
            "SecurityGroups": [
                {
                    "SecurityGroupId": "sg-xxxxxd7b",
                    "Status": "active"
                }
            ],
            "ReplicationGroupId": "my-cluster",
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "06:30-07:30",
            "AuthTokenEnabled": false,
            "TransitEncryptionEnabled": false,
            "AtRestEncryptionEnabled": false,
            "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxxx152:cluster:my-cache-
cluster",
            "ReplicationGroupLogDeliveryEnabled": false,
            "LogDeliveryConfigurations": [
                {
                    "LogType": "slow-log",
                    "DestinationType": "cloudwatch-logs",
                    "DestinationDetails": {
                        "CloudWatchLogsDetails": {
                            "LogGroup": "test-log"
                        }
                    },
```

```
                    "LogFormat": "text",
                    "Status": "active"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的管理叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCacheClusters。

describe-cache-engine-versions

以下程式碼範例顯示如何使用 describe-cache-engine-versions。

AWS CLI

描述快取引擎版本

下列describe-cache-engine-versions範例會傳回可用快取引擎及其版本的清單。

```
aws elasticache describe-cache-engine-versions \
    --engine "Redis"
```

輸出：

```
{
    "CacheEngineVersions": [
        {
            "Engine": "redis",
            "EngineVersion": "2.6.13",
            "CacheParameterGroupFamily": "redis2.6",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 2.6.13"
        },
        {
            "Engine": "redis",
            "EngineVersion": "2.8.19",
            "CacheParameterGroupFamily": "redis2.8",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 2.8.19"
        },
```

```
        {
            "Engine": "redis",
            "EngineVersion": "2.8.21",
            "CacheParameterGroupFamily": "redis2.8",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 2.8.21"
        },
        {
            "Engine": "redis",
            "EngineVersion": "2.8.22",
            "CacheParameterGroupFamily": "redis2.8",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 2.8.22"
        },
        {
            "Engine": "redis",
            "EngineVersion": "2.8.23",
            "CacheParameterGroupFamily": "redis2.8",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 2.8.23"
        },
        {
            "Engine": "redis",
            "EngineVersion": "2.8.24",
            "CacheParameterGroupFamily": "redis2.8",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 2.8.24"
        },
        {
            "Engine": "redis",
            "EngineVersion": "2.8.6",
            "CacheParameterGroupFamily": "redis2.8",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 2.8.6"
        },
        {
            "Engine": "redis",
            "EngineVersion": "3.2.10",
            "CacheParameterGroupFamily": "redis3.2",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 3.2.10"
        },
        {
            "Engine": "redis",
```

```
            "EngineVersion": "3.2.4",
            "CacheParameterGroupFamily": "redis3.2",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 3.2.4"
        },
        {
            "Engine": "redis",
            "EngineVersion": "3.2.6",
            "CacheParameterGroupFamily": "redis3.2",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 3.2.6"
        },
        {
            "Engine": "redis",
            "EngineVersion": "4.0.10",
            "CacheParameterGroupFamily": "redis4.0",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 4.0.10"
        },
        {
            "Engine": "redis",
            "EngineVersion": "5.0.0",
            "CacheParameterGroupFamily": "redis5.0",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 5.0.0"
        },
        {
            "Engine": "redis",
            "EngineVersion": "5.0.3",
            "CacheParameterGroupFamily": "redis5.0",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 5.0.3"
        },
        {
            "Engine": "redis",
            "EngineVersion": "5.0.4",
            "CacheParameterGroupFamily": "redis5.0",
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 5.0.4"
        },
        {
            "Engine": "redis",
            "EngineVersion": "5.0.5",
            "CacheParameterGroupFamily": "redis5.0",
```

```
            "CacheEngineDescription": "Redis",
            "CacheEngineVersionDescription": "redis version 5.0.5"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCacheEngineVersions。

describe-cache-parameter-groups

以下程式碼範例顯示如何使用 describe-cache-parameter-groups。

AWS CLI

描述快取參數群組

下列describe-cache-parameter-groups範例會傳回快取參數群組描述的清單。

```
aws elasticache describe-cache-parameter-groups \
    --cache-parameter-group-name "mygroup"
```

輸出：

```
{
    "CacheParameterGroups": [
        {
            "CacheParameterGroupName": "mygroup",
            "CacheParameterGroupFamily": "redis5.0",
            "Description": " "
        }
    ]
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用參數群組設定引擎參數。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCacheParameterGroups。

describe-cache-parameters

以下程式碼範例顯示如何使用 describe-cache-parameters。

AWS CLI

描述快取參數

下列 "describe-cache-parameters' 範例會傳回指定快取參數群組的詳細參數清單。

```
aws elasticache describe-cache-parameters \
    --cache-parameter-group-name "myparamgroup"
```

輸出：

```
{
    "Parameters": [
        {
            "ParameterName": "activedefrag",
            "ParameterValue": "yes",
            "Description": "Enabled active memory defragmentation",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "active-defrag-cycle-max",
            "ParameterValue": "75",
            "Description": "Maximal effort for defrag in CPU percentage",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-75",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "active-defrag-cycle-min",
            "ParameterValue": "5",
            "Description": "Minimal effort for defrag in CPU percentage",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-75",
            "IsModifiable": true,
```

```
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "active-defrag-ignore-bytes",
            "ParameterValue": "104857600",
            "Description": "Minimum amount of fragmentation waste to start active
defrag",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1048576-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "active-defrag-max-scan-fields",
            "ParameterValue": "1000",
            "Description": "Maximum number of set/hash/zset/list fields that will be
processed from the main dictionary scan",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-1000000",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "active-defrag-threshold-lower",
            "ParameterValue": "10",
            "Description": "Minimum percentage of fragmentation to start active
defrag",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-100",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "active-defrag-threshold-upper",
            "ParameterValue": "100",
            "Description": "Maximum percentage of fragmentation at which we use
maximum effort",
```

```
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-100",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "activerehashing",
            "ParameterValue": "yes",
            "Description": "Apply rehashing or not.",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "requires-reboot"
        },
        {
            "ParameterName": "appendfsync",
            "ParameterValue": "everysec",
            "Description": "fsync policy for AOF persistence",
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "always,everysec,no",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "appendonly",
            "ParameterValue": "no",
            "Description": "Enable Redis persistence.",
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-normal-hard-limit",
            "ParameterValue": "0",
            "Description": "Normal client output buffer hard limit in bytes.",
```

```
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-normal-soft-limit",
            "ParameterValue": "0",
            "Description": "Normal client output buffer soft limit in bytes.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-normal-soft-seconds",
            "ParameterValue": "0",
            "Description": "Normal client output buffer soft limit in seconds.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-pubsub-hard-limit",
            "ParameterValue": "33554432",
            "Description": "Pubsub client output buffer hard limit in bytes.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-pubsub-soft-limit",
            "ParameterValue": "8388608",
            "Description": "Pubsub client output buffer soft limit in bytes.",
```

```
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-pubsub-soft-seconds",
            "ParameterValue": "60",
            "Description": "Pubsub client output buffer soft limit in seconds.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-replica-soft-seconds",
            "ParameterValue": "60",
            "Description": "Replica client output buffer soft limit in seconds.",
            "Source": "system",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-query-buffer-limit",
            "ParameterValue": "1073741824",
            "Description": "Max size of a single client query buffer",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1048576-1073741824",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "close-on-replica-write",
            "ParameterValue": "yes",
```

```
            "Description": "If enabled, clients who attempt to write to a read-only
replica will be disconnected. Applicable to 2.8.23 and higher.",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "cluster-enabled",
            "ParameterValue": "no",
            "Description": "Enable cluster mode",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "requires-reboot"
        },
        {
            "ParameterName": "cluster-require-full-coverage",
            "ParameterValue": "no",
            "Description": "Whether cluster becomes unavailable if one or more slots
are not covered",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "databases",
            "ParameterValue": "16",
            "Description": "Set the number of databases.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-1200000",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "requires-reboot"
        },
        {
```

```
            "ParameterName": "hash-max-ziplist-entries",
            "ParameterValue": "512",
            "Description": "The maximum number of hash entries in order for the
dataset to be compressed.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "hash-max-ziplist-value",
            "ParameterValue": "64",
            "Description": "The threshold of biggest hash entries in order for the
dataset to be compressed.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "hll-sparse-max-bytes",
            "ParameterValue": "3000",
            "Description": "HyperLogLog sparse representation bytes limit",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-16000",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lazyfree-lazy-eviction",
            "ParameterValue": "no",
            "Description": "Perform an asynchronous delete on evictions",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
```

```
        },
        {
            "ParameterName": "lazyfree-lazy-expire",
            "ParameterValue": "no",
            "Description": "Perform an asynchronous delete on expired keys",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lazyfree-lazy-server-del",
            "ParameterValue": "no",
            "Description": "Perform an asynchronous delete on key updates",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lfu-decay-time",
            "ParameterValue": "1",
            "Description": "The amount of time in minutes to decrement the key
counter for LFU eviction policy",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lfu-log-factor",
            "ParameterValue": "10",
            "Description": "The log factor for incrementing key counter for LFU
eviction policy",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-",
            "IsModifiable": true,
```

```
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "list-compress-depth",
            "ParameterValue": "0",
            "Description": "Number of quicklist ziplist nodes from each side of
the list to exclude from compression. The head and tail of the list are always
uncompressed for fast push/pop operations",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "list-max-ziplist-size",
            "ParameterValue": "-2",
            "Description": "The number of entries allowed per internal list node can
be specified as a fixed maximum size or a maximum number of elements",
            "Source": "system",
            "DataType": "integer",
            "AllowedValues": "-5,-4,-3,-2,-1,1-",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lua-replicate-commands",
            "ParameterValue": "yes",
            "Description": "Always enable Lua effect replication or not",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lua-time-limit",
            "ParameterValue": "5000",
            "Description": "Max execution time of a Lua script in milliseconds. 0
for unlimited execution without warnings.",
```

```
            "Source": "system",
            "DataType": "integer",
            "AllowedValues": "5000",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "maxclients",
            "ParameterValue": "65000",
            "Description": "The maximum number of Redis clients.",
            "Source": "system",
            "DataType": "integer",
            "AllowedValues": "1-65000",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "requires-reboot"
        },
        {
            "ParameterName": "maxmemory-policy",
            "ParameterValue": "volatile-lru",
            "Description": "Max memory policy.",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "volatile-lru,allkeys-lru,volatile-lfu,allkeys-
lfu,volatile-random,allkeys-random,volatile-ttl,noeviction",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "maxmemory-samples",
            "ParameterValue": "3",
            "Description": "Max memory samples.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "min-replicas-max-lag",
            "ParameterValue": "10",
```

```
            "Description": "The maximum amount of replica lag in seconds beyond
which the master would stop taking writes. A value of 0 means the master always
takes writes.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "min-replicas-to-write",
            "ParameterValue": "0",
            "Description": "The minimum number of replicas that must be present with
lag no greater than min-replicas-max-lag for master to take writes. Setting this to
0 means the master always takes writes.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "notify-keyspace-events",
            "Description": "The keyspace events for Redis to notify Pub/Sub clients
about. By default all notifications are disabled",
            "Source": "user",
            "DataType": "string",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "proto-max-bulk-len",
            "ParameterValue": "536870912",
            "Description": "Max size of a single element request",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1048576-536870912",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
```

```
        {
            "ParameterName": "rename-commands",
            "ParameterValue": "",
            "Description": "Redis commands that can be dynamically renamed by the
customer",
            "Source": "user",
            "DataType": "string",
            "AllowedValues":
"APPEND,BITCOUNT,BITFIELD,BITOP,BITPOS,BLPOP,BRPOP,BRPOPLPUSH,BZPOPMIN,BZPOPMAX,CLIENT,COMM
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.3",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "repl-backlog-size",
            "ParameterValue": "1048576",
            "Description": "The replication backlog size in bytes for PSYNC. This is
the size of the buffer which accumulates slave data when slave is disconnected for
some time, so that when slave reconnects again, only transfer the portion of data
which the slave missed. Minimum value is 16K.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "16384-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "repl-backlog-ttl",
            "ParameterValue": "3600",
            "Description": "The amount of time in seconds after the master no longer
have any slaves connected for the master to free the replication backlog. A value
of 0 means to never release the backlog.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "replica-allow-chaining",
            "ParameterValue": "no",
            "Description": "Configures if chaining of replicas is allowed",
```

```
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "replica-ignore-maxmemory",
            "ParameterValue": "yes",
            "Description": "Determines if replica ignores maxmemory setting by not
evicting items independent from the master",
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "replica-lazy-flush",
            "ParameterValue": "no",
            "Description": "Perform an asynchronous flushDB during replica sync",
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "reserved-memory-percent",
            "ParameterValue": "25",
            "Description": "The percent of memory reserved for non-cache memory
usage. You may want to increase this parameter for nodes with read replicas, AOF
enabled, etc, to reduce swap usage.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-100",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
```

```
            "ParameterName": "set-max-intset-entries",
            "ParameterValue": "512",
            "Description": "The limit in the size of the set in order for the
dataset to be compressed.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "slowlog-log-slower-than",
            "ParameterValue": "10000",
            "Description": "The execution time, in microseconds, to exceed in order
for the command to get logged. Note that a negative number disables the slow log,
while a value of zero forces the logging of every command.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "slowlog-max-len",
            "ParameterValue": "128",
            "Description": "The length of the slow log. There is no limit to this
length. Just be aware that it will consume memory. You can reclaim memory used by
the slow log with SLOWLOG RESET.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "stream-node-max-bytes",
            "ParameterValue": "4096",
            "Description": "The maximum size of a single node in a stream in bytes",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
```

```
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "stream-node-max-entries",
            "ParameterValue": "100",
            "Description": "The maximum number of items a single node in a stream
can contain",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "tcp-keepalive",
            "ParameterValue": "300",
            "Description": "If non-zero, send ACKs every given number of seconds.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "timeout",
            "ParameterValue": "0",
            "Description": "Close connection if client is idle for a given number of
seconds, or never if 0.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0,20-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "zset-max-ziplist-entries",
            "ParameterValue": "128",
            "Description": "The maximum number of sorted set entries in order for
the dataset to be compressed.",
```

```
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "zset-max-ziplist-value",
            "ParameterValue": "64",
            "Description": "The threshold of biggest sorted set entries in order for
 the dataset to be compressed.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        }
    ]
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的參數管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCacheParameters。

describe-cache-subnet-groups

以下程式碼範例顯示如何使用 describe-cache-subnet-groups。

AWS CLI

描述快取子網路群組

下列describe-cache-subnet-groups範例會傳回子網路群組的清單。

```
aws elasticache describe-cache-subnet-groups
```

輸出：

```
{
    "CacheSubnetGroups": [
        {
```

```
            "CacheSubnetGroupName": "default",
            "CacheSubnetGroupDescription": "Default CacheSubnetGroup",
            "VpcId": "vpc-a3e97cdb",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-8d4bacf5",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    }
                },
                {
                    "SubnetIdentifier": "subnet-dde21380",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    }
                },
                {
                    "SubnetIdentifier": "subnet-6485ec4f",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    }
                },
                {
                    "SubnetIdentifier": "subnet-b4ebebff",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    }
                }
            ]
        },
        {
            "CacheSubnetGroupName": "kxkxk",
            "CacheSubnetGroupDescription": "mygroup",
            "VpcId": "vpc-a3e97cdb",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-b4ebebff",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    }
                }
            ]
        },
        {
```

```
            "CacheSubnetGroupName": "test",
            "CacheSubnetGroupDescription": "test",
            "VpcId": "vpc-a3e97cdb",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-b4ebebff",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    }
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱 Elasticache 使用者指南中的子網路和子網路群組，或 ElastiCache for Memcached 使用者指南中的子網路和子網路群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCacheSubnetGroups。

describe-engine-default-parameters

以下程式碼範例顯示如何使用 describe-engine-default-parameters。

AWS CLI

描述引擎預設參數

下列describe-engine-default-parameters範例會傳回指定快取引擎的預設引擎和系統參數資訊。

```
aws elasticache describe-engine-default-parameters \
    --cache-parameter-group-family "redis5.0"
```

輸出：

```
{
    "EngineDefaults": {
        "Parameters": [
            {
                "ParameterName": "activedefrag",
                "ParameterValue": "no",
                "Description": "Enabled active memory defragmentation",
```

```
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "active-defrag-cycle-max",
                "ParameterValue": "75",
                "Description": "Maximal effort for defrag in CPU percentage",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-75",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "active-defrag-cycle-min",
                "ParameterValue": "5",
                "Description": "Minimal effort for defrag in CPU percentage",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-75",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "active-defrag-ignore-bytes",
                "ParameterValue": "104857600",
                "Description": "Minimum amount of fragmentation waste to start
active defrag",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1048576-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "active-defrag-max-scan-fields",
                "ParameterValue": "1000",
```

```
                "Description": "Maximum number of set/hash/zset/list fields that
will be processed from the main dictionary scan",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-1000000",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "active-defrag-threshold-lower",
                "ParameterValue": "10",
                "Description": "Minimum percentage of fragmentation to start active
defrag",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-100",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "active-defrag-threshold-upper",
                "ParameterValue": "100",
                "Description": "Maximum percentage of fragmentation at which we use
maximum effort",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-100",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "activerehashing",
                "ParameterValue": "yes",
                "Description": "Apply rehashing or not.",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "requires-reboot"
            },
```

```
            {
                "ParameterName": "appendfsync",
                "ParameterValue": "everysec",
                "Description": "fsync policy for AOF persistence",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "always,everysec,no",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "appendonly",
                "ParameterValue": "no",
                "Description": "Enable Redis persistence.",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "client-output-buffer-limit-normal-hard-limit",
                "ParameterValue": "0",
                "Description": "Normal client output buffer hard limit in bytes.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "client-output-buffer-limit-normal-soft-limit",
                "ParameterValue": "0",
                "Description": "Normal client output buffer soft limit in bytes.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
```

```
            {
                "ParameterName": "client-output-buffer-limit-normal-soft-seconds",
                "ParameterValue": "0",
                "Description": "Normal client output buffer soft limit in seconds.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "client-output-buffer-limit-pubsub-hard-limit",
                "ParameterValue": "33554432",
                "Description": "Pubsub client output buffer hard limit in bytes.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "client-output-buffer-limit-pubsub-soft-limit",
                "ParameterValue": "8388608",
                "Description": "Pubsub client output buffer soft limit in bytes.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "client-output-buffer-limit-pubsub-soft-seconds",
                "ParameterValue": "60",
                "Description": "Pubsub client output buffer soft limit in seconds.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
```

```
            {
                "ParameterName": "client-output-buffer-limit-replica-soft-seconds",
                "ParameterValue": "60",
                "Description": "Replica client output buffer soft limit in
 seconds.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "client-query-buffer-limit",
                "ParameterValue": "1073741824",
                "Description": "Max size of a single client query buffer",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1048576-1073741824",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "close-on-replica-write",
                "ParameterValue": "yes",
                "Description": "If enabled, clients who attempt to write to a read-
only replica will be disconnected. Applicable to 2.8.23 and higher.",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "cluster-enabled",
                "ParameterValue": "no",
                "Description": "Enable cluster mode",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
```

```
                "ChangeType": "requires-reboot"
            },
            {
                "ParameterName": "cluster-require-full-coverage",
                "ParameterValue": "no",
                "Description": "Whether cluster becomes unavailable if one or more
slots are not covered",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "databases",
                "ParameterValue": "16",
                "Description": "Set the number of databases.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-1200000",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "requires-reboot"
            },
            {
                "ParameterName": "hash-max-ziplist-entries",
                "ParameterValue": "512",
                "Description": "The maximum number of hash entries in order for the
dataset to be compressed.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "hash-max-ziplist-value",
                "ParameterValue": "64",
                "Description": "The threshold of biggest hash entries in order for
the dataset to be compressed.",
                "Source": "system",
                "DataType": "integer",
```

```
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "hll-sparse-max-bytes",
                "ParameterValue": "3000",
                "Description": "HyperLogLog sparse representation bytes limit",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-16000",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "lazyfree-lazy-eviction",
                "ParameterValue": "no",
                "Description": "Perform an asynchronous delete on evictions",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "lazyfree-lazy-expire",
                "ParameterValue": "no",
                "Description": "Perform an asynchronous delete on expired keys",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "lazyfree-lazy-server-del",
                "ParameterValue": "no",
                "Description": "Perform an asynchronous delete on key updates",
                "Source": "system",
                "DataType": "string",
```

```
                "AllowedValues": "yes,no",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "lfu-decay-time",
                "ParameterValue": "1",
                "Description": "The amount of time in minutes to decrement the key
counter for LFU eviction policy",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "lfu-log-factor",
                "ParameterValue": "10",
                "Description": "The log factor for incrementing key counter for LFU
eviction policy",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "list-compress-depth",
                "ParameterValue": "0",
                "Description": "Number of quicklist ziplist nodes from each side
of the list to exclude from compression. The head and tail of the list are always
uncompressed for fast push/pop operations",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "list-max-ziplist-size",
```

```
                "ParameterValue": "-2",
                "Description": "The number of entries allowed per internal list node
can be specified as a fixed maximum size or a maximum number of elements",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "-5,-4,-3,-2,-1,1-",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "lua-replicate-commands",
                "ParameterValue": "yes",
                "Description": "Always enable Lua effect replication or not",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "lua-time-limit",
                "ParameterValue": "5000",
                "Description": "Max execution time of a Lua script in milliseconds.
0 for unlimited execution without warnings.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "5000",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "maxclients",
                "ParameterValue": "65000",
                "Description": "The maximum number of Redis clients.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-65000",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "requires-reboot"
            },
```

```
            {
                "ParameterName": "maxmemory-policy",
                "ParameterValue": "volatile-lru",
                "Description": "Max memory policy.",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "volatile-lru,allkeys-lru,volatile-lfu,allkeys-
lfu,volatile-random,allkeys-random,volatile-ttl,noeviction",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "maxmemory-samples",
                "ParameterValue": "3",
                "Description": "Max memory samples.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "min-replicas-max-lag",
                "ParameterValue": "10",
                "Description": "The maximum amount of replica lag in seconds beyond
 which the master would stop taking writes. A value of 0 means the master always
 takes writes.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "min-replicas-to-write",
                "ParameterValue": "0",
                "Description": "The minimum number of replicas that must be present
 with lag no greater than min-replicas-max-lag for master to take writes. Setting
 this to 0 means the master always takes writes.",
                "Source": "system",
                "DataType": "integer",
```

```
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "notify-keyspace-events",
                "Description": "The keyspace events for Redis to notify Pub/Sub
clients about. By default all notifications are disabled",
                "Source": "system",
                "DataType": "string",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "proto-max-bulk-len",
                "ParameterValue": "536870912",
                "Description": "Max size of a single element request",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "1048576-536870912",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "rename-commands",
                "ParameterValue": "",
                "Description": "Redis commands that can be dynamically renamed by
the customer",
                "Source": "system",
                "DataType": "string",
                "AllowedValues":
"APPEND,BITCOUNT,BITFIELD,BITOP,BITPOS,BLPOP,BRPOP,BRPOPLPUSH,BZPOPMIN,BZPOPMAX,CLIENT,COMM
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.3",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "repl-backlog-size",
                "ParameterValue": "1048576",
                "Description": "The replication backlog size in bytes for PSYNC.
This is the size of the buffer which accumulates slave data when slave is
```

```
disconnected for some time, so that when slave reconnects again, only transfer the
portion of data which the slave missed. Minimum value is 16K.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "16384-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "repl-backlog-ttl",
                "ParameterValue": "3600",
                "Description": "The amount of time in seconds after the master no
longer have any slaves connected for the master to free the replication backlog. A
value of 0 means to never release the backlog.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "replica-allow-chaining",
                "ParameterValue": "no",
                "Description": "Configures if chaining of replicas is allowed",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "replica-ignore-maxmemory",
                "ParameterValue": "yes",
                "Description": "Determines if replica ignores maxmemory setting by
not evicting items independent from the master",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
```

```
            },
            {
                "ParameterName": "replica-lazy-flush",
                "ParameterValue": "no",
                "Description": "Perform an asynchronous flushDB during replica
sync",
                "Source": "system",
                "DataType": "string",
                "AllowedValues": "yes,no",
                "IsModifiable": false,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "reserved-memory-percent",
                "ParameterValue": "25",
                "Description": "The percent of memory reserved for non-cache memory
usage. You may want to increase this parameter for nodes with read replicas, AOF
enabled, etc, to reduce swap usage.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-100",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "set-max-intset-entries",
                "ParameterValue": "512",
                "Description": "The limit in the size of the set in order for the
dataset to be compressed.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "slowlog-log-slower-than",
                "ParameterValue": "10000",
                "Description": "The execution time, in microseconds, to exceed in
order for the command to get logged. Note that a negative number disables the slow
log, while a value of zero forces the logging of every command.",
```

```
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "slowlog-max-len",
                "ParameterValue": "128",
                "Description": "The length of the slow log. There is no limit to
this length. Just be aware that it will consume memory. You can reclaim memory used
by the slow log with SLOWLOG RESET.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "stream-node-max-bytes",
                "ParameterValue": "4096",
                "Description": "The maximum size of a single node in a stream in
bytes",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "stream-node-max-entries",
                "ParameterValue": "100",
                "Description": "The maximum number of items a single node in a
stream can contain",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
```

```
            {
                "ParameterName": "tcp-keepalive",
                "ParameterValue": "300",
                "Description": "If non-zero, send ACKs every given number of
seconds.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "timeout",
                "ParameterValue": "0",
                "Description": "Close connection if client is idle for a given
number of seconds, or never if 0.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0,20-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "zset-max-ziplist-entries",
                "ParameterValue": "128",
                "Description": "The maximum number of sorted set entries in order
for the dataset to be compressed.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            },
            {
                "ParameterName": "zset-max-ziplist-value",
                "ParameterValue": "64",
                "Description": "The threshold of biggest sorted set entries in order
for the dataset to be compressed.",
                "Source": "system",
                "DataType": "integer",
                "AllowedValues": "0-",
```

```
                "IsModifiable": true,
                "MinimumEngineVersion": "5.0.0",
                "ChangeType": "immediate"
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEngineDefaultParameters。

describe-events

以下程式碼範例顯示如何使用 describe-events。

AWS CLI

描述複寫群組的事件

下列describe-events範例會傳回複寫群組的事件清單。

```
aws elasticache describe-events \
    --source-identifier test-cluster \
    --source-type replication-group
```

輸出：

```
{
    "Events": [
        {
            "SourceIdentifier": "test-cluster",
            "SourceType": "replication-group",
            "Message": "Automatic failover has been turned on for replication group
 test-cluster",
            "Date": "2020-03-18T23:51:34.457Z"
        },
        {
            "SourceIdentifier": "test-cluster",
            "SourceType": "replication-group",
            "Message": "Replication group test-cluster created",
            "Date": "2020-03-18T23:50:31.378Z"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的監控事件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEvents。

describe-global-replication-groups

以下程式碼範例顯示如何使用 describe-global-replication-groups。

AWS CLI

描述全域複寫群組

下列describe-global-replication-groups範例會傳回全域資料存放區的詳細資訊。

```
aws elasticache describe-global-replication-groups \
    --global-replication-group-id my-grg
```

輸出：

```
{
    "GlobalReplicationGroups": [
        {
            "GlobalReplicationGroupId": "my-grg",
            "GlobalReplicationGroupDescription": "my-grg",
            "Status": "creating",
            "CacheNodeType": "cache.r5.large",
            "Engine": "redis",
            "EngineVersion": "5.0.6",
            "ClusterEnabled": false,
            "AuthTokenEnabled": false,
            "TransitEncryptionEnabled": false,
            "AtRestEncryptionEnabled": false
        }
    ]
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用全域資料存放區跨 AWS 區域複寫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeGlobalReplicationGroups。

describe-replication-groups

以下程式碼範例顯示如何使用 describe-replication-groups。

AWS CLI

傳回複寫群組詳細資訊的清單

下列describe-replication-groups範例會傳回複寫群組。

```
aws elasticache describe-replication-groups
```

輸出：

```
{
    "ReplicationGroups": [
        {
            "ReplicationGroupId": "my-cluster",
            "Description": "mycluster",
            "Status": "available",
            "PendingModifiedValues": {},
            "MemberClusters": [
                "pat-cluster-001",
                "pat-cluster-002",
                "pat-cluster-003",
                "pat-cluster-004"
            ],
            "NodeGroups": [
                {
                    "NodeGroupId": "0001",
                    "Status": "available",
                    "PrimaryEndpoint": {
                        "Address": "my-
cluster.xxxxih.ng.0001.usw2.cache.amazonaws.com",
                        "Port": 6379
                    },
                    "ReaderEndpoint": {
                        "Address": "my-cluster-
ro.xxxxih.ng.0001.usw2.cache.amazonaws.com",
                        "Port": 6379
                    },
                    "NodeGroupMembers": [
                        {
                            "CacheClusterId": "my-cluster-001",
```

```
                            "CacheNodeId": "0001",
                            "ReadEndpoint": {
                                "Address": "pat-
cluster-001.xxxih.0001.usw2.cache.amazonaws.com",
                                "Port": 6379
                            },
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CurrentRole": "primary"
                        },
                        {
                            "CacheClusterId": "my-cluster-002",
                            "CacheNodeId": "0001",
                            "ReadEndpoint": {
                                "Address": "pat-
cluster-002.xxxxih.0001.usw2.cache.amazonaws.com",
                                "Port": 6379
                            },
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CurrentRole": "replica"
                        },
                        {
                            "CacheClusterId": "my-cluster-003",
                            "CacheNodeId": "0001",
                            "ReadEndpoint": {
                                "Address": "pat-
cluster-003.xxxxih.0001.usw2.cache.amazonaws.com",
                                "Port": 6379
                            },
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CurrentRole": "replica"
                        },
                        {
                            "CacheClusterId": "my-cluster-004",
                            "CacheNodeId": "0001",
                            "ReadEndpoint": {
                                "Address": "pat-
cluster-004.xxxih.0001.usw2.cache.amazonaws.com",
                                "Port": 6379
                            },
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CurrentRole": "replica"
                        }
                    ]
                }
```

```
            ],
            "AutomaticFailover": "disabled",
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "07:30-08:30",
            "ClusterEnabled": false,
            "CacheNodeType": "cache.r5.xlarge",
            "AuthTokenEnabled": false,
            "TransitEncryptionEnabled": false,
            "AtRestEncryptionEnabled": false,
            "ARN": "arn:aws:elasticache:us-
west-2:xxxxxxxxxxx152:replicationgroup:my-cluster",
            "LogDeliveryConfigurations": [
                {
                    "LogType": "slow-log",
                    "DestinationType": "cloudwatch-logs",
                    "DestinationDetails": {
                        "CloudWatchLogsDetails": {
                            "LogGroup": "test-log"
                        }
                    },
                    "LogFormat": "json",
                    "Status": "active"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的管理叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReplicationGroups。

describe-reserved-cache-nodes-offerings

以下程式碼範例顯示如何使用 describe-reserved-cache-nodes-offerings。

AWS CLI

描述 reserved-cache-nodes-offerings

下列describe-reserved-cache-nodes-offerings範例會傳回 reserved-cache-node 選項的詳細資訊。

```
aws elasticache describe-reserved-cache-nodes-offerings
```

輸出：

```
{
    "ReservedCacheNodesOfferings": [
        {
            "ReservedCacheNodesOfferingId": "01ce0a19-a476-41cb-8aee-48eacbcdc8e5",
            "CacheNodeType": "cache.t3.small",
            "Duration": 31536000,
            "FixedPrice": 97.0,
            "UsagePrice": 0.0,
            "ProductDescription": "memcached",
            "OfferingType": "Partial Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": 0.011,
                    "RecurringChargeFrequency": "Hourly"
                }
            ]
        },
        {
            "ReservedCacheNodesOfferingId": "0443a27b-4da5-4b90-b92d-929fbd7abed2",
            "CacheNodeType": "cache.m3.2xlarge",
            "Duration": 31536000,
            "FixedPrice": 1772.0,
            "UsagePrice": 0.0,
            "ProductDescription": "redis",
            "OfferingType": "Heavy Utilization",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": 0.25,
                    "RecurringChargeFrequency": "Hourly"
                }
            ]
        },

        ...

    ]
}
```

如需詳細資訊，請參閱《Elasticache Redis 使用者指南》中的取得預留節點方案的相關資訊，或《Elasticache Memcached 使用者指南》中的取得預留節點方案的相關資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReservedCacheNodesOfferings。

`describe-reserved-cache-nodes`

以下程式碼範例顯示如何使用 `describe-reserved-cache-nodes`。

AWS CLI

描述預留快取節點

下列`describe-reserved-cache-nodes`範例會傳回此帳戶預留快取節點，或指定預留快取節點的相關資訊。

aws elasticache describe-reserved-cache-nodes

輸出：

```
{
    "ReservedCacheNodes": [
        {
            "ReservedCacheNodeId": "mynode",
            "ReservedCacheNodesOfferingId": "xxxxxxxxx-xxxxx-xxxxx-xxxx-xxxxxxxx71",
            "CacheNodeType": "cache.t3.small",
            "StartTime": "2019-12-06T02:50:44.003Z",
            "Duration": 31536000,
            "FixedPrice": 0.0,
            "UsagePrice": 0.0,
            "CacheNodeCount": 1,
            "ProductDescription": "redis",
            "OfferingType": "No Upfront",
            "State": "payment-pending",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": 0.023,
                    "RecurringChargeFrequency": "Hourly"
                }
            ],
            "ReservationARN": "arn:aws:elasticache:us-
west-2:xxxxxxxxxxx52:reserved-instance:mynode"
        }
```

```
    ]
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用預留節點管理成本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReservedCacheNodes。

describe-service-updates

以下程式碼範例顯示如何使用 describe-service-updates。

AWS CLI

描述服務更新

下列describe-service-updates範例會傳回服務更新的詳細資訊。

```
aws elasticache describe-service-updates
```

輸出：

```
{
     "ServiceUpdates": [
         {
             "ServiceUpdateName": "elc-xxxxxxxx7-001",
             "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
             "ServiceUpdateEndDate": "2020-02-09T15:59:59Z",
             "ServiceUpdateSeverity": "important",
             "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
             "ServiceUpdateStatus": "available",
             "ServiceUpdateDescription": "Upgrades to improve the security,
 reliability, and operational performance of your ElastiCache nodes",
             "ServiceUpdateType": "security-update",
             "Engine": "redis, memcached",
             "EngineVersion": "redis 2.6.13 and onwards, memcached 1.4.5 and
 onwards",
             "AutoUpdateAfterRecommendedApplyByDate": false,
             "EstimatedUpdateTime": "30 minutes per node"
         },
         {
             "ServiceUpdateName": "elc-xxxxxxxx4-001",
             "ServiceUpdateReleaseDate": "2019-06-11T15:00:00Z",
             "ServiceUpdateEndDate": "2019-10-01T09:24:00Z",
```

```
            "ServiceUpdateSeverity": "important",
            "ServiceUpdateRecommendedApplyByDate": "2019-07-11T14:59:59Z",
            "ServiceUpdateStatus": "expired",
            "ServiceUpdateDescription": "Upgrades to improve the security,
 reliability, and operational performance of your ElastiCache nodes",
            "ServiceUpdateType": "security-update",
            "Engine": "redis",
            "EngineVersion": "redis 3.2.6, redis 4.0 and onwards",
            "AutoUpdateAfterRecommendedApplyByDate": false,
            "EstimatedUpdateTime": "30 minutes per node"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeServiceUpdates。

`describe-snapshots`

以下程式碼範例顯示如何使用 `describe-snapshots`。

AWS CLI

描述快照

下列「describe-snapshots」範例會傳回叢集或複寫群組快照的相關資訊。

```
aws elasticache describe-snapshots
```

輸出：

```
{
    "Snapshots": [
        {
            "SnapshotName": "automatic.my-cluster2-002-2019-12-05-06-38",
            "CacheClusterId": "my-cluster2-002",
            "SnapshotStatus": "available",
            "SnapshotSource": "automated",
            "CacheNodeType": "cache.r5.large",
            "Engine": "redis",
            "EngineVersion": "5.0.5",
            "NumCacheNodes": 1,
            "PreferredAvailabilityZone": "us-west-2a",
            "CacheClusterCreateTime": "2019-11-26T01:22:52.396Z",
```

```
            "PreferredMaintenanceWindow": "mon:17:30-mon:18:30",
            "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxx52:My_Topic",
            "Port": 6379,
            "CacheParameterGroupName": "default.redis5.0",
            "CacheSubnetGroupName": "kxkxk",
            "VpcId": "vpc-a3e97cdb",
            "AutoMinorVersionUpgrade": true,
            "SnapshotRetentionLimit": 1,
            "SnapshotWindow": "06:30-07:30",
            "NodeSnapshots": [
                {
                    "CacheNodeId": "0001",
                    "CacheSize": "5 MB",
                    "CacheNodeCreateTime": "2019-11-26T01:22:52.396Z",
                    "SnapshotCreateTime": "2019-12-05T06:38:23Z"
                }
            ]
        },
        {
            "SnapshotName": "myreplica-backup",
            "CacheClusterId": "myreplica",
            "SnapshotStatus": "available",
            "SnapshotSource": "manual",
            "CacheNodeType": "cache.r5.large",
            "Engine": "redis",
            "EngineVersion": "5.0.5",
            "NumCacheNodes": 1,
            "PreferredAvailabilityZone": "us-west-2a",
            "CacheClusterCreateTime": "2019-11-26T00:14:52.439Z",
            "PreferredMaintenanceWindow": "sat:10:00-sat:11:00",
            "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxx152:My_Topic",
            "Port": 6379,
            "CacheParameterGroupName": "default.redis5.0",
            "CacheSubnetGroupName": "kxkxk",
            "VpcId": "vpc-a3e97cdb",
            "AutoMinorVersionUpgrade": true,
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "09:00-10:00",
            "NodeSnapshots": [
                {
                    "CacheNodeId": "0001",
                    "CacheSize": "5 MB",
                    "CacheNodeCreateTime": "2019-11-26T00:14:52.439Z",
                    "SnapshotCreateTime": "2019-11-26T00:25:01Z"
```

```
                }
            ]
        },
        {
            "SnapshotName": "my-cluster",
            "CacheClusterId": "my-cluster-003",
            "SnapshotStatus": "available",
            "SnapshotSource": "manual",
            "CacheNodeType": "cache.r5.large",
            "Engine": "redis",
            "EngineVersion": "5.0.5",
            "NumCacheNodes": 1,
            "PreferredAvailabilityZone": "us-west-2a",
            "CacheClusterCreateTime": "2019-11-25T23:56:17.186Z",
            "PreferredMaintenanceWindow": "sat:10:00-sat:11:00",
            "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxx152:My_Topic",
            "Port": 6379,
            "CacheParameterGroupName": "default.redis5.0",
            "CacheSubnetGroupName": "kxkxk",
            "VpcId": "vpc-a3e97cdb",
            "AutoMinorVersionUpgrade": true,
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "09:00-10:00",
            "NodeSnapshots": [
                {
                    "CacheNodeId": "0001",
                    "CacheSize": "5 MB",
                    "CacheNodeCreateTime": "2019-11-25T23:56:17.186Z",
                    "SnapshotCreateTime": "2019-11-26T03:08:33Z"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱 Elasticache 使用者指南中的備份和還原 ElastiCache for Redis。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeSnapshots。

describe-update-actions

以下程式碼範例顯示如何使用 describe-update-actions。

AWS CLI

描述更新動作

下列describe-update-actions範例會傳回更新動作的詳細資訊。

```
aws elasticache describe-update-actions
```

輸出：

```
{
    "UpdateActions": [
        {
            "ReplicationGroupId": "mycluster",
            "ServiceUpdateName": "elc-20191007-001",
            "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
            "ServiceUpdateSeverity": "important",
            "ServiceUpdateStatus": "available",
            "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
            "ServiceUpdateType": "security-update",
            "UpdateActionAvailableDate": "2019-12-05T19:15:19.995Z",
            "UpdateActionStatus": "complete",
            "NodesUpdated": "9/9",
            "UpdateActionStatusModifiedDate": "2019-12-05T19:15:20.461Z",
            "SlaMet": "n/a",
            "Engine": "redis"
        },
        {
            "CacheClusterId": "my-memcached-cluster",
            "ServiceUpdateName": "elc-20191007-001",
            "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
            "ServiceUpdateSeverity": "important",
            "ServiceUpdateStatus": "available",
            "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
            "ServiceUpdateType": "security-update",
            "UpdateActionAvailableDate": "2019-12-04T18:26:05.349Z",
            "UpdateActionStatus": "complete",
            "NodesUpdated": "1/1",
            "UpdateActionStatusModifiedDate": "2019-12-04T18:26:05.352Z",
            "SlaMet": "n/a",
            "Engine": "redis"
        },
        {
```

```
            "ReplicationGroupId": "my-cluster",
            "ServiceUpdateName": "elc-20191007-001",
            "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
            "ServiceUpdateSeverity": "important",
            "ServiceUpdateStatus": "available",
            "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
            "ServiceUpdateType": "security-update",
            "UpdateActionAvailableDate": "2019-11-26T03:36:26.320Z",
            "UpdateActionStatus": "complete",
            "NodesUpdated": "4/4",
            "UpdateActionStatusModifiedDate": "2019-12-04T22:11:12.664Z",
            "SlaMet": "n/a",
            "Engine": "redis"
        },
        {
            "ReplicationGroupId": "my-cluster2",
            "ServiceUpdateName": "elc-20191007-001",
            "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
            "ServiceUpdateSeverity": "important",
            "ServiceUpdateStatus": "available",
            "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
            "ServiceUpdateType": "security-update",
            "UpdateActionAvailableDate": "2019-11-26T01:26:01.617Z",
            "UpdateActionStatus": "complete",
            "NodesUpdated": "3/3",
            "UpdateActionStatusModifiedDate": "2019-11-26T01:26:01.753Z",
            "SlaMet": "n/a",
            "Engine": "redis"
        }
    ]
}
```

如需詳細資訊，請參閱 Elasticache 使用者指南中的 Amazon ElastiCache 中的自助式更新。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeUpdateActions。

`describe-user-groups`

以下程式碼範例顯示如何使用 `describe-user-groups`。

AWS CLI

描述使用者群組

下列describe-user-groups範例會傳回使用者群組清單。

```
aws elasticache describe-user-groups
```

輸出：

```
{
    "UserGroups": [
        {
            "UserGroupId": "myusergroup",
            "Status": "active",
            "Engine": "redis",
            "UserIds": [
                "default"
            ],
            "ReplicationGroups": [],
            "ARN": "arn:aws:elasticache:us-
west-2:xxxxxxxxxx52:usergroup:myusergroup"
        }
    ]
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用角色型存取控制 (RBAC) 驗證使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeUserGroups。

describe-users

以下程式碼範例顯示如何使用 describe-users。

AWS CLI

描述使用者

下列describe-users範例會傳回使用者清單。

```
aws elasticache describe-users
```

輸出：

```
{
    "Users": [
```

```
        {
            "UserId": "default",
            "UserName": "default",
            "Status": "active",
            "Engine": "redis",
            "AccessString": "on ~* +@all",
            "UserGroupIds": [
                "myusergroup"
            ],
            "Authentication": {
                "Type": "no-password"
            },
            "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:default"
        },
        {
            "UserId": "user1",
            "UserName": "myUser",
            "Status": "active",
            "Engine": "redis",
            "AccessString": "on ~* +@all",
            "UserGroupIds": [],
            "Authentication": {
                "Type": "password",
                "PasswordCount": 1
            },
            "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user1"
        },
        {
            "UserId": "user2",
            "UserName": "myUser",
            "Status": "active",
            "Engine": "redis",
            "AccessString": "on ~app::* -@all +@read +@hash +@bitmap +@geo -setbit -
bitfield -hset -hsetnx -hmset -hincrby -hincrbyfloat -hdel -bitop -geoadd -georadius
 -georadiusbymember",
            "UserGroupIds": [],
            "Authentication": {
                "Type": "password",
                "PasswordCount": 1
            },
            "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user2"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用角色型存取控制 (RBAC) 驗證使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeUsers。

disassociate-global-replication-group

以下程式碼範例顯示如何使用 disassociate-global-replication-group。

AWS CLI

將次要叢集與全域複寫群組建立關聯

下列disassociate-global-replication-group範例會從全域資料存放區移除次要叢集

```
aws elasticache disassociate-global-replication-group \
    --global-replication-group-id my-grg \
    --replication-group-id my-cluster-grg-secondary \
    --replication-group-region us-east-1
```

輸出：

```
{
    "GlobalReplicationGroup": {
        "GlobalReplicationGroupId": "my-grg",
        "GlobalReplicationGroupDescription": "my-grg",
        "Status": "modifying",
        "CacheNodeType": "cache.r5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.6",
        "Members": [
            {
                "ReplicationGroupId": "my-cluster-grg-secondary",
                "ReplicationGroupRegion": "us-east-1",
                "Role": "SECONDARY",
                "AutomaticFailover": "enabled",
                "Status": "associated"
            },
            {
                "ReplicationGroupId": "my-cluster-grg",
                "ReplicationGroupRegion": "us-west-2",
                "Role": "PRIMARY",
```

```
                "AutomaticFailover": "enabled",
                "Status": "associated"
            }
        ],
        "ClusterEnabled": false,
        "AuthTokenEnabled": false,
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用全域資料存放區跨 AWS 區域複寫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateGlobalReplicationGroup。

increase-node-groups-in-global-replication-group

以下程式碼範例顯示如何使用 increase-node-groups-in-global-replication-group。

AWS CLI

若要增加全域複寫群組中的節點群組數量

以下使用 Redis 引擎increase-node-groups-in-global-replication-group增加節點群組計數。

```
aws elasticache increase-node-groups-in-global-replication-group \
    --global-replication-group-id sgaui-pat-test-4 \
    --node-group-count 6 \
    --apply-immediately
```

輸出：

```
{
    "GlobalReplicationGroup": {
        "GlobalReplicationGroupId": "sgaui-test-4",
        "GlobalReplicationGroupDescription": "test-4",
        "Status": "modifying",
        "CacheNodeType": "cache.r5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.6",
        "Members": [
            {
```

```
                "ReplicationGroupId": "my-cluster-b",
                "ReplicationGroupRegion": "us-east-1",
                "Role": "SECONDARY",
                "AutomaticFailover": "enabled",
                "Status": "associated"
            },
            {
                "ReplicationGroupId": "my-cluster-a",
                "ReplicationGroupRegion": "us-west-2",
                "Role": "PRIMARY",
                "AutomaticFailover": "enabled",
                "Status": "associated"
            }
        ],
        "ClusterEnabled": true,
        "GlobalNodeGroups": [
            {
                "GlobalNodeGroupId": "sgaui-test-4-0001",
                "Slots": "0-234,2420-5461"
            },
            {
                "GlobalNodeGroupId": "sgaui-test-4-0002",
                "Slots": "5462-5904,6997-9830"
            },
            {
                "GlobalNodeGroupId": "sgaui-test-4-0003",
                "Slots": "10923-11190,13375-16383"
            },
            {
                "GlobalNodeGroupId": "sgaui-test-4-0004",
                "Slots": "235-2419,5905-6996"
            },
            {
                "GlobalNodeGroupId": "sgaui-test-4-0005",
                "Slots": "9831-10922,11191-13374"
            }
        ],
        "AuthTokenEnabled": false,
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用全域資料存放區跨 AWS 區域複寫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 IncreaseNodeGroupsInGlobalReplicationGroup。

`increase-replica-count`

以下程式碼範例顯示如何使用 `increase-replica-count`。

AWS CLI

若要增加複本計數

下列`increase-replica-count`範例會執行以下兩項操作的其中之一。它可以動態增加 Redis （停用叢集模式） 複寫群組中的複本數量。或者，它可以動態增加 Redis （啟用叢集模式） 複寫群組的一或多個節點群組 （碎片） 中的複本節點數量。此操作會在沒有叢集停機時間的情況下執行。

```
aws elasticache increase-replica-count \
    --replication-group-id "my-cluster" \
    --apply-immediately \
    --new-replica-count 3
```

輸出：

```
{
     "ReplicationGroup": {
         "ReplicationGroupId": "my-cluster",
         "Description": " ",
         "Status": "modifying",
         "PendingModifiedValues": {},
         "MemberClusters": [
             "my-cluster-001",
             "my-cluster-002",
             "my-cluster-003",
             "my-cluster-004"
         ],
         "NodeGroups": [
             {
                 "NodeGroupId": "0001",
                 "Status": "modifying",
                 "PrimaryEndpoint": {
```

```
                    "Address": "my-
cluster.xxxxxih.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "ReaderEndpoint": {
                    "Address": "my-cluster-
ro.xxxxxxih.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "my-cluster-001",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "my-
cluster-001.xxxxxih.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "primary"
                    },
                    {
                        "CacheClusterId": "my-cluster-003",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "my-
cluster-003.xxxxxih.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "replica"
                    }
                ]
            }
        ],
        "AutomaticFailover": "disabled",
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "07:30-08:30",
        "ClusterEnabled": false,
        "CacheNodeType": "cache.r5.xlarge",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
```

```
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的增加碎片中的複本數量。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 IncreaseReplicaCount。

list-allowed-node-type-modifications

以下程式碼範例顯示如何使用 list-allowed-node-type-modifications。

AWS CLI

列出允許的節點修改

下列list-allowed-node-type-modifications範例列出您可以擴展 Redis 叢集或複寫群組目前節點類型的所有可用節點類型。

```
aws elasticache list-allowed-node-type-modifications \
    --replication-group-id "my-replication-group"
```

輸出：

```
{
    "ScaleUpModifications": [
        "cache.m5.12xlarge",
        "cache.m5.24xlarge",
        "cache.m5.4xlarge",
        "cache.r5.12xlarge",
        "cache.r5.24xlarge",
        "cache.r5.2xlarge",
        "cache.r5.4xlarge"
    ],
    "ScaleDownModifications": [
        "cache.m3.large",
        "cache.m3.medium",
        "cache.m3.xlarge",
        "cache.m4.large",
        "cache.m4.xlarge",
        "cache.m5.2xlarge",
        "cache.m5.large",
        "cache.m5.xlarge",
        "cache.r3.large",
        "cache.r4.large",
```

```
        "cache.r4.xlarge",
        "cache.r5.large",
        "cache.t2.medium",
        "cache.t2.micro",
        "cache.t2.small",
        "cache.t3.medium",
        "cache.t3.micro",
        "cache.t3.small"
    ]
}
```

如需詳細資訊，請參閱 Elasticache 使用者指南中的擴展 ElastiCache for Redis 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAllowedNodeTypeModifications。

list-tags-for-resource

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出資源的標籤

下列`list-tags-for-resource`範例列出資源的標籤。

```
aws elasticache list-tags-for-resource \
    --resource-name "arn:aws:elasticache:us-east-1:123456789012:cluster:my-cluster"
```

輸出：

```
{
    "TagList": [
        {
            "Key": "Project",
            "Value": "querySpeedUp"
        },
        {
            "Key": "Environment",
            "Value": "PROD"
        }
    ]
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用 AWS CLI 列出標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

modify-cache-cluster

以下程式碼範例顯示如何使用 modify-cache-cluster。

AWS CLI

修改快取叢集

下列modify-cache-cluster範例會修改指定叢集的設定。

```
aws elasticache modify-cache-cluster \
    --cache-cluster-id "my-cluster" \
    --num-cache-nodes 1
```

輸出：

```
{
    "CacheCluster": {
        "CacheClusterId": "my-cluster",
        "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
        "CacheNodeType": "cache.m5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.5",
        "CacheClusterStatus": "available",
        "NumCacheNodes": 1,
        "PreferredAvailabilityZone": "us-west-2c",
        "CacheClusterCreateTime": "2019-12-04T18:24:56.652Z",
        "PreferredMaintenanceWindow": "sat:10:00-sat:11:00",
        "PendingModifiedValues": {},
        "CacheSecurityGroups": [],
        "CacheParameterGroup": {
            "CacheParameterGroupName": "default.redis5.0",
            "ParameterApplyStatus": "in-sync",
            "CacheNodeIdsToReboot": []
        },
        "CacheSubnetGroupName": "default",
        "AutoMinorVersionUpgrade": true,
        "SnapshotRetentionLimit": 0,
```

```
        "SnapshotWindow": "07:00-08:00",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

如需詳細資訊，請參閱 Elasticache 使用者指南中的修改 ElastiCache 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyCacheCluster。

modify-cache-parameter-group

以下程式碼範例顯示如何使用 modify-cache-parameter-group。

AWS CLI

修改快取參數群組

下列modify-cache-parameter-group範例會修改指定快取參數群組的參數。

```
aws elasticache modify-cache-parameter-group \
    --cache-parameter-group-name "mygroup" \
    --parameter-name-values "ParameterName=activedefrag, ParameterValue=no"
```

輸出：

```
{
    "CacheParameterGroupName": "mygroup"
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的修改參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyCacheParameterGroup。

modify-cache-subnet-group

以下程式碼範例顯示如何使用 modify-cache-subnet-group。

AWS CLI

修改快取子網路群組

下列modify-cache-subnet-group範例會修改指定的快取子網路群組。

```
aws elasticache modify-cache-subnet-group \
    --cache-subnet-group-name kxkxk \
    --cache-subnet-group-description "mygroup"
```

輸出：

```
{
    "CacheSubnetGroup": {
        "CacheSubnetGroupName": "kxkxk",
        "CacheSubnetGroupDescription": "mygroup",
        "VpcId": "vpc-xxxxcdb",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-xxxxbff",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                }
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的修改子網路群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyCacheSubnetGroup。

modify-global-replication-group

以下程式碼範例顯示如何使用 modify-global-replication-group。

AWS CLI

修改全域複寫群組

以下內容會使用 Redis modify-global-replication-group 引擎修改全域複寫群組的屬性，在此情況下停用自動容錯移轉。

```
aws elasticache modify-global-replication-group \
    --global-replication-group-id sgaui-pat-group \
    --apply-immediately \
    --no-automatic-failover-enabled
```

輸出

```
{
    "GlobalReplicationGroup": {
        "GlobalReplicationGroupId": "sgaui-test-group",
        "GlobalReplicationGroupDescription": " ",
        "Status": "modifying",
        "CacheNodeType": "cache.r5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.6",
        "ClusterEnabled": false,
        "AuthTokenEnabled": false,
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用全域資料存放區跨 AWS 區域複寫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyGlobalReplicationGroup。

modify-replication-group-shard-configuration

以下程式碼範例顯示如何使用 modify-replication-group-shard-configuration。

AWS CLI

修改複寫群組碎片組態

以下使用 Redis 引擎modify-replication-group-shard-configuration減少節點群組計數。

```
aws elasticache modify-replication-group-shard-configuration \
    --replication-group-id mycluster \
    --node-group-count 3 \
    --apply-immediately \
    --node-groups-to-remove 0002
```

輸出

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "mycluster",
```

```
        "Description": "mycluster",
        "GlobalReplicationGroupInfo": {},
        "Status": "modifying",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "mycluster-0002-001",
            "mycluster-0002-002",
            "mycluster-0002-003",
            "mycluster-0003-001",
            "mycluster-0003-002",
            "mycluster-0003-003",
            "mycluster-0003-004",
            "mycluster-0004-001",
            "mycluster-0004-002",
            "mycluster-0004-003",
            "mycluster-0005-001",
            "mycluster-0005-002",
            "mycluster-0005-003"
        ],
        "NodeGroups": [
            {
                "NodeGroupId": "0002",
                "Status": "modifying",
                "Slots": "894-1767,3134-4443,5149-5461,6827-7332,12570-13662",
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "mycluster-0002-001",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2c"
                    },
                    {
                        "CacheClusterId": "mycluster-0002-002",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2a"
                    },
                    {
                        "CacheClusterId": "mycluster-0002-003",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2b"
                    }
                ]
            },
            {
                "NodeGroupId": "0003",
```

```
                "Status": "modifying",
                "Slots":
"0-324,5462-5692,6784-6826,7698-8191,10923-11075,12441-12569,13663-16383",
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "mycluster-0003-001",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2c"
                    },
                    {
                        "CacheClusterId": "mycluster-0003-002",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2b"
                    },
                    {
                        "CacheClusterId": "mycluster-0003-003",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2a"
                    },
                    {
                        "CacheClusterId": "mycluster-0003-004",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2c"
                    }
                ]
            },
            {
                "NodeGroupId": "0004",
                "Status": "modifying",
                "Slots": "325-336,4706-5148,7333-7697,9012-10922,11076-12440",
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "mycluster-0004-001",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2b"
                    },
                    {
                        "CacheClusterId": "mycluster-0004-002",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2a"
                    },
                    {
                        "CacheClusterId": "mycluster-0004-003",
                        "CacheNodeId": "0001",
```

```
                        "PreferredAvailabilityZone": "us-west-2c"
                    }
                ]
            },
            {
                "NodeGroupId": "0005",
                "Status": "modifying",
                "Slots": "337-893,1768-3133,4444-4705,5693-6783,8192-9011",
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "mycluster-0005-001",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2a"
                    },
                    {
                        "CacheClusterId": "mycluster-0005-002",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2c"
                    },
                    {
                        "CacheClusterId": "mycluster-0005-003",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2b"
                    }
                ]
            }
        ],
        "AutomaticFailover": "enabled",
        "MultiAZ": "enabled",
        "ConfigurationEndpoint": {
            "Address": "mycluster.g2xbih.clustercfg.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "SnapshotRetentionLimit": 1,
        "SnapshotWindow": "13:00-14:00",
        "ClusterEnabled": true,
        "CacheNodeType": "cache.r5.xlarge",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

如需詳細資訊，請參閱 Elasticache 使用者指南中的擴展 ElastiCache for Redis 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyReplicationGroupShardConfiguration。

modify-replication-group

以下程式碼範例顯示如何使用 modify-replication-group。

AWS CLI

修改複寫群組

以下使用 Redis 引擎modify-replication-group停用異地同步備份。

```
aws elasticache modify-replication-group \
    --replication-group-id test-cluster \
    --no-multi-az-enabled \
    --apply-immediately
```

輸出

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "test-cluster",
        "Description": "test-cluster",
        "GlobalReplicationGroupInfo": {
            "GlobalReplicationGroupId": "sgaui-pat-group",
            "GlobalReplicationGroupMemberRole": "PRIMARY"
        },
        "Status": "available",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "test-cluster-001",
            "test-cluster-002",
            "test-cluster-003"
        ],
        "NodeGroups": [
            {
                "NodeGroupId": "0001",
                "Status": "available",
                "PrimaryEndpoint": {
                    "Address": "test-
cluster.g2xbih.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
```

```
                },
                "ReaderEndpoint": {
                    "Address": "test-cluster-
ro.g2xbih.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "test-cluster-001",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "test-
cluster-001.g2xbih.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2c",
                        "CurrentRole": "primary"
                    },
                    {
                        "CacheClusterId": "test-cluster-002",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "test-
cluster-002.g2xbih.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2b",
                        "CurrentRole": "replica"
                    },
                    {
                        "CacheClusterId": "test-cluster-003",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "test-
cluster-003.g2xbih.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "replica"
                    }
                ]
            }
        ],
        "SnapshottingClusterId": "test-cluster-002",
```

```
        "AutomaticFailover": "enabled",
        "MultiAZ": "disabled",
        "SnapshotRetentionLimit": 1,
        "SnapshotWindow": "08:00-09:00",
        "ClusterEnabled": false,
        "CacheNodeType": "cache.r5.large",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的修改複寫群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyReplicationGroup。

modify-user-group

以下程式碼範例顯示如何使用 modify-user-group。

AWS CLI

修改使用者群組

下列modify-user-group範例會將使用者新增至使用者群組。

```
aws elasticache modify-user-group \
    --user-group-id myusergroup \
    --user-ids-to-add user1
```

輸出：

```
{
    "UserGroupId": "myusergroup",
    "Status": "modifying",
    "Engine": "redis",
    "UserIds": [
        "default"
    ],
    "PendingChanges": {
        "UserIdsToAdd": [
            "user1"
        ]
```

```
    },
    "ReplicationGroups": [],
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用角色型存取控制 (RBAC) 驗證使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyUserGroup。

`modify-user`

以下程式碼範例顯示如何使用 `modify-user`。

AWS CLI

修改使用者

下列`modify-user`範例會修改使用者的存取字串。

```
aws elasticache modify-user \
    --user-id user2 \
    --append-access-string "on ~* +@all"
```

輸出：

```
{
    "UserId": "user2",
    "UserName": "myUser",
    "Status": "modifying",
    "Engine": "redis",
    "AccessString": "on ~* +@all",
    "UserGroupIds": [],
    "Authentication": {
        "Type": "password",
        "PasswordCount": 1
    },
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user2"
}
```

如需詳細資訊，請參閱《Elasticache 使用者指南》中的使用角色型存取控制 (RBAC) 驗證使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyUser。

purchase-reserved-cache-nodes-offering

以下程式碼範例顯示如何使用 purchase-reserved-cache-nodes-offering。

AWS CLI

購買 reserved-cache-node-offering

下列purchase-reserved-cache-nodes-offering範例可讓您購買預留快取節點方案。

```
aws elasticache purchase-reserved-cache-nodes-offering \
    --reserved-cache-nodes-offering-id xxxxxxx-4da5-4b90-b92d-929fbd7abed2
```

輸出

```
{
    "ReservedCacheNode": {
        "ReservedCacheNodeId": "ri-2020-06-30-17-59-40-474",
        "ReservedCacheNodesOfferingId": "xxxxxxx-4da5-4b90-b92d-929fbd7abed2",
        "CacheNodeType": "cache.m3.2xlarge",
        "StartTime": "2020-06-30T17:59:40.474000+00:00",
        "Duration": 31536000,
        "FixedPrice": 1772.0,
        "UsagePrice": 0.0,
        "CacheNodeCount": 1,
        "ProductDescription": "redis",
        "OfferingType": "Heavy Utilization",
        "State": "payment-pending",
        "RecurringCharges": [
            {
                "RecurringChargeAmount": 0.25,
                "RecurringChargeFrequency": "Hourly"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Elasticache Redis 使用者指南》中的取得預留節點方案的相關資訊，或《Elasticache Memcached 使用者指南》中的取得預留節點方案的相關資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PurchaseReservedCacheNodesOffering。

reboot-cache-cluster

以下程式碼範例顯示如何使用 reboot-cache-cluster。

AWS CLI

重新啟動快取叢集

下列reboot-cache-cluster範例會重新啟動佈建叢集中的部分或全部快取節點。此操作會將任何修改過的快取參數群組套用至叢集。重新啟動操作會盡快執行，並導致叢集暫時中斷。在重新開機期間，叢集狀態會設為 REBOOTING。

```
aws elasticache reboot-cache-cluster \
    --cache-cluster-id "my-cluster-001" \
    --cache-node-ids-to-reboot "0001"
```

輸出：

```
{
    "CacheCluster": {
        "CacheClusterId": "my-cluster-001",
        "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
        "CacheNodeType": "cache.r5.xlarge",
        "Engine": "redis",
        "EngineVersion": "5.0.5",
        "CacheClusterStatus": "rebooting cache cluster nodes",
        "NumCacheNodes": 1,
        "PreferredAvailabilityZone": "us-west-2a",
        "CacheClusterCreateTime": "2019-11-26T03:35:04.546Z",
        "PreferredMaintenanceWindow": "mon:04:05-mon:05:05",
        "PendingModifiedValues": {},
        "NotificationConfiguration": {
            "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxx152:My_Topic",
            "TopicStatus": "active"
        },
        "CacheSecurityGroups": [],
        "CacheParameterGroup": {
            "CacheParameterGroupName": "mygroup",
            "ParameterApplyStatus": "in-sync",
            "CacheNodeIdsToReboot": []
        },
        "CacheSubnetGroupName": "kxkxk",
```

```
        "AutoMinorVersionUpgrade": true,
        "SecurityGroups": [
            {
                "SecurityGroupId": "sg-xxxxxxxxxxxx836",
                "Status": "active"
            },
            {
                "SecurityGroupId": "sg-xxxxxxxx7b",
                "Status": "active"
            }
        ],
        "ReplicationGroupId": "my-cluster",
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "07:30-08:30",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

如需詳細資訊，請參閱 Elasticache 使用者指南中的重新啟動叢集 <https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Clusters.Rebooting.html。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RebootCacheCluster。

`reset-cache-parameter-group`

以下程式碼範例顯示如何使用 `reset-cache-parameter-group`。

AWS CLI

重設快取參數群組

下列`reset-cache-parameter-group`範例會將快取參數群組的參數修改為引擎或系統預設值。您可以提交參數名稱清單來重設特定參數。若要重設整個快取參數群組，請指定 `--reset-all-parameters`和 `--cache-parameter-group-name` 參數。

```
aws elasticache reset-cache-parameter-group \
    --cache-parameter-group-name "mygroup" \
    --reset-all-parameters
```

輸出：

```
{
```

```
    "CacheParameterGroupName": "mygroup"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetCacheParameterGroup。

`start-migration`

以下程式碼範例顯示如何使用 `start-migration`。

AWS CLI

開始遷移

以下使用 Redis 引擎，將您的資料從 Amazon EC2 上的自我託管 Redis `start-migration`遷移至 Amazon ElastiCache。

```
aws elasticache start-migration \
   --replication-group-id test \
   --customer-node-endpoint-
list "Address='test.g2xbih.ng.0001.usw2.cache.amazonaws.com',Port=6379"
```

輸出

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "test",
        "Description": "test",
        "GlobalReplicationGroupInfo": {},
        "Status": "modifying",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "test-001",
            "test-002",
            "test-003"
        ],
        "NodeGroups": [
            {
                "NodeGroupId": "0001",
                "Status": "available",
                "PrimaryEndpoint": {
                    "Address": "test.g2xbih.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
```

```
                },
                "ReaderEndpoint": {
                    "Address": "test-ro.g2xbih.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "test-001",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address":
"test-001.g2xbih.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "primary"
                    },
                    {
                        "CacheClusterId": "test-002",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address":
"test-002.g2xbih.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2c",
                        "CurrentRole": "replica"
                    },
                    {
                        "CacheClusterId": "test-003",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address":
"test-003.g2xbih.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2b",
                        "CurrentRole": "replica"
                    }
                ]
            }
        ],
        "SnapshottingClusterId": "test-002",
        "AutomaticFailover": "enabled",
```

```
        "MultiAZ": "enabled",
        "SnapshotRetentionLimit": 1,
        "SnapshotWindow": "07:30-08:30",
        "ClusterEnabled": false,
        "CacheNodeType": "cache.r5.large",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

如需詳細資訊，請參閱 Elasticache 使用者指南中的線上遷移至 ElastiCache。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartMigration。

`test-failover`

以下程式碼範例顯示如何使用 `test-failover`。

AWS CLI

測試節點群組的容錯移轉

下列`test-failover`範例會在複寫群組（在主控台中稱為叢集）中的指定節點群組（在主控台中稱為碎片）上測試自動容錯移轉。

```
aws elasticache test-failover /
    --replication-group-id "mycluster" /
    --node-group-id "0001"
```

輸出：

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "mycluster",
        "Description": "My Cluster",
        "Status": "available",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "mycluster-0001-001",
            "mycluster-0001-002",
            "mycluster-0001-003",
            "mycluster-0002-001",
            "mycluster-0002-002",
```

```
            "mycluster-0002-003",
            "mycluster-0003-001",
            "mycluster-0003-002",
            "mycluster-0003-003"
        ],
        "NodeGroups": [
            {
                "NodeGroupId": "0001",
                "Status": "available",
                "Slots": "0-5461",
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "mycluster-0001-001",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2b"
                    },
                    {
                        "CacheClusterId": "mycluster-0001-002",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2a"
                    },
                    {
                        "CacheClusterId": "mycluster-0001-003",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2c"
                    }
                ]
            },
            {
                "NodeGroupId": "0002",
                "Status": "available",
                "Slots": "5462-10922",
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "mycluster-0002-001",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2a"
                    },
                    {
                        "CacheClusterId": "mycluster-0002-002",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2b"
                    },
                    {
```

```
                        "CacheClusterId": "mycluster-0002-003",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2c"
                    }
                ]
            },
            {
                "NodeGroupId": "0003",
                "Status": "available",
                "Slots": "10923-16383",
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "mycluster-0003-001",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2c"
                    },
                    {
                        "CacheClusterId": "mycluster-0003-002",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2b"
                    },
                    {
                        "CacheClusterId": "mycluster-0003-003",
                        "CacheNodeId": "0001",
                        "PreferredAvailabilityZone": "us-west-2a"
                    }
                ]
            }
        ],
        "AutomaticFailover": "enabled",
        "ConfigurationEndpoint": {
            "Address": "mycluster.xxxxih.clustercfg.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "SnapshotRetentionLimit": 1,
        "SnapshotWindow": "13:00-14:00",
        "ClusterEnabled": true,
        "CacheNodeType": "cache.r5.large",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TestFailover。

使用 的 MediaStore 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 MediaStore 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-container`

以下程式碼範例顯示如何使用 `create-container`。

AWS CLI

建立容器

下列`create-container`範例會建立新的空容器。

```
aws mediastore create-container --container-name ExampleContainer
```

輸出：

```
{
    "Container": {
        "AccessLoggingEnabled": false,
        "CreationTime": 1563557265,
        "Name": "ExampleContainer",
        "Status": "CREATING",
        "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer"
    }
}
```

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的建立容器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateContainer。

delete-container-policy

以下程式碼範例顯示如何使用 delete-container-policy。

AWS CLI

刪除容器政策

下列delete-container-policy範例會刪除指派給指定容器的政策。刪除政策時， AWS Elemental MediaStore 會自動將預設政策指派給容器。

```
aws mediastore delete-container-policy \
    --container-name LiveEvents
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaStore API 參考中的 DeleteContainerPolicy。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteContainerPolicy。

delete-container

以下程式碼範例顯示如何使用 delete-container。

AWS CLI

刪除容器

下列delete-container範例會刪除指定的容器。只有當容器沒有任何物件時，您才可以將該容器刪除。

```
aws mediastore delete-container \
    --container-name=ExampleLiveDemo
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的刪除容器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteContainer。

delete-cors-policy

以下程式碼範例顯示如何使用 delete-cors-policy。

AWS CLI

刪除 CORS 政策

下列delete-cors-policy範例會刪除指派給指定容器的跨來源資源共用 (CORS) 政策。

```
aws mediastore delete-cors-policy \
    --container-name ExampleContainer
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的刪除 CORS 政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCorsPolicy。

delete-lifecycle-policy

以下程式碼範例顯示如何使用 delete-lifecycle-policy。

AWS CLI

刪除物件生命週期政策

下列delete-lifecycle-policy範例會刪除連接至指定容器的物件生命週期政策。此變更最多可能需要 20 分鐘才會生效。

```
aws mediastore delete-lifecycle-policy \
    --container-name LiveEvents
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的刪除物件生命週期政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLifecyclePolicy。

describe-container

以下程式碼範例顯示如何使用 describe-container。

AWS CLI

檢視容器的詳細資訊

下列describe-container範例顯示指定容器的詳細資訊。

```
aws mediastore describe-container \
    --container-name ExampleContainer
```

輸出：

```
{
    "Container": {
        "CreationTime": 1563558086,
        "AccessLoggingEnabled": false,
        "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer",
        "Status": "ACTIVE",
        "Name": "ExampleContainer",
        "Endpoint": "https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com"
    }
}
```

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的檢視容器的詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeContainer。

describe-object

以下程式碼範例顯示如何使用 describe-object。

AWS CLI

檢視特定容器中的物件和資料夾清單

下列describe-object範例顯示存放在特定容器中的項目 （ 物件和資料夾 ）。

```
aws mediastore-data describe-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path /folder_name/file1234.jpg
```

輸出：

```
{
    "ContentType": "image/jpeg",
    "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
    "ContentLength": "2307346",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
}
```

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的檢視物件的詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeObject。

get-container-policy

以下程式碼範例顯示如何使用 get-container-policy。

AWS CLI

檢視容器政策

下列get-container-policy範例顯示指定容器的資源型政策。

```
aws mediastore get-container-policy \
    --container-name ExampleLiveDemo
```

輸出：

```
{
    "Policy": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "PublicReadOverHttps",
                "Effect": "Allow",
                "Principal": {
                    "AWS": "arn:aws:iam::111122223333:root"
                },
                "Action": [
                    "mediastore:GetObject",
                    "mediastore:DescribeObject"
                ],
                "Resource": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleLiveDemo/",
```

```
                "Condition": {
                    "Bool": {
                        "aws:SecureTransport": "true"
                    }
                }
            }
        ]
    }
}
```

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的檢視容器政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetContainerPolicy。

get-cors-policy

以下程式碼範例顯示如何使用 get-cors-policy。

AWS CLI

檢視 CORS 政策

下列get-cors-policy範例顯示指派給指定容器的跨來源資源共用 (CORS) 政策。

```
aws mediastore get-cors-policy \
    --container-name ExampleContainer \
    --region us-west-2
```

輸出：

```
{
    "CorsPolicy": [
        {
            "AllowedMethods": [
                "GET",
                "HEAD"
            ],
            "MaxAgeSeconds": 3000,
            "AllowedOrigins": [
                ""
            ],
            "AllowedHeaders": [
```

```
                ""
            ]
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的檢視 CORS 政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCorsPolicy。

get-lifecycle-policy

以下程式碼範例顯示如何使用 get-lifecycle-policy。

AWS CLI

檢視物件生命週期政策

下列get-lifecycle-policy範例顯示連接至指定容器的物件生命週期政策。

```
aws mediastore get-lifecycle-policy \
    --container-name LiveEvents
```

輸出：

```
{
    "LifecyclePolicy": {
        "rules": [
            {
                "definition": {
                    "path": [
                        {
                            "prefix": "Football/"
                        },
                        {
                            "prefix": "Baseball/"
                        }
                    ],
                    "days_since_create": [
                        {
                            "numeric": [
                                ">",
```

```
                                28
                            ]
                        }
                    ]
                },
                "action": "EXPIRE"
            }
        ]
    }
}
```

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的檢視物件生命週期政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLifecyclePolicy。

get-object

以下程式碼範例顯示如何使用 get-object。

AWS CLI

下載物件

下列get-object範例會將物件下載至指定的端點。

```
aws mediastore-data get-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path=/folder_name/README.md README.md
```

輸出：

```
{
    "ContentLength": "2307346",
    "ContentType": "image/jpeg",
    "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
    "StatusCode": 200
}
```

下載部分物件

下列get-object範例會將物件的一部分下載到指定的端點。

```
aws mediastore-data get-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path /folder_name/README.md \
    --range="bytes=0-100" README2.md
```

輸出：

```
{
    "StatusCode": 206,
    "ContentRange": "bytes 0-100/2307346",
    "ContentLength": "101",
    "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
    "ContentType": "image/jpeg",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
}
```

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的下載物件。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetObject。

list-containers

以下程式碼範例顯示如何使用 list-containers。

AWS CLI

檢視容器清單

下列list-containers範例顯示與您的帳戶相關聯的所有容器清單。

```
aws mediastore list-containers
```

輸出：

```
{
    "Containers": [
        {
            "CreationTime": 1505317931,
            "Endpoint": "https://aaabbbcccdddee.data.mediastore.us-
west-2.amazonaws.com",
            "Status": "ACTIVE",
```

```
            "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleLiveDemo",
            "AccessLoggingEnabled": false,
            "Name": "ExampleLiveDemo"
        },
        {
            "CreationTime": 1506528818,
            "Endpoint": "https://fffgggghhhiiijj.data.mediastore.us-
west-2.amazonaws.com",
            "Status": "ACTIVE",
            "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer",
            "AccessLoggingEnabled": false,
            "Name": "ExampleContainer"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的檢視容器清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListContainers。

list-items

以下程式碼範例顯示如何使用 `list-items`。

AWS CLI

範例 1：檢視特定容器中的物件和資料夾清單

下列`list-items`範例顯示存放在指定容器中的項目 （ 物件和資料夾 ）。

```
aws mediastore-data list-items \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com
```

輸出：

```
{
    "Items": [
        {
            "ContentType": "image/jpeg",
            "LastModified": 1563571859.379,
```

```
            "Name": "filename.jpg",
            "Type": "OBJECT",
            "ETag":
 "543ab21abcd1a234ab123456a1a2b12345ab12abc12a1234abc1a2bc12345a12",
            "ContentLength": 3784
        },
        {
            "Type": "FOLDER",
            "Name": "ExampleLiveDemo"
        }
    ]
}
```

範例 2：檢視特定資料夾中的物件和資料夾清單

下列`list-items`範例顯示存放在特定資料夾中的項目 （ 物件和資料夾 ）。

```
aws mediastore-data list-items \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com
```

輸出：

```
{
    "Items": [
        {
            "ContentType": "image/jpeg",
            "LastModified": 1563571859.379,
            "Name": "filename.jpg",
            "Type": "OBJECT",
            "ETag":
 "543ab21abcd1a234ab123456a1a2b12345ab12abc12a1234abc1a2bc12345a12",
            "ContentLength": 3784
        },
        {
            "Type": "FOLDER",
            "Name": "ExampleLiveDemo"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的檢視物件清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListItems。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出容器的標籤

下列list-tags-for-resource範例顯示指派給指定容器的標籤索引鍵和值。

```
aws mediastore list-tags-for-resource \
    --resource arn:aws:mediastore:us-west-2:1213456789012:container/ExampleContainer
```

輸出：

```
{
    "Tags": [
        {
            "Value": "Test",
            "Key": "Environment"
        },
        {
            "Value": "West",
            "Key": "Region"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Elemental MediaStore API 參考中的 ListTagsForResource。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

put-container-policy

以下程式碼範例顯示如何使用 put-container-policy。

AWS CLI

編輯容器政策

下列put-container-policy範例會將不同的政策指派給指定的容器。在此範例中，已更新的政策是在名為 的檔案中定義LiveEventsContainerPolicy.json。

```
aws mediastore put-container-policy \
    --container-name LiveEvents \
    --policy file://LiveEventsContainerPolicy.json
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的編輯容器政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutContainerPolicy。

put-cors-policy

以下程式碼範例顯示如何使用 put-cors-policy。

AWS CLI

範例 1：新增 CORS 政策

下列put-cors-policy範例會將跨來源資源共用 (CORS) 政策新增至指定的容器。CORS 政策的內容位於名為 的檔案中corsPolicy.json。

```
aws mediastore put-cors-policy \
    --container-name ExampleContainer \
    --cors-policy file://corsPolicy.json
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的將 CORS 政策新增至容器。

範例 2：編輯 CORS 政策

下列put-cors-policy範例會更新指派給指定容器的跨來源資源共用 (CORS) 政策。更新後的 CORS 政策內容位於名為 的檔案中corsPolicy2.json。

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的編輯 CORS 政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutCorsPolicy。

put-lifecycle-policy

以下程式碼範例顯示如何使用 put-lifecycle-policy。

AWS CLI

建立物件生命週期政策

下列put-lifecycle-policy範例會將物件生命週期政策連接至指定的容器。這可讓您指定服務在容器中存放物件的時間長度。MediaStore 會在物件達到過期日期後刪除容器中的物件，如 政策所述，該政策位於名為 的檔案中LiveEventsLifecyclePolicy.json。

```
aws mediastore put-lifecycle-policy \
    --container-name ExampleContainer \
    --lifecycle-policy file://ExampleLifecyclePolicy.json
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的將物件生命週期政策新增至容器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutLifecyclePolicy。

put-object

以下程式碼範例顯示如何使用 put-object。

AWS CLI

上傳物件

下列put-object範例會將物件上傳至指定的容器。您可以指定將物件儲存在容器中的資料夾路徑。如果資料夾已存在， AWS Elemental MediaStore 會將物件存放在資料夾中。如果資料夾不存在，服務會建立資料夾，然後將物件存放在資料夾中。

```
aws mediastore-data put-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --body README.md \
    --path /folder_name/README.md \
    --cache-control "max-age=6, public" \
    --content-type binary/octet-stream
```

輸出：

```
{
```

```
    "ContentSHA256":
 "74b5fdb517f423ed750ef214c44adfe2be36e37d861eafe9c842cbe1bf387a9d",
    "StorageClass": "TEMPORAL",
    "ETag": "af3e4731af032167a106015d1f2fe934e68b32ed1aa297a9e325f5c64979277b"
}
```

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的上傳物件。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutObject。

start-access-logging

以下程式碼範例顯示如何使用 start-access-logging。

AWS CLI

在容器上啟用存取記錄

下列start-access-logging範例會在指定的容器上啟用存取記錄。

```
aws mediastore start-access-logging \
    --container-name LiveEvents
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的啟用容器的存取記錄。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartAccessLogging。

stop-access-logging

以下程式碼範例顯示如何使用 stop-access-logging。

AWS CLI

在容器上停用存取記錄

下列stop-access-logging範例會停用指定容器上的存取記錄。

```
aws mediastore stop-access-logging \
    --container-name LiveEvents
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的停用容器的存取記錄。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopAccessLogging。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤新增至容器

下列tag-resource範例會將標籤索引鍵和值新增至指定的容器。

```
aws mediastore tag-resource \
    --resource arn:aws:mediastore:us-west-2:123456789012:container/ExampleContainer
 \
    --tags '[{"Key": "Region", "Value": "West"}, {"Key": "Environment", "Value":
 "Test"}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaStore API 參考中的 TagResource。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從容器移除標籤

下列untag-resource範例會從容器移除指定的標籤索引鍵及其相關聯的值。

```
aws mediastore untag-resource \
    --resource arn:aws:mediastore:us-west-2:123456789012:container/ExampleContainer
 \
    --tag-keys Region
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaStore API 參考中的 UntagResource。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

使用 的 Amazon EMR 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon EMR 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`add-instance-fleet`

以下程式碼範例顯示如何使用 `add-instance-fleet`。

AWS CLI

將任務執行個體機群新增至叢集

此範例會將新的任務執行個體機群新增至指定的叢集。

命令：

```
aws emr add-instance-fleet --cluster-id 'j-12ABCDEFGHI34JK' --instance-fleet
 InstanceFleetType=TASK,TargetSpotCapacity=1,LaunchSpecifications={SpotSpecification='{Timeo
```

輸出：

```
{
   "ClusterId": "j-12ABCDEFGHI34JK",
   "InstanceFleetId": "if-23ABCDEFGHI45JJ"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddInstanceFleet。

add-steps

以下程式碼範例顯示如何使用 add-steps。

AWS CLI

1。將自訂 JAR 步驟新增至叢集

命令：

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps
 Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://amzn-s3-demo-
bucket/mytest.jar,Args=arg1,arg2,arg3
 Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://amzn-s3-demo-
bucket/mytest.jar,MainClass=mymainclass,Args=arg1,arg2,arg3
```

必要參數：

```
Jar
```

選用參數：

```
Type, Name, ActionOnFailure, Args
```

輸出：

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

2. 將串流步驟新增至叢集

命令：

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=STREAMING,Name='Streaming
 Program',ActionOnFailure=CONTINUE,Args=[-files,s3://elasticmapreduce/samples/
```

```
wordcount/wordSplitter.py,-mapper,wordSplitter.py,-reducer,aggregate,-input,s3://
elasticmapreduce/samples/wordcount/input,-output,s3://amzn-s3-demo-bucket/wordcount/
output]
```

必要參數：

```
Type, Args
```

選用參數：

```
Name, ActionOnFailure
```

JSON 對等 (Step.json 的內容）：

```
 [
  {
    "Name": "JSON Streaming Step",
    "Args": ["-files","s3://elasticmapreduce/samples/wordcount/wordSplitter.py","-
mapper","wordSplitter.py","-reducer","aggregate","-input","s3://elasticmapreduce/
samples/wordcount/input","-output","s3://amzn-s3-demo-bucket/wordcount/output"],
    "ActionOnFailure": "CONTINUE",
    "Type": "STREAMING"
  }
]
```

注意：JSON 引數必須在清單中包含選項和值作為自己的項目。

命令（使用 step.json)：

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps file://./step.json
```

輸出：

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

3. 將具有多個檔案的串流步驟新增至叢集（僅限 JSON)

JSON (multiplefiles.json)：

```
[
  {
     "Name": "JSON Streaming Step",
     "Type": "STREAMING",
     "ActionOnFailure": "CONTINUE",
     "Args": [
         "-files",
         "s3://amzn-s3-demo-bucket/mapper.py,s3://amzn-s3-demo-bucket/reducer.py",
         "-mapper",
         "mapper.py",
         "-reducer",
         "reducer.py",
         "-input",
         "s3://amzn-s3-demo-bucket/input",
         "-output",
         "s3://amzn-s3-demo-bucket/output"]
  }
]
```

命令：

```
aws emr add-steps --cluster-id j-XXXXXXXX  --steps file://./multiplefiles.json
```

必要參數：

```
Type, Args
```

選用參數：

```
Name, ActionOnFailure
```

輸出：

```
{
    "StepIds":[
        "s-XXXXXXXX",
    ]
}
```

4. 將 Hive 步驟新增至叢集

命令：

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=HIVE,Name='Hive
 program',ActionOnFailure=CONTINUE,Args=[-f,s3://amzn-s3-demo-bucket/
myhivescript.q,-d,INPUT=s3://amzn-s3-demo-bucket/myhiveinput,-d,OUTPUT=s3://
amzn-s3-demo-bucket/myhiveoutput,arg1,arg2] Type=HIVE,Name='Hive
 steps',ActionOnFailure=TERMINATE_CLUSTER,Args=[-f,s3://elasticmapreduce/samples/
hive-ads/libs/model-build.q,-d,INPUT=s3://elasticmapreduce/samples/hive-ads/tables,-
d,OUTPUT=s3://amzn-s3-demo-bucket/hive-ads/output/2014-04-18/11-07-32,-d,LIBS=s3://
elasticmapreduce/samples/hive-ads/libs]
```

必要參數：

```
Type, Args
```

選用參數：

```
Name, ActionOnFailure
```

輸出：

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

5. 將 Pig 步驟新增至叢集

命令：

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=PIG,Name='Pig
 program',ActionOnFailure=CONTINUE,Args=[-f,s3://amzn-s3-demo-bucket/
mypigscript.pig,-p,INPUT=s3://amzn-s3-demo-bucket/mypiginput,-p,OUTPUT=s3://
amzn-s3-demo-bucket/mypigoutput,arg1,arg2] Type=PIG,Name='Pig program',Args=[-
f,s3://elasticmapreduce/samples/pig-apache/do-reports2.pig,-p,INPUT=s3://
elasticmapreduce/samples/pig-apache/input,-p,OUTPUT=s3://amzn-s3-demo-bucket/pig-
apache/output,arg1,arg2]
```

必要參數：

```
Type, Args
```

選用參數：

```
Name, ActionOnFailure
```

輸出：

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

6. 將 Impala 步驟新增至叢集

命令：

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=IMPALA,Name='Impala
 program',ActionOnFailure=CONTINUE,Args=--impala-script,s3://myimpala/input,--
console-output-path,s3://myimpala/output
```

必要參數：

```
Type, Args
```

選用參數：

```
Name, ActionOnFailure
```

輸出：

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddSteps。

add-tags

以下程式碼範例顯示如何使用 add-tags。

AWS CLI

1。將標籤新增至叢集

命令：

```
aws emr add-tags --resource-id j-xxxxxxx --tags name="John Doe" age=29 sex=male
 address="123 East NW Seattle"
```

輸出：

```
None
```

2. 列出叢集的標籤

--命令：

```
aws emr describe-cluster --cluster-id j-XXXXXXYY --query Cluster.Tags
```

輸出：

```
[
    {
        "Value": "male",
        "Key": "sex"
    },
    {
        "Value": "123 East NW Seattle",
        "Key": "address"
    },
    {
        "Value": "John Doe",
        "Key": "name"
    },
```

```
    {
        "Value": "29",
        "Key": "age"
    }
]
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddTags。

create-cluster-examples

以下程式碼範例顯示如何使用 create-cluster-examples。

AWS CLI

下列大多數範例假設您已指定 Amazon EMR 服務角色和 Amazon EC2 執行個體描述檔。如果您尚未這麼做，則必須指定每個必要的 IAM 角色，或在建立叢集時使用 --use-default-roles 參數。如需指定 IAM 角色的詳細資訊，請參閱《Amazon EMR 管理指南》中的設定 Amazon EMR AWS 服務許可的 IAM 角色。

範例 1：建立叢集

下列create-cluster範例會建立簡單的 EMR 叢集。

```
aws emr create-cluster \
    --release-label emr-5.14.0 \
    --instance-type m4.large \
    --instance-count 2
```

此命令不會產生輸出。

範例 2：使用預設 ServiceRole 和 InstanceProfile 角色建立 Amazon EMR 叢集

下列create-cluster範例會建立使用 --instance-groups組態的 Amazon EMR 叢集。

```
aws emr create-cluster \
    --release-label emr-5.14.0 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

範例 3：建立使用執行個體機群的 Amazon EMR 叢集

下列create-cluster範例會建立使用 --instance-fleets組態的 Amazon EMR 叢集，為每個機群指定兩個執行個體類型，以及兩個 EC2 子網路。

```
aws emr create-cluster \
    --release-label emr-5.14.0 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,SubnetIds=['subnet-
ab12345c','subnet-de67890f'] \
    --instance-fleets
 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m4.la
 InstanceFleetType=CORE,TargetSpotCapacity=11,InstanceTypeConfigs=['{InstanceType=m4.large,B
```

範例 4：建立具有預設角色的叢集

下列create-cluster範例使用 --use-default-roles 參數來指定預設服務角色和執行個體描述檔。

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --use-default-roles \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

範例 5：建立叢集並指定要安裝的應用程式

下列create-cluster範例使用 --applications 參數來指定 Amazon EMR 安裝的應用程式。此範例會安裝 Hadoop、Hive 和 Pig。

```
aws emr create-cluster \
    --applications Name=Hadoop Name=Hive Name=Pig \
    --release-label emr-5.9.0 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

範例 6：建立包含 Spark 的叢集

下列範例會安裝 Spark。

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --applications Name=Spark \
    --ec2-attributes KeyName=myKey \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

範例 7：指定要用於叢集執行個體的自訂 AMI

下列create-cluster範例會根據 ID 為 的 Amazon Linux AMI 建立叢集執行個體ami-a518e6df。

```
aws emr create-cluster \
    --name "Cluster with My Custom AMI" \
    --custom-ami-id ami-a518e6df \
    --ebs-root-volume-size 20 \
    --release-label emr-5.9.0 \
    --use-default-roles \
    --instance-count 2 \
    --instance-type m4.large
```

範例 8：自訂應用程式組態

下列範例使用 --configurations 參數來指定包含 Hadoop 應用程式自訂的 JSON 組態檔案。如需詳細資訊，請參閱《Amazon EMR 版本指南》中的設定應用程式。

configurations.json 的內容：

```
[
    {
       "Classification": "mapred-site",
       "Properties": {
           "mapred.tasktracker.map.tasks.maximum": 2
       }
    },
    {
        "Classification": "hadoop-env",
        "Properties": {},
        "Configurations": [
            {
```

```
                "Classification": "export",
                "Properties": {
                    "HADOOP_DATANODE_HEAPSIZE": 2048,
                    "HADOOP_NAMENODE_OPTS": "-XX:GCTimeRatio=19"
                }
            }
        ]
    }
]
```

下列範例參考 configurations.json做為本機檔案。

```
aws emr create-cluster \
    --configurations file://configurations.json \
    --release-label emr-5.9.0 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

下列範例參考 configurations.json Amazon S3 中的檔案。

```
aws emr create-cluster \
    --configurations https://s3.amazonaws.com/amzn-s3-demo-bucket/
configurations.json \
    --release-label emr-5.9.0 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

範例 9：建立具有主要、核心和任務執行個體群組的叢集

下列create-cluster範例使用 --instance-groups指定用於主要、核心和任務執行個體群組的 EC2 執行個體類型和數量。

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --instance-
groups Name=Master,InstanceGroupType=MASTER,InstanceType=m4.large,InstanceCount=1 Name=Core,
```

範例 10：指定叢集應在完成所有步驟後終止

下列create-cluster範例使用 --auto-terminate指定叢集應在完成所有步驟後自動關閉。

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large
  InstanceGroupType=CORE,InstanceCount=2,InstanceType=m4.large \
    --auto-terminate
```

範例 11：指定叢集組態詳細資訊，例如 Amazon EC2 金鑰對、網路組態和安全群組

下列create-cluster範例會使用名為 的 Amazon EC2 金鑰對myKey和名為 的自訂執行個體描述檔來建立叢集myProfile。金鑰對用於授權叢集節點的 SSH 連線，通常是主節點。如需詳細資訊，請參閱《Amazon EMR 管理指南》中的針對 SSH 登入資料使用 Amazon EC2 金鑰對。

```
aws emr create-cluster \
    --ec2-attributes KeyName=myKey,InstanceProfile=myProfile \
    --release-label emr-5.9.0 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

下列範例會在 Amazon VPC 子網路中建立叢集。

```
aws emr create-cluster \
    --ec2-attributes SubnetId=subnet-xxxxx \
    --release-label emr-5.9.0 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

下列範例會在us-east-1b可用區域中建立叢集。

```
aws emr create-cluster \
    --ec2-attributes AvailabilityZone=us-east-1b \
    --release-label emr-5.9.0 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

下列範例會建立叢集，並僅指定 Amazon EMR 受管安全群組。

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --service-role myServiceRole \
    --ec2-attributes InstanceProfile=myRole,EmrManagedMasterSecurityGroup=sg-
master1,EmrManagedSlaveSecurityGroup=sg-slave1 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

下列範例會建立叢集，並僅指定其他 Amazon EC2 安全群組。

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --service-role myServiceRole \
    --ec2-attributes InstanceProfile=myRole,AdditionalMasterSecurityGroups=[sg-
addMaster1,sg-addMaster2,sg-addMaster3,sg-
addMaster4],AdditionalSlaveSecurityGroups=[sg-addSlave1,sg-addSlave2,sg-
addSlave3,sg-addSlave4] \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

下列範例會建立叢集，並指定 EMR 受管安全群組，以及其他安全群組。

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --service-role myServiceRole \
    --ec2-attributes InstanceProfile=myRole,EmrManagedMasterSecurityGroup=sg-
master1,EmrManagedSlaveSecurityGroup=sg-slave1,AdditionalMasterSecurityGroups=[sg-
addMaster1,sg-addMaster2,sg-addMaster3,sg-
addMaster4],AdditionalSlaveSecurityGroups=[sg-addSlave1,sg-addSlave2,sg-
addSlave3,sg-addSlave4] \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

下列範例會在 VPC 私有子網路中建立叢集，並使用特定的 Amazon EC2 安全群組來啟用私有子網路中的叢集所需的 Amazon EMR 服務存取。

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --service-role myServiceRole \
    --ec2-attributes InstanceProfile=myRole,ServiceAccessSecurityGroup=sg-service-
access,EmrManagedMasterSecurityGroup=sg-master,EmrManagedSlaveSecurityGroup=sg-slave
 \
```

```
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

下列範例使用本機存放的名為 ec2_attributes.json的 JSON 檔案指定安全群組組態參數。注意：JSON 引數必須在清單中包含選項和值作為自己的項目。

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --service-role myServiceRole \
    --ec2-attributes file://ec2_attributes.json  \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

ec2_attributes.json 的內容：

```
[
    {
        "SubnetId": "subnet-xxxxx",
        "KeyName": "myKey",
        "InstanceProfile":"myRole",
        "EmrManagedMasterSecurityGroup": "sg-master1",
        "EmrManagedSlaveSecurityGroup": "sg-slave1",
        "ServiceAccessSecurityGroup": "sg-service-access",
        "AdditionalMasterSecurityGroups": ["sg-addMaster1","sg-addMaster2","sg-
addMaster3","sg-addMaster4"],
        "AdditionalSlaveSecurityGroups": ["sg-addSlave1","sg-addSlave2","sg-
addSlave3","sg-addSlave4"]
    }
]
```

範例 12：啟用偵錯並指定日誌 URI

下列create-cluster範例使用 --enable-debugging 參數，可讓您使用 Amazon EMR 主控台中的偵錯工具，更輕鬆地檢視日誌檔案。參數是 的必要--log-uri項目--enable-debugging。

```
aws emr create-cluster \
    --enable-debugging \
    --log-uri s3://amzn-s3-demo-bucket/myLog \
    --release-label emr-5.9.0 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
```

```
    --auto-terminate
```

範例 13：在建立叢集時新增標籤

標籤是索引鍵/值對，可協助您識別和管理叢集。下列create-cluster範例使用 --tags 參數為叢集建立三個標籤，一個具有金鑰名稱 name和值 Shirley Rodriguez，第二個具有金鑰名稱 age 和值 29，第三個標籤具有金鑰名稱 department和值 Analytics。

```
aws emr create-cluster \
    --tags name="Shirley Rodriguez" age=29 department="Analytics" \
    --release-label emr-5.32.0 \
    --instance-type m5.xlarge \
    --instance-count 3 \
    --use-default-roles
```

下列範例列出套用至叢集的標籤。

```
aws emr describe-cluster \
    --cluster-id j-XXXXXXYY \
    --query Cluster.Tags
```

範例 14：使用啟用加密和其他安全功能的安全組態

下列create-cluster範例使用 --security-configuration 參數來指定 EMR 叢集的安全組態。您可以搭配 Amazon EMR 4.8.0 版或更新版本使用安全組態。

```
aws emr create-cluster \
    --instance-type m4.large \
    --release-label emr-5.9.0 \
    --security-configuration mySecurityConfiguration
```

範例 15：使用為執行個體群組設定的其他 EBS 儲存磁碟區建立叢集

指定其他 EBS 磁碟區時，需要下列引數：VolumeType，SizeInGB如果EbsBlockDeviceConfigs指定 。

下列create-cluster範例會建立叢集，其中多個 EBS 磁碟區連接到核心執行個體群組中的 EC2 執行個體。

```
aws emr create-cluster \
    --release-label emr-5.9.0  \
    --use-default-roles \
```

```
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=d2.xlarge
 'InstanceGroupType=CORE,InstanceCount=2,InstanceType=d2.xlarge,EbsConfiguration={EbsOptimiz
{VolumeSpecification={VolumeType=io1,SizeInGB=100,Iops=100},VolumesPerInstance=4}]}'
 \
    --auto-terminate
```

下列範例會建立叢集，其中有多個 EBS 磁碟區連接到主要執行個體群組中的 EC2 執行個體。

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --use-default-roles \
    --instance-groups 'InstanceGroupType=MASTER, InstanceCount=1,
 InstanceType=d2.xlarge, EbsConfiguration={EbsOptimized=true,
 EbsBlockDeviceConfigs=[{VolumeSpecification={VolumeType=io1, SizeInGB=100,
 Iops=100}},
{VolumeSpecification={VolumeType=standard,SizeInGB=50},VolumesPerInstance=3}]}' InstanceGrou
 \
    --auto-terminate
```

範例 16：使用自動擴展政策建立叢集

您可以使用 Amazon EMR 4.0 版及更新版本，將自動擴展政策連接至核心和任務執行個體群組。自動擴展政策會動態新增和移除 EC2 執行個體，以回應 Amazon CloudWatch 指標。如需詳細資訊，請參閱《Amazon EMR 管理指南》中的在 Amazon EMR <https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-automatic-scaling.html>`_ 中使用自動擴展。

連接自動擴展政策時，您還必須使用 指定自動擴展的預設角色`--auto-scaling-role EMR_AutoScaling_DefaultRole`。

下列`create-cluster`範例使用具有內嵌 JSON 結構的 `AutoScalingPolicy` 引數指定`CORE`執行個體群組的自動擴展政策，該引數指定擴展政策組態。具有內嵌 JSON 結構的執行個體群組必須具有以單引號括住的整個引數集合。對於沒有內嵌 JSON 結構的執行個體群組，使用單引號是選用的。

```
aws emr create-cluster
    --release-label emr-5.9.0 \
    --use-default-roles --auto-scaling-role EMR_AutoScaling_DefaultRole \
    --instance-
groups InstanceGroupType=MASTER,InstanceType=d2.xlarge,InstanceCount=1
 'InstanceGroupType=CORE,InstanceType=d2.xlarge,InstanceCount=2,AutoScalingPolicy={Constrain
```

下列範例使用 JSON 檔案 instancegroupconfig.json來指定叢集中所有執行個體群組的組態。JSON 檔案會指定核心執行個體群組的自動擴展政策組態。

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
    --instance-groups file://myfolder/instancegroupconfig.json \
    --auto-scaling-role EMR_AutoScaling_DefaultRole
```

instancegroupconfig.json 的內容：

```
[
    {
        "InstanceCount": 1,
        "Name": "MyMasterIG",
        "InstanceGroupType": "MASTER",
        "InstanceType": "m4.large"
    },
    {
        "InstanceCount": 2,
        "Name": "MyCoreIG",
        "InstanceGroupType": "CORE",
        "InstanceType": "m4.large",
        "AutoScalingPolicy": {
            "Constraints": {
                "MinCapacity": 2,
                "MaxCapacity": 10
            },
            "Rules": [
                {
                    "Name": "Default-scale-out",
                    "Description": "Replicates the default scale-out rule in the
 console for YARN memory.",
                    "Action": {
                        "SimpleScalingPolicyConfiguration": {
                            "AdjustmentType": "CHANGE_IN_CAPACITY",
                            "ScalingAdjustment": 1,
                            "CoolDown": 300
                        }
                    },
                    "Trigger": {
                        "CloudWatchAlarmDefinition": {
```

```
                            "ComparisonOperator": "LESS_THAN",
                            "EvaluationPeriods": 1,
                            "MetricName": "YARNMemoryAvailablePercentage",
                            "Namespace": "AWS/ElasticMapReduce",
                            "Period": 300,
                            "Threshold": 15,
                            "Statistic": "AVERAGE",
                            "Unit": "PERCENT",
                            "Dimensions": [
                                {
                                    "Key": "JobFlowId",
                                    "Value": "${emr.clusterId}"
                                }
                            ]
                        }
                    }
                }
            ]
        }
    }
]
```

範例 17：建立叢集時新增自訂 JAR 步驟

下列create-cluster範例透過指定存放在 Amazon S3 中的 JAR 檔案來新增步驟。步驟會將工作提交至叢集。JAR 檔案中定義的主要函數會在佈建 EC2 執行個體、執行任何引導操作，以及安裝應用程式後執行。步驟是使用 指定Type=CUSTOM_JAR。

自訂 JAR 步驟需要 Jar= 參數，指定 JAR 的路徑和檔案名稱。選用參數為 Type、Name、Args、ActionOnFailure和 MainClass。如果未指定主要類別，則 JAR 檔案應在Main-Class其資訊清單檔案中指定 。

```
aws emr create-cluster \
    --steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://amzn-
s3-demo-bucket/
mytest.jar,Args=arg1,arg2,arg3 Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s
amzn-s3-demo-bucket/mytest.jar,MainClass=mymainclass,Args=arg1,arg2,arg3  \
    --release-label emr-5.3.1 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

範例 18：在建立叢集時新增串流步驟

下列create-cluster範例會將串流步驟新增至叢集，該叢集會在所有步驟執行後終止。串流步驟需要參數 Type和 Args。串流步驟選用參數為 Name和 ActionOnFailure。

下列範例會指定步驟內嵌。

```
aws emr create-cluster \
    --steps Type=STREAMING,Name='Streaming Program',ActionOnFailure=CONTINUE,Args=[-
files,s3://elasticmapreduce/samples/wordcount/wordSplitter.py,-
mapper,wordSplitter.py,-reducer,aggregate,-input,s3://elasticmapreduce/samples/
wordcount/input,-output,s3://amzn-s3-demo-bucket/wordcount/output] \
    --release-label emr-5.3.1 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

下列範例使用名為 的本機儲存 JSON 組態檔案multiplefiles.json。JSON 組態會指定多個檔案。若要在步驟中指定多個檔案，您必須使用 JSON 組態檔案來指定步驟。JSON 引數必須在清單中包含選項和值做為自己的項目。

```
aws emr create-cluster \
    --steps file://./multiplefiles.json \
    --release-label emr-5.9.0  \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

multiplefiles.json 的內容：

```
[
    {
        "Name": "JSON Streaming Step",
        "Args": [
            "-files",
            "s3://elasticmapreduce/samples/wordcount/wordSplitter.py",
            "-mapper",
            "wordSplitter.py",
            "-reducer",
```

```
            "aggregate",
            "-input",
            "s3://elasticmapreduce/samples/wordcount/input",
            "-output",
            "s3://amzn-s3-demo-bucket/wordcount/output"
        ],
        "ActionOnFailure": "CONTINUE",
        "Type": "STREAMING"
    }
]
```

範例 19：在建立叢集時新增 Hive 步驟

下列範例會在建立叢集時新增 Hive 步驟。Hive 步驟需要參數 Type和 Args。Hive 步驟選用參數為 Name和 ActionOnFailure。

```
aws emr create-cluster \
    --steps Type=HIVE,Name='Hive
 program',ActionOnFailure=CONTINUE,ActionOnFailure=TERMINATE_CLUSTER,Args=[-
f,s3://elasticmapreduce/samples/hive-ads/libs/model-build.q,-d,INPUT=s3://
elasticmapreduce/samples/hive-ads/tables,-d,OUTPUT=s3://amzn-s3-demo-bucket/hive-
ads/output/2014-04-18/11-07-32,-d,LIBS=s3://elasticmapreduce/samples/hive-ads/libs]
 \
    --applications Name=Hive \
    --release-label emr-5.3.1 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

範例 20：在建立叢集時新增 Pig 步驟

下列範例會在建立叢集時新增 Pig 步驟。Pig 步驟所需的參數為 Type和 Args。Pig 步驟選用參數為 Name和 ActionOnFailure。

```
aws emr create-cluster \
    --steps Type=PIG,Name='Pig program',ActionOnFailure=CONTINUE,Args=[-f,s3://
elasticmapreduce/samples/pig-apache/do-reports2.pig,-p,INPUT=s3://elasticmapreduce/
samples/pig-apache/input,-p,OUTPUT=s3://amzn-s3-demo-bucket/pig-apache/output] \
    --applications Name=Pig \
    --release-label emr-5.3.1 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

範例 21：新增引導操作

下列create-cluster範例會執行兩個引導操作，定義為存放在 Amazon S3 中的指令碼。

```
aws emr create-cluster \
    --bootstrap-actions Path=s3://amzn-s3-demo-bucket/
myscript1,Name=BootstrapAction1,Args=[arg1,arg2] Path=s3://amzn-s3-demo-bucket/
myscript2,Name=BootstrapAction2,Args=[arg1,arg2] \
    --release-label emr-5.3.1 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

範例 22：啟用 EMRFS 一致性檢視並自訂 RetryCount 和 RetryPeriod 設定

下列create-cluster範例指定 EMRFS 一致性檢視的重試計數和重試期間。Consistent=true是必要引數。

```
aws emr create-cluster \
    --instance-type m4.large \
    --release-label emr-5.9.0 \
    --emrfs Consistent=true,RetryCount=6,RetryPeriod=30
```

下列範例會使用名為 的本機儲存 JSON 組態檔案，指定與先前範例相同的 EMRFS 組態emrfsconfig.json。

```
aws emr create-cluster \
    --instance-type m4.large \
    --release-label emr-5.9.0 \
    --emrfs file://emrfsconfig.json
```

emrfsconfig.json 的內容：

```
{
    "Consistent": true,
    "RetryCount": 6,
    "RetryPeriod": 30
}
```

範例 23：建立已設定 Kerberos 的叢集

下列create-cluster範例使用啟用 Kerberos 的安全組態建立叢集，並使用 為叢集建立 Kerberos 參數--kerberos-attributes。

下列命令會指定叢集內嵌的 Kerberos 屬性。

```
aws emr create-cluster \
    --instance-type m3.xlarge \
    --release-label emr-5.10.0 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
    --security-configuration mySecurityConfiguration \
    --kerberos-
attributes Realm=EC2.INTERNAL,KdcAdminPassword=123,CrossRealmTrustPrincipalPassword=123
```

下列命令會指定相同的屬性，但會參考名為 的本機儲存 JSON 檔案kerberos_attributes.json。在此範例中，檔案會儲存在您執行 命令的相同目錄中。您也可以參考儲存在 Amazon S3 中的組態檔案。

```
aws emr create-cluster \
    --instance-type m3.xlarge \
    --release-label emr-5.10.0 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
    --security-configuration mySecurityConfiguration \
    --kerberos-attributes file://kerberos_attributes.json
```

kerberos_attributes.json 的內容：

```
{
    "Realm": "EC2.INTERNAL",
    "KdcAdminPassword": "123",
    "CrossRealmTrustPrincipalPassword": "123",
}
```

下列create-cluster範例會建立使用 --instance-groups組態並具有受管擴展政策的 Amazon EMR 叢集。

```
aws emr create-cluster \
    --release-label emr-5.30.0 \
    --service-role EMR_DefaultRole \
```

```
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
    --managed-scaling-policy
 ComputeLimits='{MinimumCapacityUnits=2,MaximumCapacityUnits=4,UnitType=Instances}'
```

下列create-cluster範例會建立 Amazon EMR 叢集，該叢集使用 "--log-encryption-kms-key-id" 來定義用於日誌加密的 KMS 金鑰 ID。

```
aws emr create-cluster \
    --release-label emr-5.30.0 \
    --log-uri s3://amzn-s3-demo-bucket/myLog \
    --log-encryption-kms-key-id arn:aws:kms:us-east-1:110302272565:key/
dd559181-283e-45d7-99d1-66da348c4d33 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

下列create-cluster範例會建立使用「--placement-group-configs」組態的 Amazon EMR 叢集，使用SPREAD置放策略將主節點放置在 EC2 置放群組內的高可用性 (HA) 叢集中。

```
aws emr create-cluster \
    --release-label emr-5.30.0 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m4.largeInstanceGroupType=CORE,
 \
    --placement-group-configs InstanceRole=MASTER
```

下列create-cluster範例會建立使用 "--auto-termination-policy" 組態的 Amazon EMR 叢集，以放置叢集的自動閒置終止閾值。

```
aws emr create-cluster \
    --release-label emr-5.34.0 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-termination-policy IdleTimeout=100
```

下列create-cluster範例會建立使用 "--os-release-label" 的 Amazon EMR 叢集，以定義叢集啟動的 Amazon Linux 版本

```
aws emr create-cluster \
    --release-label emr-6.6.0 \
    --os-release-label 2.0.20220406.1 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

範例 24：指定 EBS 根磁碟區屬性：EMR 6.15.0 版及更新版本所建立叢集執行個體的大小、iops 和輸送量

下列create-cluster範例會建立 Amazon EMR 叢集，該叢集使用根磁碟區屬性來設定 EC2 執行個體的根磁碟區規格。

```
aws emr create-cluster \
    --name "Cluster with My Custom AMI" \
    --custom-ami-id ami-a518e6df \
    --ebs-root-volume-size 20 \
    --ebs-root-volume-iops 3000 \
    --ebs-root-volume-throughput 125 \
    --release-label emr-6.15.0 \
    --use-default-roles \
    --instance-count 2 \
    --instance-type m4.large
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateClusterExamples。

create-default-roles

以下程式碼範例顯示如何使用 create-default-roles。

AWS CLI

1。建立 EC2 的預設 IAM 角色

命令：

```
aws emr create-default-roles
```

輸出：

```
If the role already exists then the command returns nothing.

If the role does not exist then the output will be:

[
    {
        "RolePolicy": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "cloudwatch:*",
                        "dynamodb:*",
                        "ec2:Describe*",
                        "elasticmapreduce:Describe*",
                        "elasticmapreduce:ListBootstrapActions",
                        "elasticmapreduce:ListClusters",
                        "elasticmapreduce:ListInstanceGroups",
                        "elasticmapreduce:ListInstances",
                        "elasticmapreduce:ListSteps",
                        "kinesis:CreateStream",
                        "kinesis:DeleteStream",
                        "kinesis:DescribeStream",
                        "kinesis:GetRecords",
                        "kinesis:GetShardIterator",
                        "kinesis:MergeShards",
                        "kinesis:PutRecord",
                        "kinesis:SplitShard",
                        "rds:Describe*",
                        "s3:*",
                        "sdb:*",
                        "sns:*",
                        "sqs:*"
                    ],
                    "Resource": "*",
                    "Effect": "Allow"
                }
            ]
        },
        "Role": {
            "AssumeRolePolicyDocument": {
                "Version": "2008-10-17",
```

```
                "Statement": [
                    {
                        "Action": "sts:AssumeRole",
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        }
                    }
                ]
            },
            "RoleId": "AROAIQ5SIQUGL5KMYBJX6",
            "CreateDate": "2015-06-09T17:09:04.602Z",
            "RoleName": "EMR_EC2_DefaultRole",
            "Path": "/",
            "Arn": "arn:aws:iam::176430881729:role/EMR_EC2_DefaultRole"
        }
    },
    {
        "RolePolicy": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "ec2:AuthorizeSecurityGroupIngress",
                        "ec2:CancelSpotInstanceRequests",
                        "ec2:CreateSecurityGroup",
                        "ec2:CreateTags",
                        "ec2:DeleteTags",
                        "ec2:DescribeAvailabilityZones",
                        "ec2:DescribeAccountAttributes",
                        "ec2:DescribeInstances",
                        "ec2:DescribeInstanceStatus",
                        "ec2:DescribeKeyPairs",
                        "ec2:DescribePrefixLists",
                        "ec2:DescribeRouteTables",
                        "ec2:DescribeSecurityGroups",
                        "ec2:DescribeSpotInstanceRequests",
                        "ec2:DescribeSpotPriceHistory",
                        "ec2:DescribeSubnets",
                        "ec2:DescribeVpcAttribute",
                        "ec2:DescribeVpcEndpoints",
                        "ec2:DescribeVpcEndpointServices",
                        "ec2:DescribeVpcs",
```

```
                    "ec2:ModifyImageAttribute",
                    "ec2:ModifyInstanceAttribute",
                    "ec2:RequestSpotInstances",
                    "ec2:RunInstances",
                    "ec2:TerminateInstances",
                    "iam:GetRole",
                    "iam:GetRolePolicy",
                    "iam:ListInstanceProfiles",
                    "iam:ListRolePolicies",
                    "iam:PassRole",
                    "s3:CreateBucket",
                    "s3:Get*",
                    "s3:List*",
                    "sdb:BatchPutAttributes",
                    "sdb:Select",
                    "sqs:CreateQueue",
                    "sqs:Delete*",
                    "sqs:GetQueue*",
                    "sqs:ReceiveMessage"
                ],
                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    },
    "Role": {
        "AssumeRolePolicyDocument": {
            "Version": "2008-10-17",
            "Statement": [
                {
                    "Action": "sts:AssumeRole",
                    "Sid": "",
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "elasticmapreduce.amazonaws.com"
                    }
                }
            ]
        },
        "RoleId": "AROAI3SRVPPVSRDLARBPY",
        "CreateDate": "2015-06-09T17:09:10.401Z",
        "RoleName": "EMR_DefaultRole",
        "Path": "/",
        "Arn": "arn:aws:iam::176430881729:role/EMR_DefaultRole"
```

```
        }
    }
]
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDefaultRoles。

create-security-configuration

以下程式碼範例顯示如何使用 create-security-configuration。

AWS CLI

1。若要建立啟用 PEM 的憑證提供者傳輸中加密的安全組態，以及啟用 SSE-S3 的靜態加密 S3 加密和本機磁碟金鑰提供者的 AWS KMS

命令：

```
 aws emr create-security-configuration --name MySecurityConfig --security-
configuration '{
        "EncryptionConfiguration": {
                "EnableInTransitEncryption" : true,
                "EnableAtRestEncryption" : true,
                "InTransitEncryptionConfiguration" : {
                        "TLSCertificateConfiguration" : {
                                "CertificateProviderType" : "PEM",
                                "S3Object" : "s3://mycertstore/artifacts/
MyCerts.zip"
                        }
                },
                "AtRestEncryptionConfiguration" : {
                        "S3EncryptionConfiguration" : {
                                "EncryptionMode" : "SSE-S3"
                        },
                        "LocalDiskEncryptionConfiguration" : {
                                "EncryptionKeyProviderType" : "AwsKms",
                                "AwsKmsKey" : "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
                        }
                }
        }
}'
```

輸出：

```
{
"CreationDateTime": 1474070889.129,
"Name": "MySecurityConfig"
}
```

JSON 對等 (Security_configuration.json 的內容）：

```
{
    "EncryptionConfiguration": {
        "EnableInTransitEncryption": true,
        "EnableAtRestEncryption": true,
        "InTransitEncryptionConfiguration": {
            "TLSCertificateConfiguration": {
                "CertificateProviderType": "PEM",
                "S3Object": "s3://mycertstore/artifacts/MyCerts.zip"
            }
        },
        "AtRestEncryptionConfiguration": {
            "S3EncryptionConfiguration": {
                "EncryptionMode": "SSE-S3"
            },
            "LocalDiskEncryptionConfiguration": {
                "EncryptionKeyProviderType": "AwsKms",
                "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
            }
        }
    }
}
```

命令（使用 security_configuration.json)：

```
aws emr create-security-configuration --name "MySecurityConfig" --security-
configuration file://./security_configuration.json
```

輸出：

```
{
"CreationDateTime": 1474070889.129,
"Name": "MySecurityConfig"
}
```

2. 使用叢集專用 KDC 和跨領域信任建立已啟用 Kerberos 的安全組態

命令：

```
 aws emr create-security-configuration --name MySecurityConfig --security-
configuration '{
     "AuthenticationConfiguration": {
         "KerberosConfiguration": {
             "Provider": "ClusterDedicatedKdc",
             "ClusterDedicatedKdcConfiguration": {
                 "TicketLifetimeInHours": 24,
                 "CrossRealmTrustConfiguration": {
                   "Realm": "AD.DOMAIN.COM",
                   "Domain": "ad.domain.com",
                   "AdminServer": "ad.domain.com",
                   "KdcServer": "ad.domain.com"
                 }
             }
         }
     }
}'
```

輸出：

```
{
"CreationDateTime": 1490225558.982,
"Name": "MySecurityConfig"
}
```

JSON 對等 (Security_configuration.json 的內容) ：

```
{
    "AuthenticationConfiguration": {
        "KerberosConfiguration": {
            "Provider": "ClusterDedicatedKdc",
            "ClusterDedicatedKdcConfiguration": {
                "TicketLifetimeInHours": 24,
                "CrossRealmTrustConfiguration": {
                    "Realm": "AD.DOMAIN.COM",
                    "Domain": "ad.domain.com",
                    "AdminServer": "ad.domain.com",
                    "KdcServer": "ad.domain.com"
                }
```

```
            }
        }
    }
}
```

命令（使用 security_configuration.json）：

```
aws emr create-security-configuration --name "MySecurityConfig" --security-
configuration file://./security_configuration.json
```

輸出：

```
{
"CreationDateTime": 1490225558.982,
"Name": "MySecurityConfig"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSecurityConfiguration。

delete-security-configuration

以下程式碼範例顯示如何使用 delete-security-configuration。

AWS CLI

刪除目前區域中的安全組態

命令：

```
aws emr delete-security-configuration --name MySecurityConfig
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSecurityConfiguration。

describe-cluster

以下程式碼範例顯示如何使用 describe-cluster。

AWS CLI

命令：

```
aws emr describe-cluster --cluster-id j-XXXXXXXX
```

輸出：

```
For release-label based uniform instance groups cluster:

        {
            "Cluster": {
                "Status": {
                    "Timeline": {
                        "ReadyDateTime": 1436475075.199,
                        "CreationDateTime": 1436474656.563,
                    },
                    "State": "WAITING",
                    "StateChangeReason": {
                        "Message": "Waiting for steps to run"
                    }
                },
                "Ec2InstanceAttributes": {
                    "ServiceAccessSecurityGroup": "sg-xxxxxxxx",
                    "EmrManagedMasterSecurityGroup": "sg-xxxxxxxx",
                    "IamInstanceProfile": "EMR_EC2_DefaultRole",
                    "Ec2KeyName": "myKey",
                    "Ec2AvailabilityZone": "us-east-1c",
                    "EmrManagedSlaveSecurityGroup": "sg-yyyyyyyyy"
                },
                "Name": "My Cluster",
                "ServiceRole": "EMR_DefaultRole",
                "Tags": [],
                "TerminationProtected": true,
                "UnhealthyNodeReplacement": true,
                "ReleaseLabel": "emr-4.0.0",
                "NormalizedInstanceHours": 96,
                "InstanceGroups": [
                    {
                        "RequestedInstanceCount": 2,
                        "Status": {
                            "Timeline": {
                                "ReadyDateTime": 1436475074.245,
```

```
                            "CreationDateTime": 1436474656.564,
                            "EndDateTime": 1436638158.387
                        },
                        "State": "RUNNING",
                        "StateChangeReason": {
                            "Message": "",
                        }
                    },
                    "Name": "CORE",
                    "InstanceGroupType": "CORE",
                    "Id": "ig-YYYYYYY",
                    "Configurations": [],
                    "InstanceType": "m3.large",
                    "Market": "ON_DEMAND",
                    "RunningInstanceCount": 2
                },
                {
                    "RequestedInstanceCount": 1,
                    "Status": {
                        "Timeline": {
                            "ReadyDateTime": 1436475074.245,
                            "CreationDateTime": 1436474656.564,
                            "EndDateTime": 1436638158.387
                        },
                        "State": "RUNNING",
                        "StateChangeReason": {
                            "Message": "",
                        }
                    },
                    "Name": "MASTER",
                    "InstanceGroupType": "MASTER",
                    "Id": "ig-XXXXXXXXX",
                    "Configurations": [],
                    "InstanceType": "m3.large",
                    "Market": "ON_DEMAND",
                    "RunningInstanceCount": 1
                }
            ],
            "Applications": [
                {
                    "Name": "Hadoop"
                }
            ],
            "VisibleToAllUsers": true,
```

```
                "BootstrapActions": [],
                "MasterPublicDnsName": "ec2-54-147-144-78.compute-1.amazonaws.com",
                "AutoTerminate": false,
                "Id": "j-XXXXXXXX",
                "Configurations": [
                    {
                        "Properties": {
                            "fs.s3.consistent.retryPeriodSeconds": "20",
                            "fs.s3.enableServerSideEncryption": "true",
                            "fs.s3.consistent": "false",
                            "fs.s3.consistent.retryCount": "2"
                        },
                        "Classification": "emrfs-site"
                    }
                ]
            }
        }


For release-label based instance fleet cluster:
{
    "Cluster": {
        "Status": {
            "Timeline": {
                "ReadyDateTime": 1487897289.705,
                "CreationDateTime": 1487896933.942
            },
            "State": "WAITING",
            "StateChangeReason": {
                "Message": "Waiting for steps to run"
            }
        },
        "Ec2InstanceAttributes": {
            "EmrManagedMasterSecurityGroup": "sg-xxxxx",
            "RequestedEc2AvailabilityZones": [],
            "RequestedEc2SubnetIds": [],
            "IamInstanceProfile": "EMR_EC2_DefaultRole",
            "Ec2AvailabilityZone": "us-east-1a",
            "EmrManagedSlaveSecurityGroup": "sg-xxxxx"
        },
        "Name": "My Cluster",
        "ServiceRole": "EMR_DefaultRole",
        "Tags": [],
        "TerminationProtected": false,
```

```
        "UnhealthyNodeReplacement": false,
        "ReleaseLabel": "emr-5.2.0",
        "NormalizedInstanceHours": 472,
        "InstanceCollectionType": "INSTANCE_FLEET",
        "InstanceFleets": [
            {
                "Status": {
                    "Timeline": {
                        "ReadyDateTime": 1487897212.74,
                        "CreationDateTime": 1487896933.948
                    },
                    "State": "RUNNING",
                    "StateChangeReason": {
                        "Message": ""
                    }
                },
                "ProvisionedSpotCapacity": 1,
                "Name": "MASTER",
                "InstanceFleetType": "MASTER",
                "LaunchSpecifications": {
                    "SpotSpecification": {
                        "TimeoutDurationMinutes": 60,
                        "TimeoutAction": "TERMINATE_CLUSTER"
                    }
                },
                "TargetSpotCapacity": 1,
                "ProvisionedOnDemandCapacity": 0,
                "InstanceTypeSpecifications": [
                    {
                        "BidPrice": "0.5",
                        "InstanceType": "m3.xlarge",
                        "WeightedCapacity": 1
                    }
                ],
                "Id": "if-xxxxxxx",
                "TargetOnDemandCapacity": 0
            }
        ],
        "Applications": [
            {
                "Version": "2.7.3",
                "Name": "Hadoop"
            }
        ],
```

```
        "ScaleDownBehavior": "TERMINATE_AT_INSTANCE_HOUR",
        "VisibleToAllUsers": true,
        "BootstrapActions": [],
        "MasterPublicDnsName": "ec2-xxx-xx-xxx-xx.compute-1.amazonaws.com",
        "AutoTerminate": false,
        "Id": "j-xxxxx",
        "Configurations": []
    }
}

For ami based uniform instance group cluster:

    {
        "Cluster": {
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1399400564.432,
                    "CreationDateTime": 1399400268.62
                },
                "State": "WAITING",
                "StateChangeReason": {
                    "Message": "Waiting for steps to run"
                }
            },
            "Ec2InstanceAttributes": {
                "IamInstanceProfile": "EMR_EC2_DefaultRole",
                "Ec2AvailabilityZone": "us-east-1c"
            },
            "Name": "My Cluster",
            "Tags": [],
            "TerminationProtected": true,
            "UnhealthyNodeReplacement": true,
            "RunningAmiVersion": "2.5.4",
            "InstanceGroups": [
                {
                    "RequestedInstanceCount": 1,
                    "Status": {
                        "Timeline": {
                            "ReadyDateTime": 1399400558.848,
                            "CreationDateTime": 1399400268.621
                        },
                        "State": "RUNNING",
                        "StateChangeReason": {
                            "Message": ""
```

```
                    }
                },
                "Name": "Master instance group",
                "InstanceGroupType": "MASTER",
                "InstanceType": "m1.small",
                "Id": "ig-ABCD",
                "Market": "ON_DEMAND",
                "RunningInstanceCount": 1
            },
            {
                "RequestedInstanceCount": 2,
                "Status": {
                    "Timeline": {
                        "ReadyDateTime": 1399400564.439,
                        "CreationDateTime": 1399400268.621
                    },
                    "State": "RUNNING",
                    "StateChangeReason": {
                        "Message": ""
                    }
                },
                "Name": "Core instance group",
                "InstanceGroupType": "CORE",
                "InstanceType": "m1.small",
                "Id": "ig-DEF",
                "Market": "ON_DEMAND",
                "RunningInstanceCount": 2
            }
        ],
        "Applications": [
            {
                "Version": "1.0.3",
                "Name": "hadoop"
            }
        ],
        "BootstrapActions": [],
        "VisibleToAllUsers": false,
        "RequestedAmiVersion": "2.4.2",
        "LogUri": "s3://myLogUri/",
        "AutoTerminate": false,
        "Id": "j-XXXXXXXX"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCluster。

describe-step

以下程式碼範例顯示如何使用 describe-step。

AWS CLI

下列命令說明叢集s-3LZC0QUT43AM中步驟 ID 為 且叢集 ID 為 的步驟j-3SD91U2E1L2QX：

```
aws emr describe-step --cluster-id j-3SD91U2E1L2QX --step-id s-3LZC0QUT43AM
```

輸出：

```
{
    "Step": {
        "Status": {
            "Timeline": {
                "EndDateTime": 1433200470.481,
                "CreationDateTime": 1433199926.597,
                "StartDateTime": 1433200404.959
            },
            "State": "COMPLETED",
            "StateChangeReason": {}
        },
        "Config": {
            "Args": [
                "s3://us-west-2.elasticmapreduce/libs/hive/hive-script",
                "--base-path",
                "s3://us-west-2.elasticmapreduce/libs/hive/",
                "--install-hive",
                "--hive-versions",
                "0.13.1"
            ],
            "Jar": "s3://us-west-2.elasticmapreduce/libs/script-runner/script-
runner.jar",
            "Properties": {}
        },
        "Id": "s-3LZC0QUT43AM",
        "ActionOnFailure": "TERMINATE_CLUSTER",
        "Name": "Setup hive"
    }
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStep。

get

以下程式碼範例顯示如何使用 get。

AWS CLI

以下內容會從叢集 ID 為 的叢集中主執行個體下載hadoop-examples.jar封存j-3SD91U2E1L2QX：

```
aws emr get --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem --src /
home/hadoop-examples.jar --dest ~
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的取得。

list-clusters

以下程式碼範例顯示如何使用 list-clusters。

AWS CLI

下列命令會列出目前區域中所有作用中的 EMR 叢集：

```
aws emr list-clusters --active
```

輸出：

```
{
    "Clusters": [
        {
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1433200405.353,
                    "CreationDateTime": 1433199926.596
                },
                "State": "WAITING",
                "StateChangeReason": {
                    "Message": "Waiting after step completed"
```

```
                }
            },
            "NormalizedInstanceHours": 6,
            "Id": "j-3SD91U2E1L2QX",
            "Name": "my-cluster"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListClusters。

`list-instance-fleets`

以下程式碼範例顯示如何使用 `list-instance-fleets`。

AWS CLI

取得叢集中執行個體機群的組態詳細資訊

此範例列出指定叢集中執行個體機群的詳細資訊。

命令：

```
list-instance-fleets --cluster-id 'j-12ABCDEFGHI34JK'
```

輸出：

```
{
  "InstanceFleets": [
      {
          "Status": {
              "Timeline": {
                  "ReadyDateTime": 1488759094.637,
                  "CreationDateTime": 1488758719.817
              },
              "State": "RUNNING",
              "StateChangeReason": {
                  "Message": ""
              }
          },
          "ProvisionedSpotCapacity": 6,
          "Name": "CORE",
          "InstanceFleetType": "CORE",
```

```
            "LaunchSpecifications": {
                "SpotSpecification": {
                    "TimeoutDurationMinutes": 60,
                    "TimeoutAction": "TERMINATE_CLUSTER"
                }
            },
            "ProvisionedOnDemandCapacity": 2,
            "InstanceTypeSpecifications": [
                {
                    "BidPrice": "0.5",
                    "InstanceType": "m3.xlarge",
                    "WeightedCapacity": 2
                }
            ],
            "Id": "if-1ABC2DEFGHIJ3"
        },
        {
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1488759058.598,
                    "CreationDateTime": 1488758719.811
                },
                "State": "RUNNING",
                "StateChangeReason": {
                    "Message": ""
                }
            },
            "ProvisionedSpotCapacity": 0,
            "Name": "MASTER",
            "InstanceFleetType": "MASTER",
            "ProvisionedOnDemandCapacity": 1,
            "InstanceTypeSpecifications": [
                {
                    "BidPriceAsPercentageOfOnDemandPrice": 100.0,
                    "InstanceType": "m3.xlarge",
                    "WeightedCapacity": 1
                }
            ],
           "Id": "if-2ABC4DEFGHIJ4"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListInstanceFleets。

list-instances

以下程式碼範例顯示如何使用 list-instances。

AWS CLI

下列命令會列出叢集中叢集 ID 為 的所有執行個體j-3C6XNQ39VR9WL：

```
aws emr list-instances --cluster-id j-3C6XNQ39VR9WL
```

輸出：

```
For a uniform instance group based cluster
  {
    "Instances": [
         {
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1433200400.03,
                    "CreationDateTime": 1433199960.152
                },
                "State": "RUNNING",
                "StateChangeReason": {}
            },
            "Ec2InstanceId": "i-f19ecfee",
            "PublicDnsName": "ec2-52-52-41-150.us-west-2.compute.amazonaws.com",
            "PrivateDnsName": "ip-172-21-11-216.us-west-2.compute.internal",
            "PublicIpAddress": "52.52.41.150",
            "Id": "ci-3NNHQUQ2TWB6Y",
            "PrivateIpAddress": "172.21.11.216"
        },
        {
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1433200400.031,
                    "CreationDateTime": 1433199949.102
                },
                "State": "RUNNING",
                "StateChangeReason": {}
            },
            "Ec2InstanceId": "i-1feee4c2",
            "PublicDnsName": "ec2-52-63-246-32.us-west-2.compute.amazonaws.com",
            "PrivateDnsName": "ip-172-31-24-130.us-west-2.compute.internal",
```

```
            "PublicIpAddress": "52.63.246.32",
            "Id": "ci-GAOCMKNKDCV7",
            "PrivateIpAddress": "172.21.11.215"
        },
        {
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1433200400.031,
                    "CreationDateTime": 1433199949.102
                },
                "State": "RUNNING",
                "StateChangeReason": {}
            },
            "Ec2InstanceId": "i-15cfeee3",
            "PublicDnsName": "ec2-52-25-246-63.us-west-2.compute.amazonaws.com",
            "PrivateDnsName": "ip-172-31-24-129.us-west-2.compute.internal",
            "PublicIpAddress": "52.25.246.63",
            "Id": "ci-2W3TDFFB47UAD",
            "PrivateIpAddress": "172.21.11.214"
        }
    ]
  }


For a fleet based cluster:
   {
      "Instances": [
          {
              "Status": {
                  "Timeline": {
                      "ReadyDateTime": 1487810810.878,
                      "CreationDateTime": 1487810588.367,
                      "EndDateTime": 1488022990.924
                  },
                  "State": "TERMINATED",
                  "StateChangeReason": {
                      "Message": "Instance was terminated."
                  }
              },
              "Ec2InstanceId": "i-xxxxx",
              "InstanceFleetId": "if-xxxxx",
              "EbsVolumes": [],
              "PublicDnsName": "ec2-xx-xxx-xxx-xxx.compute-1.amazonaws.com",
              "InstanceType": "m3.xlarge",
```

```
            "PrivateDnsName": "ip-xx-xx-xxx-xx.ec2.internal",
            "Market": "SPOT",
            "PublicIpAddress": "xx.xx.xxx.xxx",
            "Id": "ci-xxxxx",
            "PrivateIpAddress": "10.47.191.80"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListInstances。

list-security-configurations

以下程式碼範例顯示如何使用 list-security-configurations。

AWS CLI

列出目前區域中的安全組態

命令：

```
aws emr list-security-configurations
```

輸出：

```
{
    "SecurityConfigurations": [
        {
            "CreationDateTime": 1473889697.417,
            "Name": "MySecurityConfig-1"
        },
        {
            "CreationDateTime": 1473889697.417,
            "Name": "MySecurityConfig-2"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSecurityConfigurations。

list-steps

以下程式碼範例顯示如何使用 list-steps。

AWS CLI

下列命令會列出叢集中具有叢集 ID 的所有步驟j-3SD91U2E1L2QX：

```
aws emr list-steps --cluster-id j-3SD91U2E1L2QX
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSteps。

modify-cluster-attributes

以下程式碼範例顯示如何使用 modify-cluster-attributes。

AWS CLI

下列命令會將 ID 為 的 EMR 叢集可見性j-301CDNY0J5XM4設定為所有使用者：

```
aws emr modify-cluster-attributes --cluster-id j-301CDNY0J5XM4 --visible-to-all-
users
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyClusterAttributes。

modify-instance-fleet

以下程式碼範例顯示如何使用 modify-instance-fleet。

AWS CLI

變更執行個體機群的目標容量

此範例會將指定執行個體機群的隨需和 Spot 目標容量變更為 1。

命令：

```
aws emr modify-instance-fleet --cluster-id 'j-12ABCDEFGHI34JK' --instance-fleet
 InstanceFleetId='if-2ABC4DEFGHIJ4',TargetOnDemandCapacity=1,TargetSpotCapacity=1
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyInstanceFleet。

put

以下程式碼範例顯示如何使用 put。

AWS CLI

下列命令會將名為 的檔案上傳至叢集中叢集 ID 為 healthcheck.sh的主執行個體j-3SD91U2E1L2QX：

```
aws emr put --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem --src ~/scripts/healthcheck.sh --dest /home/hadoop/bin/healthcheck.sh
```

- 如需 API 詳細資訊，請參閱 Put in AWS CLI Command Reference。

remove-tags

以下程式碼範例顯示如何使用 remove-tags。

AWS CLI

下列命令prod會從叢集 ID 為 的叢集中移除具有 金鑰的標籤j-3SD91U2E1L2QX：

```
aws emr remove-tags --resource-id j-3SD91U2E1L2QX --tag-keys prod
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveTags。

schedule-hbase-backup

以下程式碼範例顯示如何使用 schedule-hbase-backup。

AWS CLI

注意：此命令只能與 AMI 2.x 版和 3.x 版上的 HBase 搭配使用

1。排程完整的 HBase 備份>>>> 06ab6d6e13564b5733d75abaf3b599f93cf39a23

命令：

```
aws emr schedule-hbase-backup --cluster-id j-XXXXXXYY --type full --dir
s3://amzn-s3-demo-bucket/backup --interval 10 --unit hours --start-time
2014-04-21T05:26:10Z --consistent
```

輸出：

```
None
```

2. 排程增量 HBase 備份

命令：

```
aws emr schedule-hbase-backup --cluster-id j-XXXXXXYY --type incremental
 --dir s3://amzn-s3-demo-bucket/backup --interval 30 --unit minutes --start-time
2014-04-21T05:26:10Z --consistent
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ScheduleHbaseBackup。

socks

以下程式碼範例顯示如何使用 socks。

AWS CLI

下列命令會開啟與叢集 ID 為 之叢集中主要執行個體的襪子連線j-3SD91U2E1L2QX：

```
aws emr socks --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem
```

金鑰對檔案選項會取得私有金鑰檔案的本機路徑。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的襪子。

ssh

以下程式碼範例顯示如何使用 ssh。

AWS CLI

下列命令會開啟與叢集 ID 為 之叢集中主要執行個體的 ssh 連線j-3SD91U2E1L2QX：

```
aws emr ssh --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem
```

金鑰對檔案選項會取得私有金鑰檔案的本機路徑。

輸出：

```
ssh -o StrictHostKeyChecking=no -o ServerAliveInterval=10 -i /home/local/user/.ssh/
mykey.pem hadoop@ec2-52-52-41-150.us-west-2.compute.amazonaws.com
Warning: Permanently added 'ec2-52-52-41-150.us-
west-2.compute.amazonaws.com,52.52.41.150' (ECDSA) to the list of known hosts.
Last login: Mon Jun  1 23:15:38 2015

      __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2015.03-release-notes/
26 package(s) needed for security, out of 39 available
Run "sudo yum update" to apply all updates.

--------------------------------------------------------------------------------

Welcome to Amazon Elastic MapReduce running Hadoop and Amazon Linux.

Hadoop is installed in /home/hadoop. Log files are in /mnt/var/log/hadoop. Check
/mnt/var/log/hadoop/steps for diagnosing step failures.

The Hadoop UI can be accessed via the following commands:

  ResourceManager    lynx http://ip-172-21-11-216:9026/
  NameNode           lynx http://ip-172-21-11-216:9101/

--------------------------------------------------------------------------------

[hadoop@ip-172-31-16-216 ~]$
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 Ssh。

使用 的 Amazon EMR on EKS 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon EMR on EKS 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`update-role-trust-policy`

以下程式碼範例顯示如何使用 `update-role-trust-policy`。

AWS CLI

更新要與 Amazon EMR on EKS 搭配使用之 IAM 角色的信任政策

此範例命令會更新名為 example_iam_role 之角色的信任政策，以便與名為 example_cluster 之 EKS 叢集的 example_namespace 命名空間搭配使用。

命令：

```
aws emr-containers update-role-trust-policy \
    --cluster example_cluster \
    --namespace example_namespace \
    --role-name example_iam_role
```

輸出：

```
If the trust policy has already been updated, then the output will be:
Trust policy statement already exists for role example_iam_role. No
changes were made!

If the trust policy has not been updated yet, then the output will be:
Successfully updated trust policy of role example_iam_role.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRoleTrustPolicy。

使用 的 EventBridge 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 EventBridge 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`delete-rule`

以下程式碼範例顯示如何使用 `delete-rule`。

AWS CLI

刪除 CloudWatch Events 規則

此範例會刪除名為 EC2InstanceStateChanges 的規則：

```
aws events delete-rule --name "EC2InstanceStateChanges"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteRule。

`describe-rule`

以下程式碼範例顯示如何使用 `describe-rule`。

AWS CLI

顯示 CloudWatch Events 規則的相關資訊

此範例會顯示名為 DailyLambdaFunction 之規則的相關資訊：

```
aws events describe-rule --name "DailyLambdaFunction"
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeRule。

disable-rule

以下程式碼範例顯示如何使用 disable-rule。

AWS CLI

停用 CloudWatch Events 規則

此範例會停用名為 DailyLambdaFunction 的規則。此規則不會遭到刪除：

```
aws events disable-rule --name "DailyLambdaFunction"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DisableRule。

enable-rule

以下程式碼範例顯示如何使用 enable-rule。

AWS CLI

啟用 CloudWatch Events 規則

此範例會啟用先前已停用之名為 DailyLambdaFunction 的規則：

```
aws events enable-rule --name "DailyLambdaFunction"
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 EnableRule。

list-rule-names-by-target

以下程式碼範例顯示如何使用 list-rule-names-by-target。

AWS CLI

顯示具有指定目標的所有規則

此範例會顯示將名為「MyFunctionName」的 Lambda 函數作為目標的所有規則：

```
aws events list-rule-names-by-target --target-arn "arn:aws:lambda:us-east-1:123456789012:function:MyFunctionName"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListRuleNamesByTarget。

list-rules

以下程式碼範例顯示如何使用 `list-rules`。

AWS CLI

顯示列出所有 CloudWatch Events 規則的清單

此範例會顯示區域中的所有 CloudWatch Events 規則：

```
aws events list-rules
```

顯示列出以特定字串開頭之 CloudWatch Events 規則的清單。

此範例顯示區域中名稱以「Daily」開頭的所有 CloudWatch Events 規則：

```
aws events list-rules --name-prefix "Daily"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListRules。

list-targets-by-rule

以下程式碼範例顯示如何使用 `list-targets-by-rule`。

AWS CLI

顯示 CloudWatch Events 規則的所有目標

此範例會顯示名為 DailyLambdaFunction 之規則的所有目標：

```
aws events list-targets-by-rule --rule  "DailyLambdaFunction"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTargetsByRule。

put-events

以下程式碼範例顯示如何使用 put-events。

AWS CLI

將自訂事件傳送 CloudWatch Events

此範例會將自訂事件傳送至 CloudWatch Events。putevents.json 檔案中包含該事件：

```
aws events put-events --entries file://putevents.json
```

以下為 putevents.json 檔案的內容：

```
[
  {
    "Source": "com.mycompany.myapp",
    "Detail": "{ \"key1\": \"value1\", \"key2\": \"value2\" }",
    "Resources": [
      "resource1",
      "resource2"
    ],
    "DetailType": "myDetailType"
  },
  {
    "Source": "com.mycompany.myapp",
    "Detail": "{ \"key1\": \"value3\", \"key2\": \"value4\" }",
    "Resources": [
      "resource1",
      "resource2"
    ],
    "DetailType": "myDetailType"
   }
]
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutEvents。

put-rule

以下程式碼範例顯示如何使用 put-rule。

AWS CLI

建立 CloudWatch Events 規則

此範例會建立規則，該規則會在每天 UTC 時間上午 9:00 時觸發。如果您使用 put-targets，新增 Lambda 函數作為此規則的目標，則可以在指定的時間每天執行 Lambda 函數：

```
aws events put-rule --name "DailyLambdaFunction" --schedule-expression "cron(0 9 *
 * ? *)"
```

此範例會建立規則，當區域中的任何 EC2 執行個體變更狀態時便會觸發此規則：

```
aws events put-rule --name "EC2InstanceStateChanges" --event-pattern "{\"source\":
[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance State-change Notification\"]}"  --
role-arn "arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

此範例會建立規則，當區域中的任何 EC2 執行個體停止或終止時便會觸發此規則：

```
aws events put-rule --name "EC2InstanceStateChangeStopOrTerminate" --event-
pattern "{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance State-change
 Notification\"],\"detail\":{\"state\":[\"stopped\",\"terminated\"]}}" --role-
arn "arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutRule。

put-targets

以下程式碼範例顯示如何使用 put-targets。

AWS CLI

新增 CloudWatch Events 規則的目標

此範例會新增 Lambda 函數作為規則的目標：

```
aws events put-targets --rule DailyLambdaFunction --targets
 "Id"="1","Arn"="arn:aws:lambda:us-east-1:123456789012:function:MyFunctionName"
```

此範例會將 Amazon Kinesis 串流設定為目標，以便將此規則捕捉到的事件轉送至串流：

```
aws events put-targets --rule EC2InstanceStateChanges --targets
 "Id"="1","Arn"="arn:aws:kinesis:us-east-1:123456789012:stream/
MyStream","RoleArn"="arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

此範例會將兩個 Amazon Kinesis 串流設定為單一規則的目標：

```
aws events put-targets --rule DailyLambdaFunction --targets
 "Id"="Target1","Arn"="arn:aws:kinesis:us-east-1:379642911888:stream/
MyStream1","RoleArn"="arn:aws:iam::379642911888:role/ MyRoleToAccessLambda"
  "Id"="Target2"," Arn"="arn:aws:kinesis:us-east-1:379642911888:stream/
MyStream2","RoleArn"="arn:aws:iam::379642911888:role/MyRoleToAccessLambda"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutTargets。

`remove-targets`

以下程式碼範例顯示如何使用 `remove-targets`。

AWS CLI

移除事件的目標

此範例會將名為 MyStream1 的 Amazon Kinesis 串流移除，使其不再是規則 DailyLambdaFunction 的目標。建立 DailyLambdaFunction 時，會將此串流設為 ID 為 Target1 的目標：

```
aws events remove-targets --rule "DailyLambdaFunction" --ids "Target1"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 RemoveTargets。

`test-event-pattern`

以下程式碼範例顯示如何使用 `test-event-pattern`。

AWS CLI

檢查事件模式是否符合指定的事件

此範例會測試模式 "source：com.mycompany.myapp" 是否符合指定的事件。在此範例中，輸出會是「true」：

```
aws events test-event-pattern --event-pattern "{\"source\":[\"com.mycompany.myapp
\"]}" --event "{\"id\":\"1\",\"source\":\"com.mycompany.myapp\",\"detail-type\":
\"myDetailType\",\"account\":\"123456789012\",\"region\":\"us-east-1\",\"time\":
\"2017-04-11T20:11:04Z\"}"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TestEventPattern。

使用 的 EventBridge 管道範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 EventBridge 管道來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-pipe`

以下程式碼範例顯示如何使用 `create-pipe`。

AWS CLI

建立管道

下列`create-pipe`範例會建立名為 的管道`Demo_Pipe`，並以 SQS 做為來源，並以 CloudWatch Log Group 做為管道的目標。

```
aws pipes create-pipe \
    --name Demo_Pipe \
    --desired-state RUNNING \
    --role-arn arn:aws:iam::123456789012:role/service-role/
Amazon_EventBridge_Pipe_Demo_Pipe_28b3aa4f \
    --source arn:aws:sqs:us-east-1:123456789012:Demo_Queue \
    --target arn:aws:logs:us-east-1:123456789012:log-group:/aws/pipes/Demo_LogGroup
```

輸出：

```
{
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
    "Name": "Demo_Pipe",
    "DesiredState": "RUNNING",
    "CurrentState": "CREATING",
    "CreationTime": "2024-10-08T12:33:59-05:00",
    "LastModifiedTime": "2024-10-08T12:33:59.684839-05:00"
}
```

如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的 Amazon EventBridge 管道概念。 EventBridge

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePipe。

delete-pipe

以下程式碼範例顯示如何使用 delete-pipe。

AWS CLI

刪除現有的管道

下列delete-pipe範例會刪除指定帳戶中名為 Demo_Pipe的管道。

```
aws pipes delete-pipe \
    --name Demo_Pipe
```

輸出：

```
{
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
    "Name": "Demo_Pipe",
    "DesiredState": "STOPPED",
    "CurrentState": "DELETING",
    "CreationTime": "2024-10-08T09:29:10-05:00",
    "LastModifiedTime": "2024-10-08T11:57:22-05:00"
}
```

如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的 Amazon EventBridge 管道概念。 EventBridge

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePipe。

describe-pipe

以下程式碼範例顯示如何使用 describe-pipe。

AWS CLI

擷取管道的相關資訊

下列describe-pipe範例顯示Demo_Pipe指定帳戶中管道的相關資訊。

```
aws pipes describe-pipe \
    --name Demo_Pipe
```

輸出：

```
{
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
    "Name": "Demo_Pipe",
    "DesiredState": "RUNNING",
    "CurrentState": "RUNNING",
    "StateReason": "User initiated",
    "Source": "arn:aws:sqs:us-east-1:123456789012:Demo_Queue",
    "SourceParameters": {
        "SqsQueueParameters": {
            "BatchSize": 1
        }
    },
    "EnrichmentParameters": {},
    "Target": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/pipes/
Demo_LogGroup",
    "TargetParameters": {},
    "RoleArn": "arn:aws:iam::123456789012:role/service-role/
Amazon_EventBridge_Pipe_Demo_Pipe_28b3aa4f",
    "Tags": {},
    "CreationTime": "2024-10-08T09:29:10-05:00",
    "LastModifiedTime": "2024-10-08T10:23:47-05:00",
    "LogConfiguration": {
        "CloudwatchLogsLogDestination": {
            "LogGroupArn": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
vendedlogs/pipes/Demo_Pipe"
        },
```

```
        "Level": "ERROR"
    }
}
```

如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的 Amazon EventBridge 管道概念。 EventBridge

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePipe。

list-pipes

以下程式碼範例顯示如何使用 list-pipes。

AWS CLI

擷取管道清單

下列list-pipes範例顯示指定帳戶中的所有管道。

```
aws pipes list-pipes
```

輸出：

```
{
    "Pipes": [
        {
            "Name": "Demo_Pipe",
            "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
            "DesiredState": "RUNNING",
            "CurrentState": "RUNNING",
            "StateReason": "User initiated",
            "CreationTime": "2024-10-08T09:29:10-05:00",
            "LastModifiedTime": "2024-10-08T10:23:47-05:00",
            "Source": "arn:aws:sqs:us-east-1:123456789012:Demo_Queue",
            "Target": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/pipes/
Demo_LogGroup"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的 Amazon EventBridge 管道概念。 EventBridge

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPipes。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出與現有管道相關聯的標籤

下列`list-tags-for-resource`範例列出與指定帳戶中名為 `Demo_Pipe`的管道相關聯的所有標籤。

```
aws pipes list-tags-for-resource \
    --resource-arn arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe
```

輸出：

```
{
    "tags": {
        "stack": "Production",
        "team": "DevOps"
    }
}
```

如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的 Amazon EventBridge 管道概念。 EventBridge

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`start-pipe`

以下程式碼範例顯示如何使用 `start-pipe`。

AWS CLI

啟動現有的管道

下列`start-pipe`範例會在指定的帳戶中啟動名為 `Demo_Pipe`的管道。

```
aws pipes start-pipe \
```

```
    --name Demo_Pipe
```

輸出：

```
{
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
    "Name": "Demo_Pipe",
    "DesiredState": "RUNNING",
    "CurrentState": "STARTING",
    "CreationTime": "2024-10-08T09:29:10-05:00",
    "LastModifiedTime": "2024-10-08T10:17:24-05:00"
}
```

如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的啟動或停止 Amazon EventBridge 管道。 EventBridge

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartPipe。

stop-pipe

以下程式碼範例顯示如何使用 stop-pipe。

AWS CLI

停止現有的管道

下列stop-pipe範例會停止指定帳戶中名為 Demo_Pipe的管道。

```
aws pipes stop-pipe \
    --name Demo_Pipe
```

輸出：

```
{
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
    "Name": "Demo_Pipe",
    "DesiredState": "STOPPED",
    "CurrentState": "STOPPING",
    "CreationTime": "2024-10-08T09:29:10-05:00",
    "LastModifiedTime": "2024-10-08T09:29:49-05:00"
}
```

如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的啟動或停止 Amazon EventBridge 管道。 EventBridge

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopPipe。

`tag-resource`

以下程式碼範例顯示如何使用 `tag-resource`。

AWS CLI

標記現有的管道

下列`tag-resource`範例會標記名為 的管道`Demo_Pipe`。如果命令成功，則不會傳回任何輸出。

```
aws pipes tag-resource \
    --resource-arn arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe \
    --tags stack=Production
```

如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的 Amazon EventBridge 管道概念。 EventBridge

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

`untag-resource`

以下程式碼範例顯示如何使用 `untag-resource`。

AWS CLI

從現有管道移除標籤

下列`untag-resource`範例`stack`會從名為 的管道中移除具有 金鑰的標籤`Demo_Pipe`。如果命令成功，則不會傳回任何輸出。

```
aws pipes untag-resource \
    --resource-arn arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe \
    --tags stack
```

如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的 Amazon EventBridge 管道概念。 EventBridge

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-pipe

以下程式碼範例顯示如何使用 update-pipe。

AWS CLI

更新現有的管道

下列update-pipe範例Demo_Pipe透過新增 CloudWatch Log 組態參數來更新名為 的管道，確保更新管道的執行角色，使其具有日誌目的地的正確許可。

```
aws pipes update-pipe \
    --name Demo_Pipe \
    --desired-state RUNNING \
    --log-configuration CloudwatchLogsLogDestination={LogGroupArn=arn:aws:logs:us-east-1:123456789012:log-group:/aws/vendedlogs/pipes/Demo_Pipe},Level=TRACE \
    --role-arn arn:aws:iam::123456789012:role/service-role/Amazon_EventBridge_Pipe_Demo_Pipe_28b3aa4f
```

輸出：

```
{
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
    "Name": "Demo_Pipe",
    "DesiredState": "RUNNING",
    "CurrentState": "UPDATING",
    "CreationTime": "2024-10-08T09:29:10-05:00",
    "LastModifiedTime": "2024-10-08T11:35:48-05:00"
}
```

如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的 Amazon EventBridge 管道概念。 EventBridge

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePipe。

使用 的 Firewall Manager 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Firewall Manager 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-admin-account`

以下程式碼範例顯示如何使用 `associate-admin-account`。

AWS CLI

設定 Firewall Manager 管理員帳戶

下列`associate-admin-account`範例會設定 Firewall Manager 的管理員帳戶。

```
aws fms associate-admin-account \
    --admin-account 123456789012
```

此命令不會產生輸出。

如需詳細資訊，請參閱 WAF、Firewall Manager 和 Shield Advanced Developer Guide 中的設定 AWS Firewall Manager 管理員帳戶。 AWS AWS AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateAdminAccount。

`delete-notification-channel`

以下程式碼範例顯示如何使用 `delete-notification-channel`。

AWS CLI

移除 Firewall Manager 日誌的 SNS 主題資訊

下列`delete-notification-channel`範例會移除 SNS 主題資訊。

```
aws fms delete-notification-channel
```

此命令不會產生輸出。

如需詳細資訊，請參閱 WAF、Firewall Manager 和 Shield Advanced Developer Guide 中的設定 Amazon SNS 通知和 Amazon CloudWatch 警示。 AWS AWS AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNotificationChannel。

`delete-policy`

以下程式碼範例顯示如何使用 `delete-policy`。

AWS CLI

刪除 Firewall Manager 政策

下列`delete-policy`範例會移除具有指定 ID 的政策及其所有資源。

```
aws fms delete-policy \
    --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --delete-all-policy-resources
```

此命令不會產生輸出。

如需詳細資訊，請參閱 WAF、Firewall Manager 和 Shield Advanced Developer Guide 中的使用 AWS Firewall Manager 政策。 AWS AWS AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeletePolicy。

`disassociate-admin-account`

以下程式碼範例顯示如何使用 `disassociate-admin-account`。

AWS CLI

移除 Firewall Manager 管理員帳戶

下列`disassociate-admin-account`範例會從 Firewall Manager 移除目前的管理員帳戶關聯。

```
aws fms disassociate-admin-account
```

此命令不會產生輸出。

如需詳細資訊，請參閱 WAF、Firewall Manager 和 Shield Advanced Developer Guide 中的設定 AWS Firewall Manager 管理員帳戶。 AWS AWS AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateAdminAccount。

get-admin-account

以下程式碼範例顯示如何使用 get-admin-account。

AWS CLI

擷取 Firewall Manager 管理員帳戶

下列get-admin-account範例會擷取管理員帳戶。

```
aws fms get-admin-account
```

輸出：

```
{
    "AdminAccount": "123456789012",
    "RoleStatus": "READY"
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide 中的 AWS Firewall Manager 先決條件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAdminAccount。

get-compliance-detail

以下程式碼範例顯示如何使用 get-compliance-detail。

AWS CLI

擷取帳戶的合規資訊

下列get-compliance-detail範例會擷取指定政策和成員帳戶的合規資訊。

```
aws fms get-compliance-detail \
    --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
```

```
    --member-account 123456789012
```

輸出：

```
{
    "PolicyComplianceDetail": {
    "EvaluationLimitExceeded": false,
    "IssueInfoMap": {},
    "MemberAccount": "123456789012",
    "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "PolicyOwner": "123456789012",
    "Violators": []
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide 中的檢視資源是否符合政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetComplianceDetail。

get-notification-channel

以下程式碼範例顯示如何使用 get-notification-channel。

AWS CLI

擷取 Firewall Manager 日誌的 SNS 主題資訊

下列get-notification-channel範例會擷取 SNS 主題資訊。

```
aws fms get-notification-channel
```

輸出：

```
{
    "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:us-west-2-fms",
    "SnsRoleName": "arn:aws:iam::123456789012:role/aws-service-role/
fms.amazonaws.com/AWSServiceRoleForFMS"
}
```

如需詳細資訊，請參閱 WAF、Firewall Manager 和 Shield Advanced Developer Guide 中的設定 Amazon SNS 通知和 Amazon CloudWatch 警示。 AWS AWS AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetNotificationChannel。

get-policy

以下程式碼範例顯示如何使用 get-policy。

AWS CLI

擷取 Firewall Manager 政策

下列get-policy範例會擷取具有指定 ID 的政策。

```
aws fms get-policy \
    --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "Policy": {
        "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "PolicyName": "test",
        "PolicyUpdateToken": "1:p+2RpKR4wPFx7mcrL1UOQQ==",
        "SecurityServicePolicyData": {
            "Type": "SECURITY_GROUPS_COMMON",
            "ManagedServiceData": "{\"type\":\"SECURITY_GROUPS_COMMON\",
\"revertManualSecurityGroupChanges\":true,\"exclusiveResourceSecurityGroupManagement
\":false,\"securityGroups\":[{\"id\":\"sg-045c43ccc9724e63e\"}]}"
        },
        "ResourceType": "AWS::EC2::Instance",
        "ResourceTags": [],
        "ExcludeResourceTags": false,
        "RemediationEnabled": false
    },
    "PolicyArn": "arn:aws:fms:us-west-2:123456789012:policy/d1ac59b8-938e-42b3-
b2e0-7c620422ddc2"
}
```

如需詳細資訊，請參閱 WAF、Firewall Manager 和 Shield Advanced Developer Guide 中的使用 AWS Firewall Manager 政策。 AWS AWS AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetPolicy。

list-compliance-status

以下程式碼範例顯示如何使用 list-compliance-status。

AWS CLI

擷取成員帳戶的政策合規資訊

下列list-compliance-status範例會擷取指定政策的成員帳戶合規資訊。

```
aws fms list-compliance-status \
    --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "PolicyComplianceStatusList": [
        {
            "PolicyOwner": "123456789012",
            "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "PolicyName": "test",
            "MemberAccount": "123456789012",
            "EvaluationResults": [
                {
                    "ComplianceStatus": "COMPLIANT",
                    "ViolatorCount": 0,
                    "EvaluationLimitExceeded": false
                },
                {
                    "ComplianceStatus": "NON_COMPLIANT",
                    "ViolatorCount": 2,
                    "EvaluationLimitExceeded": false
                }
            ],
            "LastUpdated": 1576283774.0,
            "IssueInfoMap": {}
        }
    ]
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide 中的檢視資源是否符合政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListComplianceStatus。

list-member-accounts

以下程式碼範例顯示如何使用 list-member-accounts。

AWS CLI

擷取組織中的成員帳戶

下列list-member-accounts範例列出 Firewall Manager 管理員組織中的所有成員帳戶。

```
aws fms list-member-accounts
```

輸出：

```
{
    "MemberAccounts": [
        "222222222222",
        "333333333333",
        "444444444444"
    ]
}
```

如需詳細資訊，請參閱 WAF、Firewall Manager 和 Shield Advanced Developer Guide 中的 AWS Firewall Manager。 AWS AWS AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMemberAccounts。

list-policies

以下程式碼範例顯示如何使用 list-policies。

AWS CLI

擷取所有 Firewall Manager 政策

下列list-policies範例會擷取帳戶的 政策清單。在此範例中，每個請求的輸出限制為兩個結果。每個呼叫都會傳回 NextToken，可用來做為下次list-policies呼叫中 --starting-token 參數的值，以取得清單的下一組結果。

```
aws fms list-policies \
    --max-items 2
```

輸出：

```
{
    "PolicyList": [
        {
            "PolicyArn": "arn:aws:fms:us-west-2:123456789012:policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "PolicyName": "test",
            "ResourceType": "AWS::EC2::Instance",
            "SecurityServiceType": "SECURITY_GROUPS_COMMON",
            "RemediationEnabled": false
        },
        {
            "PolicyArn": "arn:aws:fms:us-west-2:123456789012:policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "PolicyId": "457c9b21-fc94-406c-ae63-21217395ba72",
            "PolicyName": "test",
            "ResourceType": "AWS::EC2::Instance",
            "SecurityServiceType": "SECURITY_GROUPS_COMMON",
            "RemediationEnabled": false
        }
    ],
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

如需詳細資訊，請參閱 WAF、Firewall Manager 和 Shield Advanced Developer Guide 中的使用 AWS Firewall Manager 政策。 AWS AWS AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListPolicies。

put-notification-channel

以下程式碼範例顯示如何使用 put-notification-channel。

AWS CLI

設定 Firewall Manager 日誌的 SNS 主題資訊

下列put-notification-channel範例會設定 SNS 主題資訊。

```
aws fms put-notification-channel \
    --sns-topic-arn arn:aws:sns:us-west-2:123456789012:us-west-2-fms \
```

```
    --sns-role-name arn:aws:iam::123456789012:role/aws-service-role/
fms.amazonaws.com/AWSServiceRoleForFMS
```

此命令不會產生輸出。

如需詳細資訊，請參閱 WAF、Firewall Manager 和 Shield Advanced Developer Guide 中的設定 Amazon SNS 通知和 Amazon CloudWatch 警示。 AWS AWS AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutNotificationChannel。

put-policy

以下程式碼範例顯示如何使用 put-policy。

AWS CLI

建立 Firewall Manager 政策

下列put-policy範例會建立 Firewall Manager 安全群組政策。

```
aws fms put-policy \
    --cli-input-json file://policy.json
```

policy.json 的內容：

```
{
    "Policy": {
        "PolicyName": "test",
        "SecurityServicePolicyData": {
            "Type": "SECURITY_GROUPS_USAGE_AUDIT",
            "ManagedServiceData": "{\"type\":\"SECURITY_GROUPS_USAGE_AUDIT\",
\"deleteUnusedSecurityGroups\":false,\"coalesceRedundantSecurityGroups\":true}"
        },
        "ResourceType": "AWS::EC2::SecurityGroup",
        "ResourceTags": [],
        "ExcludeResourceTags": false,
        "RemediationEnabled": false
    },
    "TagList": [
        {
            "Key": "foo",
            "Value": "foo"
        }
```

```
    ]
}
```

輸出：

```
{
    "Policy": {
        "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "PolicyName": "test",
        "PolicyUpdateToken": "1:X9QGexP7HASDlsFp+G31Iw==",
        "SecurityServicePolicyData": {
            "Type": "SECURITY_GROUPS_USAGE_AUDIT",
            "ManagedServiceData": "{\"type\":\"SECURITY_GROUPS_USAGE_AUDIT\",
\"deleteUnusedSecurityGroups\":false,\"coalesceRedundantSecurityGroups\":true,
\"optionalDelayForUnusedInMinutes\":null}"
        },
        "ResourceType": "AWS::EC2::SecurityGroup",
        "ResourceTags": [],
        "ExcludeResourceTags": false,
        "RemediationEnabled": false
    },
    "PolicyArn": "arn:aws:fms:us-west-2:123456789012:policy/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
}
```

如需詳細資訊，請參閱 WAF、Firewall Manager 和 Shield Advanced Developer Guide 中的使用 AWS Firewall Manager 政策。 AWS AWS AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutPolicy。

AWS FIS 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS FIS。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-experiment-template`

以下程式碼範例顯示如何使用 `create-experiment-template`。

AWS CLI

建立實驗範本

下列`create-experiment-template`範例會在您的 AWS FIS 帳戶中建立實驗範本。

```
aws fis create-experiment-template \
    --cli-input-json file://myfile.json
```

myfile.json 的內容：

```
{
    "description": "experimentTemplate",
    "stopConditions": [
        {
            "source": "aws:cloudwatch:alarm",
            "value": "arn:aws:cloudwatch:us-west-2:123456789012:alarm:alarmName"
        }
    ],
    "targets": {
        "Instances-Target-1": {
            "resourceType": "aws:ec2:instance",
            "resourceArns": [
                "arn:aws:ec2:us-west-2:123456789012:instance/i-12a3b4c56d78e9012"
            ],
            "selectionMode": "ALL"
        }
    },
    "actions": {
        "reboot": {
            "actionId": "aws:ec2:reboot-instances",
            "description": "reboot",
            "parameters": {},
            "targets": {
                "Instances": "Instances-Target-1"
            }
```

```
        }
    },
    "roleArn": "arn:aws:iam::123456789012:role/myRole"
}
```

輸出：

```
{
    "experimentTemplate": {
        "id": "ABCDE1fgHIJkLmNop",
        "description": "experimentTemplate",
        "targets": {
            "Instances-Target-1": {
                "resourceType": "aws:ec2:instance",
                "resourceArns": [
                    "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
                ],
                "selectionMode": "ALL"
            }
        },
        "actions": {
            "reboot": {
                "actionId": "aws:ec2:reboot-instances",
                "description": "reboot",
                "parameters": {},
                "targets": {
                    "Instances": "Instances-Target-1"
                }
            }
        },
        "stopConditions": [
            {
                "source": "aws:cloudwatch:alarm",
                "value": "arn:aws:cloudwatch:us-west-2:123456789012:alarm:alarmName"
            }
        ],
        "creationTime": 1616434850.659,
        "lastUpdateTime": 1616434850.659,
        "roleArn": "arn:aws:iam::123456789012:role/myRole",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 AWS Fault Injection Simulator 使用者指南中的建立實驗範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateExperimentTemplate。

delete-experiment-template

以下程式碼範例顯示如何使用 delete-experiment-template。

AWS CLI

刪除實驗範本

下列delete-experiment-template範例會刪除指定的實驗範本。

```
aws fis delete-experiment-template \
    --id ABCDE1fgHIJkLmNop
```

輸出：

```
{
    "experimentTemplate": {
        "id": "ABCDE1fgHIJkLmNop",
        "description": "myExperimentTemplate",
        "targets": {
            "Instances-Target-1": {
                "resourceType": "aws:ec2:instance",
                "resourceArns": [
                    "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
                ],
                "selectionMode": "ALL"
            }
        },
        "actions": {
            "testaction": {
                "actionId": "aws:ec2:stop-instances",
                "parameters": {},
                "targets": {
                    "Instances": "Instances-Target-1"
                }
            }
        },
        "stopConditions": [
```

```
            {
                "source": "none"
            }
        ],
        "creationTime": 1616017191.124,
        "lastUpdateTime": 1616017859.607,
        "roleArn": "arn:aws:iam::123456789012:role/FISRole"
    }
}
```

如需詳細資訊，請參閱 AWS Fault Injection Simulator 使用者指南中的刪除實驗範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteExperimentTemplate。

get-action

以下程式碼範例顯示如何使用 get-action。

AWS CLI

取得動作詳細資訊

下列get-action範例會取得指定動作的詳細資訊。

```
aws fis get-action \
    --id aws:ec2:stop-instances
```

輸出：

```
{
    "action": {
        "id": "aws:ec2:stop-instances",
        "description": "Stop the specified EC2 instances.",
        "parameters": {
            "startInstancesAfterDuration": {
                "description": "The time to wait before restarting the instances
 (ISO 8601 duration).",
                "required": false
            }
        },
        "targets": {
            "Instances": {
```

```
                "resourceType": "aws:ec2:instance"
            }
        },
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 AWS Fault Injection Simulator 使用者指南中的動作。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAction。

get-experiment-template

以下程式碼範例顯示如何使用 get-experiment-template。

AWS CLI

取得實驗範本詳細資訊

下列get-experiment-template範例會取得指定實驗範本的詳細資訊。

```
aws fis get-experiment-template \
    --id ABCDE1fgHIJkLmNop
```

輸出：

```
{
    "experimentTemplate": {
        "id": "ABCDE1fgHIJkLmNop",
        "description": "myExperimentTemplate",
        "targets": {
            "Instances-Target-1": {
                "resourceType": "aws:ec2:instance",
                "resourceArns": [
                    "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
                ],
                "selectionMode": "ALL"
            }
        },
        "actions": {
            "testaction": {
```

```
                "actionId": "aws:ec2:stop-instances",
                "parameters": {},
                "targets": {
                    "Instances": "Instances-Target-1"
                }
            }
        },
        "stopConditions": [
            {
                "source": "none"
            }
        ],
        "creationTime": 1616017191.124,
        "lastUpdateTime": 1616017331.51,
        "roleArn": "arn:aws:iam::123456789012:role/FISRole",
        "tags": {
        "key: "value"
        }
    }
}
```

如需詳細資訊，請參閱《AWS 故障注入模擬器使用者指南》中的實驗範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetExperimentTemplate。

get-experiment

以下程式碼範例顯示如何使用 get-experiment。

AWS CLI

若要取得實驗詳細資訊

下列get-experiment範例會取得指定實驗的詳細資訊。

```
aws fis get-experiment \
    --id ABC12DeFGhI3jKLMNOP
```

輸出：

```
{
    "experiment": {
        "id": "ABC12DeFGhI3jKLMNOP",
```

```
        "experimentTemplateId": "ABCDE1fgHIJkLmNop",
        "roleArn": "arn:aws:iam::123456789012:role/myRole",
        "state": {
            "status": "completed",
            "reason": "Experiment completed."
        },
        "targets": {
            "Instances-Target-1": {
                "resourceType": "aws:ec2:instance",
                "resourceArns": [
                    "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
                ],
                "selectionMode": "ALL"
            }
        },
        "actions": {
            "reboot": {
                "actionId": "aws:ec2:reboot-instances",
                "parameters": {},
                "targets": {
                    "Instances": "Instances-Target-1"
                },
                "state": {
                    "status": "completed",
                    "reason": "Action was completed."
                }
            }
        },
        "stopConditions": [
            {
                "source": "none"
            }
        ],
        "creationTime": 1616432509.662,
        "startTime": 1616432509.962,
        "endTime": 1616432522.307,
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 AWS Fault Injection Simulator 使用者指南中的 AWS FIS 實驗。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetExperiment。

list-actions

以下程式碼範例顯示如何使用 list-actions。

AWS CLI

列出動作

下列list-actions範例列出可用的動作。

```
aws fis list-actions
```

輸出：

```
{
    "actions": [
        {
            "id": "aws:ec2:reboot-instances",
            "description": "Reboot the specified EC2 instances.",
            "targets": {
                "Instances": {
                    "resourceType": "aws:ec2:instance"
                }
            },
            "tags": {}
        },
        {
            "id": "aws:ec2:stop-instances",
            "description": "Stop the specified EC2 instances.",
            "targets": {
                "Instances": {
                    "resourceType": "aws:ec2:instance"
                }
            },
            "tags": {}
        },
        {
            "id": "aws:ec2:terminate-instances",
            "description": "Terminate the specified EC2 instances.",
            "targets": {
                "Instances": {
                    "resourceType": "aws:ec2:instance"
                }
            },
```

```
            "tags": {}
        },
        {
            "id": "aws:ecs:drain-container-instances",
            "description": "Drain percentage of underlying EC2 instances on an ECS
cluster.",
            "targets": {
                "Clusters": {
                    "resourceType": "aws:ecs:cluster"
                }
            },
            "tags": {}
        },
        {
            "id": "aws:eks:terminate-nodegroup-instances",
            "description": "Terminates a percentage of the underlying EC2 instances
in an EKS cluster.",
            "targets": {
                "Nodegroups": {
                    "resourceType": "aws:eks:nodegroup"
                }
            },
            "tags": {}
        },
        {
            "id": "aws:fis:inject-api-internal-error",
            "description": "Cause an AWS service to return internal error responses
for specific callers and operations.",
            "targets": {
                "Roles": {
                    "resourceType": "aws:iam:role"
                }
            },
            "tags": {}
        },
        {
            "id": "aws:fis:inject-api-throttle-error",
            "description": "Cause an AWS service to return throttled responses for
specific callers and operations.",
            "targets": {
                "Roles": {
                    "resourceType": "aws:iam:role"
                }
            },
```

```
            "tags": {}
        },
        {
        "id": "aws:fis:inject-api-unavailable-error",
            "description": "Cause an AWS service to return unavailable error
responses for specific callers and operations.",
            "targets": {
                "Roles": {
                    "resourceType": "aws:iam:role"
                }
            },
            "tags": {}
        },
        {
            "id": "aws:fis:wait",
            "description": "Wait for the specified duration. Stop condition
monitoring will continue during this time.",
            "tags": {}
        },
        {
            "id": "aws:rds:failover-db-cluster",
            "description": "Failover a DB Cluster to one of the replicas.",
            "targets": {
                "Clusters": {
                    "resourceType": "aws:rds:cluster"
                }
            },
            "tags": {}
        },
        {
            "id": "aws:rds:reboot-db-instances",
            "description": "Reboot the specified DB instances.",
            "targets": {
                "DBInstances": {
                    "resourceType": "aws:rds:db"
                }
            },
            "tags": {}
        },
        {
            "id": "aws:ssm:send-command",
            "description": "Run the specified SSM document.",
            "targets": {
                "Instances": {
```

```
                    "resourceType": "aws:ec2:instance"
                }
            },
            "tags": {}
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Fault Injection Simulator 使用者指南中的動作。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListActions。

`list-experiment-templates`

以下程式碼範例顯示如何使用 `list-experiment-templates`。

AWS CLI

列出實驗範本

下列`list-experiment-templates`範例列出您 AWS 帳戶中的實驗範本。

```
aws fis list-experiment-templates
```

輸出：

```
{
    "experimentTemplates": [
        {
            "id": "ABCDE1fgHIJkLmNop",
            "description": "myExperimentTemplate",
            "creationTime": 1616017191.124,
            "lastUpdateTime": 1616017191.124,
            "tags": {
                "key": "value"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《AWS 故障注入模擬器使用者指南》中的實驗範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListExperimentTemplates。

list-experiments

以下程式碼範例顯示如何使用 list-experiments。

AWS CLI

列出實驗

下列list-experiments範例列出您 AWS 帳戶中的實驗。

```
aws fis list-experiments
```

輸出：

```
{
    "experiments": [
        {
            "id": "ABCdeF1GHiJkLM23NO",
            "experimentTemplateId": "ABCDE1fgHIJkLmNop",
            "state": {
                "status": "running",
                "reason": "Experiment is running."
            },
            "creationTime": 1616017341.197,
            "tags": {
            "key": "value"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《 AWS Fault Injection Simulator 使用者指南》中的實驗。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListExperiments。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出資源的標籤

下列list-tags-for-resource範例列出指定資源的標籤。

```
aws fis list-tags-for-resource \
    --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP
```

輸出：

```
{
    "tags": {
        "key1": "value1",
        "key2": "value2"
    }
}
```

如需詳細資訊，請參閱 AWS Fault Injection Simulator 使用者指南中的標記您的 AWS FIS 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

start-experiment

以下程式碼範例顯示如何使用 start-experiment。

AWS CLI

開始實驗

下列start-experiment範例會啟動指定的實驗。

```
aws fis start-experiment \
    --experiment-template-id ABCDE1fgHIJkLmNop
```

輸出：

```
{
    "experiment": {
        "id": "ABC12DeFGhI3jKLMNOP",
        "experimentTemplateId": "ABCDE1fgHIJkLmNop",
        "roleArn": "arn:aws:iam::123456789012:role/myRole",
        "state": {
            "status": "initiating",
```

```
            "reason": "Experiment is initiating."
        },
        "targets": {
            "Instances-Target-1": {
                "resourceType": "aws:ec2:instance",
                "resourceArns": [
                    "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
                ],
                "selectionMode": "ALL"
            }
        },
        "actions": {
            "reboot": {
                "actionId": "aws:ec2:reboot-instances",
                "parameters": {},
                "targets": {
                    "Instances": "Instances-Target-1"
                },
                "state": {
                    "status": "pending",
                    "reason": "Initial state"
                }
            }
        },
        "stopConditions": [
            {
                "source": "none"
            }
        ],
        "creationTime": 1616432464.025,
        "startTime": 1616432464.374,
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 AWS Fault Injection Simulator 使用者指南中的 AWS FIS 實驗。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartExperiment。

stop-experiment

以下程式碼範例顯示如何使用 stop-experiment。

AWS CLI

停止實驗

下列stop-experiment範例會停止執行指定的實驗。

```
aws fis stop-experiment \
    --id ABC12DeFGhI3jKLMNOP
```

輸出：

```
{
    "experiment": {
        "id": "ABC12DeFGhI3jKLMNOP",
        "experimentTemplateId": "ABCDE1fgHIJkLmNop",
        "roleArn": "arn:aws:iam::123456789012:role/myRole",
        "state": {
            "status": "stopping",
            "reason": "Stopping Experiment."
        },
        "targets": {
            "Instances-Target-1": {
                "resourceType": "aws:ec2:instance",
                "resourceArns": [
                    "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
                ],
                "selectionMode": "ALL"
            }
        },
        "actions": {
            "reboot": {
                "actionId": "aws:ec2:reboot-instances",
                "parameters": {},
                "targets": {
                    "Instances": "Instances-Target-1"
                },
                "startAfter": [
                    "wait"
                ],
                "state": {
                    "status": "pending",
                    "reason": "Initial state."
```

```
                }
            },
            "wait": {
                "actionId": "aws:fis:wait",
                "parameters": {
                    "duration": "PT5M"
                },
                "state": {
                    "status": "running",
                    "reason": ""
                }
            }
        },
        "stopConditions": [
            {
                "source": "none"
            }
        ],
        "creationTime": 1616432680.927,
        "startTime": 1616432681.177,
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 AWS Fault Injection Simulator 使用者指南中的 AWS FIS 實驗。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopExperiment。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

標記資源

下列tag-resource範例會標記指定的資源。

```
aws fis tag-resource \
    --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP
 \
    --tags key1=value1,key2=value2
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Fault Injection Simulator 使用者指南中的標記您的 AWS FIS 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

取消標記資源

下列untag-resource範例會從指定的資源移除標籤。

```
aws fis untag-resource \
    --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Fault Injection Simulator 使用者指南中的標記您的 AWS FIS 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-experiment-template

以下程式碼範例顯示如何使用 update-experiment-template。

AWS CLI

更新實驗範本

下列update-experiment-template範例會更新指定實驗範本的描述。

```
aws fis update-experiment-template \
    --id ABCDE1fgHIJkLmNop \
    ---description myExperimentTemplate
```

輸出：

```
{
    "experimentTemplate": {
```

```
        "id": "ABCDE1fgHIJkLmNop",
        "description": "myExperimentTemplate",
        "targets": {
            "Instances-Target-1": {
                "resourceType": "aws:ec2:instance",
                "resourceArns": [
                    "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
                ],
                "selectionMode": "ALL"
            }
        },
        "actions": {
            "testaction": {
                "actionId": "aws:ec2:stop-instances",
                "parameters": {},
                "targets": {
                    "Instances": "Instances-Target-1"
                }
            }
        },
        "stopConditions": [
            {
                "source": "none"
            }
        ],
        "creationTime": 1616017191.124,
        "lastUpdateTime": 1616017859.607,
        "roleArn": "arn:aws:iam::123456789012:role/FISRole",
        "tags": {
            "key": "value"
        }
    }
}
```

如需詳細資訊，請參閱 AWS Fault Injection Simulator 使用者指南中的更新實驗範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateExperimentTemplate。

使用 的 Amazon GameLift 伺服器範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon GameLift Servers 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-build`

以下程式碼範例顯示如何使用 `create-build`。

AWS CLI

Example1：從 S3 儲存貯體中的檔案建立遊戲組建

下列`create-build`範例會建立自訂遊戲建置資源。它使用存放在您控制 AWS 之帳戶中 S3 位置的壓縮檔案。此範例假設您已建立 IAM 角色，授予 Amazon GameLift 存取 S3 位置的許可。由於請求未指定作業系統，新的建置資源預設為 WINDOWS_2012。

```
aws gamelift create-build \
    --storage-location file://storage-loc.json \
    --name MegaFrogRaceServer.NA \
    --build-version 12345.678
```

`storage-loc.json` 的內容：

```
{
    "Bucket":"MegaFrogRaceServer_NA_build_files"
    "Key":"MegaFrogRaceServer_build_123.zip"
    "RoleArn":"arn:aws:iam::123456789012:role/gamelift"
}
```

輸出：

```
{
    "Build": {
```

```
        "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "CreationTime": 1496708916.18,
        "Name": "MegaFrogRaceServer.NA",
        "OperatingSystem": "WINDOWS_2012",
        "SizeOnDisk": 479303,
        "Status": "INITIALIZED",
        "Version": "12345.678"
    },
    "StorageLocation": {
        "Bucket": "MegaFrogRaceServer_NA_build_files",
        "Key": "MegaFrogRaceServer_build_123.zip"
    }
}
```

Example2：建立遊戲建置資源，以手動將檔案上傳至 GameLift

下列create-build範例會建立新的建置資源。它也會取得儲存位置和臨時登入資料，可讓您手動將遊戲組建上傳到 Amazon S3 中的 GameLift 位置。成功上傳組建後，GameLift 服務會驗證組建並更新新組建的狀態。

```
aws gamelift create-build \
    --name MegaFrogRaceServer.NA \
    --build-version 12345.678 \
    --operating-system AMAZON_LINUX
```

輸出：

```
{
    "Build": {
        "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "CreationTime": 1496708916.18,
        "Name": "MegaFrogRaceServer.NA",
        "OperatingSystem": "AMAZON_LINUX",
        "SizeOnDisk": 0,
        "Status": "INITIALIZED",
        "Version": "12345.678"
    },
    "StorageLocation": {
```

```
        "Bucket": "gamelift-builds-us-west-2",
        "Key": "123456789012/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    },
    "UploadCredentials": {
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "SessionToken": "AgoGb3JpZ2luENz...EXAMPLETOKEN=="
    }
}
```

如需詳細資訊，請參閱《Amazon GameLift 開發人員指南》中的將自訂伺服器建置上傳至 GameLift。 GameLift

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateBuild。

create-fleet

以下程式碼範例顯示如何使用 create-fleet。

AWS CLI

範例 1：建立基本 Linux 機群

下列create-fleet範例會建立最少設定的隨需 Linux 執行個體機群，以託管自訂伺服器建置。您可以使用 完成組態update-fleet。

```
aws gamelift create-fleet \
    --name MegaFrogRaceServer.NA.v2 \
    --description 'Hosts for v2 North America' \
    --build-id build-1111aaaa-22bb-33cc-44dd-5555eeee66ff \
    --certificate-configuration 'CertificateType=GENERATED' \
    --ec2-instance-type c4.large \
    --fleet-type ON_DEMAND \
    --runtime-configuration 'ServerProcesses=[{LaunchPath=/local/game/release-na/
MegaFrogRace_Server.exe,ConcurrentExecutions=1}]'
```

輸出：

```
{
    "FleetAttributes": {
        "BuildId": "build-1111aaaa-22bb-33cc-44dd-5555eeee66ff",
        "CertificateConfiguration": {
```

```
            "CertificateType": "GENERATED"
        },
        "CreationTime": 1496365885.44,
        "Description": "Hosts for v2 North America",
        "FleetArn": "arn:aws:gamelift:us-west-2:444455556666:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "FleetType": "ON_DEMAND",
        "InstanceType": "c4.large",
        "MetricGroups": ["default"],
        "Name": "MegaFrogRace.NA.v2",
        "NewGameSessionProtectionPolicy": "NoProtection",
        "OperatingSystem": "AMAZON_LINUX",
        "ServerLaunchPath": "/local/game/release-na/MegaFrogRace_Server.exe",
        "Status": "NEW"
    }
}
```

範例 2：建立基本 Windows 機群

下列create-fleet範例會建立最少設定的 Spot Windows 執行個體機群，以託管自訂伺服器建置。您可以使用 完成組態update-fleet。

```
aws gamelift create-fleet \
    --name MegaFrogRace.NA.v2 \
    --description 'Hosts for v2 North America' \
    --build-id build-2222aaaa-33bb-44cc-55dd-6666eeee77ff  \
    --certificate-configuration 'CertificateType=GENERATED' \
    --ec2-instance-type c4.large \
    --fleet-type SPOT \
    --runtime-configuration 'ServerProcesses=[{LaunchPath=C:\game
\Bin64.Release.Dedicated\MegaFrogRace_Server.exe,ConcurrentExecutions=1}]'
```

輸出：

```
{
    "FleetAttributes": {
        "BuildId": "build-2222aaaa-33bb-44cc-55dd-6666eeee77ff",
        "CertificateConfiguration": {
            "CertificateType": "GENERATED"
        },
        "CreationTime": 1496365885.44,
```

```
        "Description": "Hosts for v2 North America",
        "FleetArn": "arn:aws:gamelift:us-west-2:444455556666:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "FleetType": "SPOT",
        "InstanceType": "c4.large",
        "MetricGroups": ["default"],
        "Name": "MegaFrogRace.NA.v2",
        "NewGameSessionProtectionPolicy": "NoProtection",
        "OperatingSystem": "WINDOWS_2012",
        "ServerLaunchPath": "C:\game\Bin64.Release.Dedicated
\MegaFrogRace_Server.exe",
        "Status": "NEW"
    }
}
```

範例 3：建立完全設定的機群

下列`create-fleet`範例會建立自訂伺服器建置的 Spot Windows 執行個體機群，並提供最常用的組態設定。

```
aws gamelift create-fleet \
    --name MegaFrogRace.NA.v2 \
    --description 'Hosts for v2 North America' \
    --build-id build-2222aaaa-33bb-44cc-55dd-6666eeee77ff \
    --certificate-configuration 'CertificateType=GENERATED' \
    --ec2-instance-type c4.large \
    --ec2-inbound-permissions
 'FromPort=33435,ToPort=33435,IpRange=10.24.34.0/23,Protocol=UDP' \
    --fleet-type SPOT \
    --new-game-session-protection-policy FullProtection \
    --runtime-configuration file://runtime-config.json \
    --metric-groups default \
    --instance-role-arn 'arn:aws:iam::444455556666:role/GameLiftS3Access'
```

`runtime-config.json` 的內容：

```
GameSessionActivationTimeoutSeconds=300,
 MaxConcurrentGameSessionActivations=2,
 ServerProcesses=[
   {LaunchPath=C:\game\Bin64.Release.Dedicated\MegaFrogRace_Server.exe,Parameters=-
debug,ConcurrentExecutions=1},
```

```
   {LaunchPath=C:\game\Bin64.Release.Dedicated
\MegaFrogRace_Server.exe,ConcurrentExecutions=1}]
```

輸出：

```
{
    "FleetAttributes": {
        "InstanceRoleArn": "arn:aws:iam::444455556666:role/GameLiftS3Access",
        "Status": "NEW",
        "InstanceType": "c4.large",
        "FleetArn": "arn:aws:gamelift:us-west-2:444455556666:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "Description": "Hosts for v2 North America",
        "FleetType": "SPOT",
        "OperatingSystem": "WINDOWS_2012",
        "Name": "MegaFrogRace.NA.v2",
        "CreationTime": 1569309011.11,
        "MetricGroups": [
            "default"
        ],
        "BuildId": "build-2222aaaa-33bb-44cc-55dd-6666eeee77ff",
        "ServerLaunchParameters": "abc",
        "ServerLaunchPath": "C:\\game\\Bin64.Release.Dedicated\
\MegaFrogRace_Server.exe",
        "NewGameSessionProtectionPolicy": "FullProtection",
        "CertificateConfiguration": {
            "CertificateType": "GENERATED"
        }
    }
}
```

範例 4：建立即時伺服器機群

下列`create-fleet`範例會使用已上傳至 Amazon GameLift 的 Realtime 組態指令碼建立 Spot 執行個體機群。所有即時伺服器都會部署到 Linux 機器。基於此範例的目的，假設上傳的 Realtime 指令碼包含多個指令碼檔案，其`Init()`函數位於指令碼檔案中名為 `MainScript.js`。如圖所示，此檔案在執行時間組態中識別為啟動指令碼。

```
aws gamelift create-fleet \
    --name MegaFrogRace.NA.realtime \
    --description 'Mega Frog Race Realtime fleet' \
```

```
    --script-id script-1111aaaa-22bb-33cc-44dd-5555eeee66ff \
    --ec2-instance-type c4.large \
    --fleet-type SPOT \
    --certificate-configuration 'CertificateType=GENERATED' --runtime-configuration
 'ServerProcesses=[{LaunchPath=/local/game/MainScript.js,Parameters=+map
 Winter444,ConcurrentExecutions=5}]'
```

輸出：

```
{
    "FleetAttributes": {
        "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "Status": "NEW",
        "CreationTime": 1569310745.212,
        "InstanceType": "c4.large",
        "NewGameSessionProtectionPolicy": "NoProtection",
        "CertificateConfiguration": {
            "CertificateType": "GENERATED"
        },
        "Name": "MegaFrogRace.NA.realtime",
        "ScriptId": "script-1111aaaa-22bb-33cc-44dd-5555eeee66ff",
        "FleetArn": "arn:aws:gamelift:us-west-2:444455556666:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "FleetType": "SPOT",
        "MetricGroups": [
            "default"
        ],
        "Description": "Mega Frog Race Realtime fleet",
        "OperatingSystem": "AMAZON_LINUX"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFleet。

create-game-session-queue

以下程式碼範例顯示如何使用 create-game-session-queue。

AWS CLI

Example1：設定排序的遊戲工作階段佇列

下列create-game-session-queue範例會建立新的遊戲工作階段佇列，其中目的地位於兩個區域。它也會設定佇列，讓遊戲工作階段請求在等待 10 分鐘後逾時。由於未定義延遲政策，GameLift 會嘗試放置列出第一個目的地的所有遊戲工作階段。

```
aws gamelift create-game-session-queue \
    --name MegaFrogRaceServer-NA \
    --destinations file://destinations.json \
    --timeout-in-seconds 600
```

destinations.json 的內容：

```
{
    "Destinations": [
        {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" },
        {"DestinationArn": "arn:aws:gamelift:us-west-1::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222" }
    ]
}
```

輸出：

```
{
    "GameSessionQueues": [
        {
            "Name": "MegaFrogRaceServer-NA",
            "GameSessionQueueArn": "arn:aws:gamelift:us-
west-2:123456789012:gamesessionqueue/MegaFrogRaceServer-NA",
            "TimeoutInSeconds": 600,
            "Destinations": [
                {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"},
                {"DestinationArn": "arn:aws:gamelift:us-west-1::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"}
            ]
        }
    ]
}
```

Example2：使用玩家延遲政策設定遊戲工作階段佇列

下列create-game-session-queue範例會建立具有兩個玩家延遲政策的新遊戲工作階段佇列。第一個政策會設定 100 毫秒的延遲上限，在遊戲工作階段放置嘗試的第一分鐘強制執行。第二個政策會將延遲上限提高到 200 毫秒，直到置放請求在 3 分鐘逾時為止。

```
aws gamelift create-game-session-queue \
    --name MegaFrogRaceServer-NA \
    --destinations file://destinations.json \
    --player-latency-policies file://latency-policies.json \
    --timeout-in-seconds 180
```

destinations.json 的內容：

```
{
    "Destinations": [
        { "DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" },
        { "DestinationArn": "arn:aws:gamelift:us-east-1::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222" }
    ]
}
```

latency-policies.json 的內容：

```
{
    "PlayerLatencyPolicies": [
        {"MaximumIndividualPlayerLatencyMilliseconds": 200},
        {"MaximumIndividualPlayerLatencyMilliseconds": 100, "PolicyDurationSeconds":
 60}
    ]
}
```

輸出：

```
{
    "GameSessionQueue": {
        "Name": "MegaFrogRaceServer-NA",
        "GameSessionQueueArn": "arn:aws:gamelift:us-
west-2:111122223333:gamesessionqueue/MegaFrogRaceServer-NA",
        "TimeoutInSeconds": 600,
        "PlayerLatencyPolicies": [
            {
```

```
                "MaximumIndividualPlayerLatencyMilliseconds": 100,
                "PolicyDurationSeconds": 60
            },
            {
                "MaximumIndividualPlayerLatencyMilliseconds": 200
            }
        ]
        "Destinations": [
            {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"},
            {"DestinationArn": "arn:aws:gamelift:us-east-1::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"}
        ],
    }
}
```

如需詳細資訊，請參閱《Amazon GameLift 開發人員指南》中的建立佇列。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateGameSessionQueue。

delete-build

以下程式碼範例顯示如何使用 delete-build。

AWS CLI

刪除自訂遊戲組建

下列delete-build範例會從 Amazon GameLift 帳戶移除組建。刪除組建之後，您就無法使用它來建立新的機群。此操作無法復原。

```
aws gamelift delete-build \
   --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBuild。

delete-fleet

以下程式碼範例顯示如何使用 delete-fleet。

AWS CLI

刪除不再使用的機群

下列delete-fleet範例會移除已縮減至零個執行個體的機群。如果機群容量大於零，請求會失敗並出現 HTTP 400 錯誤。

```
aws gamelift delete-fleet \
   --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon GameLift 開發人員指南》中的管理 GameLift 機群。 GameLift

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFleet。

delete-game-session-queue

以下程式碼範例顯示如何使用 delete-game-session-queue。

AWS CLI

刪除遊戲工作階段佇列

下列delete-game-session-queue範例會刪除指定的遊戲工作階段佇列。

```
aws gamelift delete-game-session-queue \
    --name MegaFrogRace-NA
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteGameSessionQueue。

describe-build

以下程式碼範例顯示如何使用 describe-build。

AWS CLI

取得自訂遊戲組建的相關資訊

下列describe-build範例會擷取遊戲伺服器建置資源的屬性。

```
aws gamelift describe-build \
    --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "Build": {
        "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "CreationTime": 1496708916.18,
        "Name": "My_Game_Server_Build_One",
        "OperatingSystem": "AMAZON_LINUX",
        "SizeOnDisk": 1304924,
        "Status": "READY",
        "Version": "12345.678"
    }
}
```

如需詳細資訊，請參閱《Amazon GameLift 開發人員指南》中的將自訂伺服器建置上傳至 GameLift。 GameLift

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeBuild。

describe-ec2-instance-limits

以下程式碼範例顯示如何使用 describe-ec2-instance-limits。

AWS CLI

擷取 EC2 執行個體類型的服務限制

下列describe-ec2-instance-limits範例顯示目前區域中指定 EC2 執行個體類型使用的最大允許執行個體和目前執行個體。結果指出只使用 5 個允許的 20 個執行個體。

```
aws gamelift describe-ec2-instance-limits \
    --ec2-instance-type m5.large
```

輸出：

```
{
    "EC2InstanceLimits": [
```

```
        {
            "EC2InstanceType": ""m5.large",
            "CurrentInstances": 5,
            "InstanceLimit": 20
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon GameLift 開發人員指南》中的選擇運算資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEc2InstanceLimits。

describe-fleet-attributes

以下程式碼範例顯示如何使用 describe-fleet-attributes。

AWS CLI

Example1：檢視機群清單的屬性

下列describe-fleet-attributes範例會擷取兩個指定機群的機群屬性。如所示，請求的機群使用相同的組建進行部署，一個用於隨需執行個體，另一個用於 Spot 執行個體，具有一些細微的組態差異。

```
aws gamelift describe-fleet-attributes \
    --fleet-ids arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

輸出：

```
{
    "FleetAttributes": [
        {
            "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "FleetArn": "arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "FleetType": "ON_DEMAND",
            "InstanceType": "c4.large",
            "Description": "On-demand hosts for v2 North America",
            "Name": "MegaFrogRaceServer.NA.v2-od",
            "CreationTime": 1568836191.995,
            "Status": "ACTIVE",
            "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
```

```
            "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE33333",
            "ServerLaunchPath": "C:\\game\\MegaFrogRace_Server.exe",
            "ServerLaunchParameters": "+gamelift_start_server",
            "NewGameSessionProtectionPolicy": "NoProtection",
            "OperatingSystem": "WINDOWS_2012",
            "MetricGroups": [
                "default"
            ],
            "CertificateConfiguration": {
                "CertificateType": "DISABLED"
            }
        },
        {
            "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "FleetArn": "arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222",
            "FleetType": "SPOT",
            "InstanceType": "c4.large",
            "Description": "On-demand hosts for v2 North America",
            "Name": "MegaFrogRaceServer.NA.v2-spot",
            "CreationTime": 1568838275.379,
            "Status": "ACTIVATING",
            "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE33333",
            "ServerLaunchPath": "C:\\game\\MegaFrogRace_Server.exe",
            "NewGameSessionProtectionPolicy": "NoProtection",
            "OperatingSystem": "WINDOWS_2012",
                "MetricGroups": [
                "default"
            ],
            "CertificateConfiguration": {
                "CertificateType": "GENERATED"
            }
        }
    ]
}
```

Example2：請求所有機群的屬性

下列會`describe-fleet-attributes`傳回任何狀態之所有機群的機群屬性。此範例說明使用分頁參數一次傳回一個機群。

```
aws gamelift describe-fleet-attributes \
    --limit 1
```

輸出：

```
{
    "FleetAttributes": [
        {
            "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "FleetArn": "arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222",
            "FleetType": "SPOT",
            "InstanceType": "c4.large",
            "Description": "On-demand hosts for v2 North America",
            "Name": "MegaFrogRaceServer.NA.v2-spot",
            "CreationTime": 1568838275.379,
            "Status": "ACTIVATING",
            "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE33333",
            "ServerLaunchPath": "C:\\game\\MegaFrogRace_Server.exe",
            "NewGameSessionProtectionPolicy": "NoProtection",
            "OperatingSystem": "WINDOWS_2012",
            "MetricGroups": [
                "default"
            ],
            "CertificateConfiguration": {
                "CertificateType": "GENERATED"
            }
        }
    ],
    "NextToken":
 "eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}
```

輸出包含您可以再次呼叫命令時使用NextToken的值。將 值傳遞至 --next-token 參數，以指定在何處取得輸出。下列命令會傳回輸出中的第二個結果。

```
aws gamelift describe-fleet-attributes \
    --limit 1 \
    --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZT
```

重複此動作，直到回應不包含NextToken值為止。

如需詳細資訊，請參閱《Amazon GameLift 開發人員指南》中的設定 GameLift 機群。 GameLift

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFleetAttributes。

describe-fleet-capacity

以下程式碼範例顯示如何使用 describe-fleet-capacity。

AWS CLI

檢視機群清單的容量狀態

下列describe-fleet-capacity範例會擷取兩個指定機群的目前容量。

```
aws gamelift describe-fleet-capacity \
    --fleet-ids arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111 fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

輸出：

```
{
    "FleetCapacity": [
        {
            "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "InstanceType": "c5.large",
            "InstanceCounts": {
                "DESIRED": 10,
                "MINIMUM": 1,
                "MAXIMUM": 20,
                "PENDING": 0,
                "ACTIVE": 10,
                "IDLE": 3,
                "TERMINATING": 0
            }
        },
        {
            "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "InstanceType": "c5.large",
            "InstanceCounts": {
                "DESIRED": 13,
                "MINIMUM": 1,
```

```
                "MAXIMUM": 20,
                "PENDING": 0,
                "ACTIVE": 15,
                "IDLE": 2,
                "TERMINATING": 2
            }
        }

    ]
}
```

如需詳細資訊，請參閱《Amazon GameLift 開發人員指南》中的機群 GameLift 指標。 GameLift

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFleetCapacity。

describe-fleet-events

以下程式碼範例顯示如何使用 describe-fleet-events。

AWS CLI

請求指定時間範圍的事件

下列describe-fleet-events範例顯示指定時間範圍期間發生的所有機群相關事件的詳細資訊。

```
aws gamelift describe-fleet-events \
    --fleet-id arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111 \
    --start-time 1579647600 \
    --end-time 1579649400 \
    --limit 5
```

輸出：

```
{
    "Events": [
        {
            "EventId": "a37b6892-5d07-4d3b-8b47-80244ecf66b9",
            "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "EventCode": "FLEET_STATE_ACTIVE",
            "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed
 state to ACTIVE",
```

```
            "EventTime": 1579649342.191
        },
        {
            "EventId": "67da4ec9-92a3-4d95-886a-5d6772c24063",
            "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "EventCode": "FLEET_STATE_ACTIVATING",
            "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed
 state to ACTIVATING",
            "EventTime": 1579649321.427
        },
        {
            "EventId": "23813a46-a9e6-4a53-8847-f12e6a8381ac",
            "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "EventCode": "FLEET_STATE_BUILDING",
            "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed
 state to BUILDING",
            "EventTime": 1579649321.243
        },
        {
            "EventId": "3bf217d0-1d44-42f9-9202-433ed475d2e8",
            "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "EventCode": "FLEET_STATE_VALIDATING",
            "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed
 state to VALIDATING",
            "EventTime": 1579649197.449
        },
        {
            "EventId": "2ecd0130-5986-44eb-99a7-62df27741084",
            "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "EventCode": "FLEET_VALIDATION_LAUNCH_PATH_NOT_FOUND",
            "Message": "Failed to find a valid path",
            "EventTime": 1569319075.839,
            "PreSignedLogUrl": "https://gamelift-event-logs-prod-
us-west-2.s3.us-west-2.amazonaws.com/logs/fleet-83422059-8329-42a2-
a4d6-c4444386a6f8/events/2ecd0130-5986-44eb-99a7-62df27741084/
FLEET_VALIDATION_LAUNCH_PATH_NOT_FOUND.txt?X-Amz-Security-
Token=IQoJb3JpZ2luX2VjEB8aCXVzLXdlc3QtMiJHMEUCIHV5K%2FLPx8h310D
%2FAvx0%2FZxsDy5XA3cJOwPdu3T0eBa%2FAiEA1yovokcZYy%2FV4CWW6l26aFyiSHO
%2Bxz%2FBMAhEHYHMQNcqkQMImP%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F
%2FARAAGgw3NDEwNjE1OTIxNzEiDI8rsZtzLzlwEDQhXSrlAtl5Ae
%2Fgo6FCIzqXPbXfBOnSvFYqeDlriZarEpKqKrUt8mXQv9iqHResqCph9AKo49lwgSYTT2QoSxnrD7%2FUgv
%2BZm2pVuczvuKtUA0fcx6s0GxpjIAzdIE%2F5P%2FB7B9M%2BVZ
%2F9KF82hbJi0HTE6Y7BjKsEgFCvk4UXILhfjtan9iQl8%2F21ZTurAcJbm7Y5tuLF9SWSK3%2BEa7VXOcCK4D4O1sMj
%2FIaXoHkNvg0RVTa0hIqdvpaDQlsSBNdqTXbjHTu6fETE9Y9Ky%2BiJK5KiUG
```

```
%2F59GjCpDcvS1FqKeLUEmKT7wysGmvjMc2n%2Fr
%2F9VxQfte7w9srXwlLAQuwhiXAAyI5ICMZ5JvzjzQwTqD4CHTVKUUDwL
%2BRZzbuuqkJObZml02CkRGp%2B74RTAzLbWptVqZTIfzctiCTmWxb
%2FmKyELRYsVLrwNJ%2BGJ7%2BCrN0RC%2FjlgfLYIZyeAqjPgAu5HjgX
%2BM7jCo9M7wBTrnAXKOFQuf9dvA84SuwXOJFp17LYGjrHMKv0qC3GfbTMrZ6kzeNV9awKCpXB2Gnx9z2KvIlJdqirWV
%2F9C6%2B4jIZPME3jXmZcEHqqw5uvAVF7aeIavtUZU8pxpDIWT0YE4p3Kriy2AA7ziCRKtVfjV839InyLk8LUjsioWK
%2BYUq8%2FDTlLxqj1S%2FiO4TI0Wo7ilAo%2FKKWWF4guuNDexj8EOOynSp1yImB
%2BZf2Fua3O44W4eEXAMPLE33333&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20170621T231808Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20170621%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-
Signature=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
        }
    ],
    "NextToken":
 "eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}
```

如需詳細資訊，請參閱《Amazon GameLift 開發人員指南》中的偵錯 GameLift 機群問題。GameLift

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFleetEvents。

describe-fleet-port-settings

以下程式碼範例顯示如何使用 describe-fleet-port-settings。

AWS CLI

檢視機群的傳入連線許可

下列describe-fleet-port-settings範例會擷取指定機群的連線設定。

```
aws gamelift describe-fleet-port-settings \
    --fleet-id arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111
```

輸出：

```
{
    "InboundPermissions": [
        {
            "FromPort": 33400,
```

```
            "ToPort": 33500,
            "IpRange": "0.0.0.0/0",
            "Protocol": "UDP"
        },
        {
            "FromPort": 1900,
            "ToPort": 2000,
            "IpRange": "0.0.0.0/0",
            "Protocol": "TCP"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon GameLift 開發人員指南》中的設定 GameLift 機群。 GameLift

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFleetPortSettings。

describe-fleet-utilization

以下程式碼範例顯示如何使用 describe-fleet-utilization。

AWS CLI

Example1：檢視機群清單的用量資料

下列describe-fleet-utilization範例會擷取一個指定機群的目前用量資訊。

```
aws gamelift describe-fleet-utilization \
    --fleet-ids arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "FleetUtilization": [
        {
        "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "ActiveServerProcessCount": 100,
        "ActiveGameSessionCount": 62,
        "CurrentPlayerSessionCount": 329,
        "MaximumPlayerSessionCount": 1000
        }
```

```
    ]
}
```

Example2：請求所有機群的用量資料

下列會describe-fleet-utilization傳回任何狀態之所有機群的機群用量資料。此範例使用分頁參數一次傳回兩個機群的資料。

```
aws gamelift describe-fleet-utilization \
    --limit 2
```

輸出：

```
{
    "FleetUtilization": [
        {
            "FleetId": "fleet-1111aaaa-22bb-33cc-44dd-5555eeee66ff",
            "ActiveServerProcessCount": 100,
            "ActiveGameSessionCount": 13,
            "CurrentPlayerSessionCount": 98,
            "MaximumPlayerSessionCount": 1000
        },
        {
            "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
            "ActiveServerProcessCount": 100,
            "ActiveGameSessionCount": 62,
            "CurrentPlayerSessionCount": 329,
            "MaximumPlayerSessionCount": 1000
        }
    ],
    "NextToken":
 "eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}
```

再次呼叫 命令，將 NextToken值做為引數傳遞至 --next-token 參數，以查看接下來的兩個結果。

```
aws gamelift describe-fleet-utilization \
    --limit 2 \
    --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZT
```

重複此動作，直到回應不再包含輸出中的NextToken值為止。

如需詳細資訊，請參閱《Amazon GameLift 開發人員指南》中的機群 GameLift 指標。 GameLift

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFleetUtilization。

describe-game-session-queues

以下程式碼範例顯示如何使用 describe-game-session-queues。

AWS CLI

檢視遊戲工作階段佇列

下列describe-game-session-queues範例會擷取兩個指定佇列的屬性。

```
aws gamelift describe-game-session-queues \
    --names MegaFrogRace-NA MegaFrogRace-EU
```

輸出：

```
{
    "GameSessionQueues": [{
            "Destinations": [{
                    "DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
                },
                {
                    "DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
                }
            ],
            "Name": "MegaFrogRace-NA",
            "TimeoutInSeconds": 600,
            "GameSessionQueueArn": "arn:aws:gamelift:us-west-2::gamesessionqueue/
MegaFrogRace-NA",
            "PlayerLatencyPolicies": [{
                    "MaximumIndividualPlayerLatencyMilliseconds": 200
                },
                {
                    "MaximumIndividualPlayerLatencyMilliseconds": 100,
                    "PolicyDurationSeconds": 60
                }
```

```
            ],
            "FilterConfiguration": {
                "AllowedLocations": ["us-west-2", "ap-south-1", "us-east-1"]
            },
            "PriorityConfiguration": {
                "PriorityOrder": ["LOCATION", "FLEET_TYPE", "DESTINATION"],
                "LocationOrder": ["us-west-2", "ap-south-1", "us-east-1"]
            }
        },
        {
            "Destinations": [{
                "DestinationArn": "arn:aws:gamelift:eu-west-3::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
            }],
            "Name": "MegaFrogRace-EU",
            "TimeoutInSeconds": 600,
            "GameSessionQueueArn": "arn:aws:gamelift:us-west-2::gamesessionqueue/
MegaFrogRace-EU"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon GameLift 開發人員指南》中的使用多區域佇列。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeGameSessionQueues。

describe-runtime-configuration

以下程式碼範例顯示如何使用 describe-runtime-configuration。

AWS CLI

請求機群的執行時間組態

下列describe-runtime-configuration範例會擷取指定機群目前執行時間組態的詳細資訊。

```
aws gamelift describe-runtime-configuration \
    --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
```

```
    "RuntimeConfiguration": {
        "ServerProcesses": [
            {
                "LaunchPath": "C:\game\Bin64.Release.Dedicated
\MegaFrogRace_Server.exe",
                "Parameters": "+gamelift_start_server",
                "ConcurrentExecutions": 3
            },
            {
                "LaunchPath": "C:\game\Bin64.Release.Dedicated
\MegaFrogRace_Server.exe",
                "Parameters": "+gamelift_start_server +debug",
                "ConcurrentExecutions": 1
            }
        ],
        "MaxConcurrentGameSessionActivations": 2147483647,
        "GameSessionActivationTimeoutSeconds": 300
    }
}
```

如需詳細資訊，請參閱《Amazon GameLift 開發人員指南》中的在機群上執行多個程序。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRuntimeConfiguration。

list-builds

以下程式碼範例顯示如何使用 `list-builds`。

AWS CLI

Example1：取得自訂遊戲組建的清單

下列`list-builds`範例會擷取目前區域中所有遊戲伺服器組建的屬性。範例請求說明如何使用分頁參數 `Limit`和 `NextToken`，以循序集擷取結果。第一個命令會擷取前兩個組建。由於有兩個以上的可用，回應會包含 `NextToken`，表示有更多結果可用。

```
aws gamelift list-builds \
    --limit 2
```

輸出：

```
{
```

```
    "Builds": [
        {
            "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
            "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "CreationTime": 1495664528.723,
            "Name": "My_Game_Server_Build_One",
            "OperatingSystem": "WINDOWS_2012",
            "SizeOnDisk": 8567781,
            "Status": "READY",
            "Version": "12345.678"
        },
        {
            "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222",
            "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "CreationTime": 1495528748.555,
            "Name": "My_Game_Server_Build_Two",
            "OperatingSystem": "AMAZON_LINUX_2",
            "SizeOnDisk": 8567781,
            "Status": "FAILED",
            "Version": "23456.789"
        }
    ],
    "NextToken":
 "eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}
```

然後，您可以使用 `--next-token` 參數再次呼叫 命令，如下所示以查看接下來的兩個組建。

```
aws gamelift list-builds \
    --limit 2
    --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZT
```

重複此動作，直到回應不包含`NextToken`值為止。

Example2：取得失敗狀態的自訂遊戲組建清單

下列`list-builds`範例會在目前狀態為 FAILED 的區域中，擷取所有遊戲伺服器組建的屬性。

```
aws gamelift list-builds \
```

```
    --status FAILED
```

輸出：

```
{
    "Builds": [
        {
            "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222",
            "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "CreationTime": 1495528748.555,
            "Name": "My_Game_Server_Build_Two",
            "OperatingSystem": "AMAZON_LINUX_2",
            "SizeOnDisk": 8567781,
            "Status": "FAILED",
            "Version": "23456.789"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBuilds。

list-fleets

以下程式碼範例顯示如何使用 `list-fleets`。

AWS CLI

Example1：取得區域中所有機群的清單

下列`list-fleets`範例顯示目前區域中所有機群的機群 IDs。此範例使用分頁參數一次擷取兩個機群 IDs。回應包含`next-token`屬性，這表示要擷取的結果更多。

```
aws gamelift list-fleets \
    --limit 2
```

輸出：

```
{
    "FleetIds": [
        "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```
        "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
    ],
    "NextToken":
 "eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}
```

您可以在下一個命令中傳遞先前回應NextToken的值，如此處所示，以取得接下來的兩個結果。

```
aws gamelift list-fleets \
    --limit 2 \
    --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC00NDRlZj
```

Example2：取得具有特定建置或指令碼之區域中所有機群的清單

下列list-builds範例會擷取使用指定遊戲組建部署的機群 IDs。如果您使用的是即時伺服器，您可以提供指令碼 ID 來取代組建 ID。由於此範例未指定限制參數，因此結果最多可包含 16 個機群 IDs。

```
aws gamelift list-fleets \
    --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "FleetIds": [
        "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
        "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE44444"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFleets。

request-upload-credentials

以下程式碼範例顯示如何使用 request-upload-credentials。

AWS CLI

重新整理上傳組建的存取憑證

下列create-build範例會取得新的有效存取憑證，以將 GameLift 建置檔案上傳至 Amazon S3 位置。登入資料的生命週期有限。您可以從原始CreateBuild請求的回應中取得建置 ID。

```
aws gamelift request-upload-credentials \
    --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "StorageLocation": {
        "Bucket": "gamelift-builds-us-west-2",
        "Key": "123456789012/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    },
    "UploadCredentials": {
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "SessionToken": "AgoGb3JpZ2luENz...EXAMPLETOKEN=="
    }
}
```

如需詳細資訊，請參閱《Amazon GameLift 開發人員指南》中的將自訂伺服器建置上傳至 GameLift。 GameLift

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RequestUploadCredentials。

start-fleet-actions

以下程式碼範例顯示如何使用 start-fleet-actions。

AWS CLI

重新啟動機群自動擴展活動

下列start-fleet-actions範例會繼續使用為指定機群定義但呼叫 ``stop-fleet-actions`` 而停止的所有擴展政策。開始之後，擴展政策會立即開始追蹤各自的指標。

```
aws gamelift start-fleet-actions \
    --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --actions AUTO_SCALING
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartFleetActions。

stop-fleet-actions

以下程式碼範例顯示如何使用 stop-fleet-actions。

AWS CLI

停止機群的自動擴展活動

下列stop-fleet-actions範例會停止使用為指定機群定義的所有擴展政策。政策暫停後，除非您手動調整，否則機群容量會保持在相同的作用中執行個體計數。

```
aws gamelift start-fleet-actions \
    --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --actions AUTO_SCALING
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopFleetActions。

update-build

以下程式碼範例顯示如何使用 update-build。

AWS CLI

更新自訂遊戲組建

下列update-build範例會變更與指定建置資源相關聯的名稱和版本資訊。傳回的建置物件會驗證已成功進行變更。

```
aws gamelift update-build \
    --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --name MegaFrogRaceServer.NA.east \
    --build-version 12345.east
```

輸出：

```
{
    "Build": {
```

```
        "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "CreationTime": 1496708916.18,
        "Name": "MegaFrogRaceServer.NA.east",
        "OperatingSystem": "AMAZON_LINUX_2",
        "SizeOnDisk": 1304924,
        "Status": "READY",
        "Version": "12345.east"
    }
}
```

如需詳細資訊，請參閱《Amazon GameLift 開發人員指南》中的更新您的建置檔案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateBuild。

update-game-session-queue

以下程式碼範例顯示如何使用 update-game-session-queue。

AWS CLI

更新遊戲工作階段佇列組態

下列update-game-session-queue範例會新增新的目的地，並更新現有遊戲工作階段佇列的玩家延遲政策。

```
aws gamelift update-game-session-queue \
    --name MegaFrogRace-NA \
    --destinations file://destinations.json \
    --player-latency-policies file://latency-policies.json
```

destinations.json 的內容：

```
{
    "Destinations": [
        {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/
fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d"},
        {"DestinationArn": "arn:aws:gamelift:us-east-1::fleet/
fleet-5c6d3c4d-5e6f-7a8b-9c0d-1e2f3a4b5a2b"},
        {"DestinationArn": "arn:aws:gamelift:us-east-1::alias/
alias-11aa22bb-3c4d-5e6f-000a-1111aaaa22bb"}
    ]
```

```
}
```

latency-policies.json 的內容：

```
{
    "PlayerLatencyPolicies": [
        {"MaximumIndividualPlayerLatencyMilliseconds": 200},
        {"MaximumIndividualPlayerLatencyMilliseconds": 150, "PolicyDurationSeconds":
 120},
        {"MaximumIndividualPlayerLatencyMilliseconds": 100, "PolicyDurationSeconds":
 120}
    ]
}
```

輸出：

```
{
    "GameSessionQueue": {
        "Destinations": [
            {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/
fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d"},
            {"DestinationArn": "arn:aws:gamelift:us-east-1::fleet/
fleet-5c6d3c4d-5e6f-7a8b-9c0d-1e2f3a4b5a2b"},
            {"DestinationArn": "arn:aws:gamelift:us-east-1::alias/
alias-11aa22bb-3c4d-5e6f-000a-1111aaaa22bb"}
        ],
        "GameSessionQueueArn": "arn:aws:gamelift:us-
west-2:111122223333:gamesessionqueue/MegaFrogRace-NA",
        "Name": "MegaFrogRace-NA",
        "TimeoutInSeconds": 600,
        "PlayerLatencyPolicies": [
            {"MaximumIndividualPlayerLatencyMilliseconds": 200},
            {"MaximumIndividualPlayerLatencyMilliseconds": 150,
 "PolicyDurationSeconds": 120},
            {"MaximumIndividualPlayerLatencyMilliseconds": 100,
 "PolicyDurationSeconds": 120}
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon GameLift 開發人員指南》中的使用多區域佇列。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateGameSessionQueue。

upload-build

以下程式碼範例顯示如何使用 upload-build。

AWS CLI

Example1：上傳 Linux 遊戲伺服器組建

下列upload-build範例會將 Linux 遊戲伺服器建置檔案從檔案目錄上傳至 GameLift 服務，並建立建置資源。

```
aws gamelift upload-build \
    --name MegaFrogRaceServer.NA \
    --build-version 2.0.1 \
    --build-root ~/MegaFrogRace_Server/release-na \
    --operating-system AMAZON_LINUX_2
    --server-sdk-version 4.0.2
```

輸出：

```
Uploading ~/MegaFrogRace_Server/release-na:  16.0 KiB / 74.6 KiB (21.45%)
Uploading ~/MegaFrogRace_Server/release-na:  32.0 KiB / 74.6 KiB (42.89%)
Uploading ~/MegaFrogRace_Server/release-na:  48.0 KiB / 74.6 KiB (64.34%)
Uploading ~/MegaFrogRace_Server/release-na:  64.0 KiB / 74.6 KiB (85.79%)
Uploading ~/MegaFrogRace_Server/release-na:  74.6 KiB / 74.6 KiB (100.00%)
Successfully uploaded ~/MegaFrogRace_Server/release-na to AWS GameLift
Build ID: build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Example2：上傳 Windows 遊戲伺服器組建

下列upload-build範例會將 Windows 遊戲伺服器建置檔案從目錄上傳至 GameLift 服務，並建立建置記錄。

```
aws gamelift upload-build \
    --name MegaFrogRaceServer.NA \
    --build-version 2.0.1 \
    --build-root C:\MegaFrogRace_Server\release-na \
    --operating-system WINDOWS_2012
    --server-sdk-version 4.0.2
```

輸出：

```
Uploading C:\MegaFrogRace_Server\release-na:  16.0 KiB / 74.6 KiB (21.45%)
Uploading C:\MegaFrogRace_Server\release-na:  32.0 KiB / 74.6 KiB (42.89%)
Uploading C:\MegaFrogRace_Server\release-na:  48.0 KiB / 74.6 KiB (64.34%)
Uploading C:\MegaFrogRace_Server\release-na:  64.0 KiB / 74.6 KiB (85.79%)
Uploading C:\MegaFrogRace_Server\release-na:  74.6 KiB / 74.6 KiB (100.00%)
Successfully uploaded C:\MegaFrogRace_Server\release-na to AWS GameLift
Build ID: build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

如需詳細資訊，請參閱《Amazon GameLift 開發人員指南》中的將自訂伺服器建置上傳至 GameLift。 GameLift

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UploadBuild。

使用 的 Global Accelerator 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Global Accelerator 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`add-custom-routing-endpoints`

以下程式碼範例顯示如何使用 `add-custom-routing-endpoints`。

AWS CLI

將 VPC 子網路端點新增至自訂路由加速器的端點群組

下列`add-custom-routing-endpoints`範例會將 VPC 子網路端點新增至自訂路由加速器的端點群組。

```
aws globalaccelerator add-custom-routing-endpoints \
```

```
    --endpoint-group-
arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/4321abcd \
    --endpoint-configurations "EndpointId=subnet-1234567890abcdef0"
```

輸出：

```
{
    "EndpointDescriptions": [
        {
            "EndpointId": "subnet-1234567890abcdef0"
        }
    ],

 "EndpointGroupArn":"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/4321abcd"
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的自訂路由加速器的 VPC 子網路端點。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddCustomRoutingEndpoints。

advertise-byoip-cidr

以下程式碼範例顯示如何使用 advertise-byoip-cidr。

AWS CLI

公告地址範圍

下列advertise-byoip-cidr範例 AWS 請求公告您佈建用於 AWS 資源的地址範圍。

```
aws globalaccelerator advertise-byoip-cidr \
    --cidr 198.51.100.0/24
```

輸出：

```
{
    "ByoipCidr": {
        "Cidr": "198.51.100.0/24",
```

```
        "State": "PENDING_ADVERTISING"
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的在 Global Accelerator 中使用您自己的 IP 地址。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AdvertiseByoipCidr。

`allow-custom-routing-traffic`

以下程式碼範例顯示如何使用 `allow-custom-routing-traffic`。

AWS CLI

允許流量傳送至自訂路由加速器 VPC 子網路中的特定 Amazon EC2 執行個體目的地

下列`allow-custom-routing-traffic`範例指定允許流量到自訂路由加速器中 VPC 子網路端點的特定 Amazon EC2 執行個體 （ 目的地 ） IP 地址和連接埠，可以接收流量。

```
aws globalaccelerator allow-custom-routing-traffic \
    --endpoint-group-
arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/ab88888example \
    --endpoint-id subnet-abcd123example \
    --destination-addresses "172.31.200.6" "172.31.200.7" \
    --destination-ports 80 81
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的自訂路由加速器的 VPC 子網路端點。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AllowCustomRoutingTraffic。

`create-accelerator`

以下程式碼範例顯示如何使用 `create-accelerator`。

AWS CLI

建立加速器

下列create-accelerator範例會建立具有兩個標籤的加速器，其中包含兩個 BYOIP 靜態 IP 地址。您必須指定要建立或更新加速器US-West-2 (Oregon)的區域。

```
aws globalaccelerator create-accelerator \
    --name ExampleAccelerator \
    --tags Key="Name",Value="Example Name" Key="Project",Value="Example Project" \
    --ip-addresses 192.0.2.250 198.51.100.52
```

輸出：

```
{
    "Accelerator": {
        "AcceleratorArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh",
        "IpAddressType": "IPV4",
        "Name": "ExampleAccelerator",
        "Enabled": true,
        "Status": "IN_PROGRESS",
        "IpSets": [
            {
                "IpAddresses": [
                    "192.0.2.250",
                    "198.51.100.52"
                ],
                "IpFamily": "IPv4"
            }
        ],
        "DnsName":"a1234567890abcdef.awsglobalaccelerator.com",
        "CreatedTime": 1542394847.0,
        "LastModifiedTime": 1542394847.0
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 AWS Global Accelerator。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAccelerator。

create-custom-routing-accelerator

以下程式碼範例顯示如何使用 create-custom-routing-accelerator。

AWS CLI

建立自訂路由加速器

下列create-custom-routing-accelerator範例會建立具有標籤 Name和 的自訂路由加速器Project。

```
aws globalaccelerator create-custom-routing-accelerator \
    --name ExampleCustomRoutingAccelerator \
    --tags Key="Name",Value="Example Name" Key="Project",Value="Example Project" \
    --ip-addresses 192.0.2.250 198.51.100.52
```

輸出：

```
{
    "Accelerator": {
        "AcceleratorArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh",
        "IpAddressType": "IPV4",
        "Name": "ExampleCustomRoutingAccelerator",
        "Enabled": true,
        "Status": "IN_PROGRESS",
        "IpSets": [
            {
                "IpAddresses": [
                    "192.0.2.250",
                    "198.51.100.52"
                ],
                "IpFamily": "IPv4"
            }
        ],
        "DnsName":"a1234567890abcdef.awsglobalaccelerator.com",
        "CreatedTime": 1542394847.0,
        "LastModifiedTime": 1542394847.0
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中的自訂路由加速器。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCustomRoutingAccelerator。

create-custom-routing-endpoint-group

以下程式碼範例顯示如何使用 create-custom-routing-endpoint-group。

AWS CLI

為自訂路由加速器建立端點群組

下列create-custom-routing-endpoint-group範例會建立自訂路由加速器的端點群組。

```
aws globalaccelerator create-custom-routing-endpoint-group \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \
    --endpoint-group-region us-east-2 \
    --destination-configurations "FromPort=80,ToPort=81,Protocols=TCP,UDP"
```

輸出：

```
{
    "EndpointGroup": {
        "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/4321abcd",
        "EndpointGroupRegion": "us-east-2",
        "DestinationDescriptions": [
            {
                "FromPort": 80,
                "ToPort": 81,
                "Protocols": [
                    "TCP",
                    "UDP"
                ]
            }
        ],
        "EndpointDescriptions": []
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中自訂路由加速器的端點群組。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCustomRoutingEndpointGroup。

create-custom-routing-listener

以下程式碼範例顯示如何使用 create-custom-routing-listener。

AWS CLI

為自訂路由加速器建立接聽程式

下列create-custom-routing-listener範例會為自訂路由加速器建立連接埠範圍從 5000 到 10000 的接聽程式。

```
aws globalaccelerator create-custom-routing-listener \
    --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --port-ranges FromPort=5000,ToPort=10000
```

輸出：

```
{
    "Listener": {
        "PortRange": [
            "FromPort": 5000,
            "ToPort": 10000
        ],
        "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz"
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中自訂路由加速器的接聽程式。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCustomRoutingListener。

create-endpoint-group

以下程式碼範例顯示如何使用 create-endpoint-group。

AWS CLI

建立端點群組

下列create-endpoint-group範例會建立具有一個端點的端點群組。

```
aws globalaccelerator create-endpoint-group \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \
    --endpoint-group-region us-east-1 \
    --endpoint-configurations EndpointId=i-1234567890abcdef0,Weight=128
```

輸出：

```
{
    "EndpointGroup": {
        "TrafficDialPercentage": 100.0,
        "EndpointDescriptions": [
            {
                "Weight": 128,
                "EndpointId": "i-1234567890abcdef0"
            }
        ],
        "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/098765zyxwvu",
        "EndpointGroupRegion": "us-east-1"
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中的端點群組。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateEndpointGroup。

create-listener

以下程式碼範例顯示如何使用 create-listener。

AWS CLI

建立接聽程式

下列create-listener範例會建立具有兩個連接埠的接聽程式。

```
aws globalaccelerator create-listener \
```

```
    --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh \
    --port-ranges FromPort=80,ToPort=80 FromPort=81,ToPort=81 \
    --protocol TCP
```

輸出：

```
{
    "Listener": {
        "PortRanges": [
            {
                "ToPort": 80,
                "FromPort": 80
            },
            {
                "ToPort": 81,
                "FromPort": 81
            }
        ],
        "ClientAffinity": "NONE",
        "Protocol": "TCP",
        "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz"
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中的接聽程式。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateListener。

deny-custom-routing-traffic

以下程式碼範例顯示如何使用 deny-custom-routing-traffic。

AWS CLI

指定無法在自訂路由加速器中接收流量的目的地地址

下列deny-custom-routing-traffic範例指定子網路端點中無法接收自訂路由加速器流量的目的地地址或地址。若要指定多個目的地地址，請以空格分隔地址。對於成功的deny-custom-routing-traffic呼叫，沒有回應。

```
aws globalaccelerator deny-custom-routing-traffic \
    --endpoint-group-
arn "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/ab88888example" \
    --endpoint-id "subnet-abcd123example" \
    --destination-addresses "198.51.100.52"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的自訂路由加速器的 VPC 子網路端點。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DenyCustomRoutingTraffic。

deprovision-byoip-cidr

以下程式碼範例顯示如何使用 deprovision-byoip-cidr。

AWS CLI

取消佈建地址範圍

下列deprovision-byoip-cidr範例會釋出您佈建用於 AWS 資源的指定地址範圍。

```
aws globalaccelerator deprovision-byoip-cidr \
    --cidr "198.51.100.0/24"
```

輸出：

```
{
    "ByoipCidr": {
        "Cidr": "198.51.100.0/24",
        "State": "PENDING_DEPROVISIONING"
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的在 Global Accelerator 中使用您自己的 IP 地址。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeprovisionByoipCidr。

describe-accelerator-attributes

以下程式碼範例顯示如何使用 describe-accelerator-attributes。

AWS CLI

描述加速器的屬性

下列describe-accelerator-attributes範例會擷取 加速器的屬性詳細資訊。

```
aws globalaccelerator describe-accelerator-attributes \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

輸出：

```
{
    "AcceleratorAttributes": {
        "FlowLogsEnabled": true
        "FlowLogsS3Bucket": flowlogs-abc
        "FlowLogsS3Prefix": bucketprefix-abc
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 AWS Global Accelerator。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAcceleratorAttributes。

describe-accelerator

以下程式碼範例顯示如何使用 describe-accelerator。

AWS CLI

描述加速器

下列describe-accelerator範例會擷取指定加速器的詳細資訊。

```
aws globalaccelerator describe-accelerator \
    --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

輸出：

```
{
    "Accelerator": {
        "AcceleratorArn":
 "arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh",
        "IpAddressType": "IPV4",
        "Name": "ExampleAccelerator",
        "Enabled": true,
        "Status": "IN_PROGRESS",
        "IpSets": [
            {
                "IpAddresses": [
                    "192.0.2.250",
                    "198.51.100.52"
                ],
                "IpFamily": "IPv4"
            }
        ],
        "DnsName":"a1234567890abcdef.awsglobalaccelerator.com",
        "CreatedTime": 1542394847,
        "LastModifiedTime": 1542395013
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 AWS Global Accelerator。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAccelerator。

describe-custom-routing-accelerator-attributes

以下程式碼範例顯示如何使用 describe-custom-routing-accelerator-attributes。

AWS CLI

描述自訂路由加速器的屬性

下列describe-custom-routing-accelerator-attributes範例說明自訂路由加速器的屬性。

```
aws globalaccelerator describe-custom-routing-accelerator-attributes \
   --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh
```

輸出：

```
{
    "AcceleratorAttributes": {
    "FlowLogsEnabled": false
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中的自訂路由加速器。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCustomRoutingAcceleratorAttributes。

describe-custom-routing-accelerator

以下程式碼範例顯示如何使用 describe-custom-routing-accelerator。

AWS CLI

描述自訂路由加速器

下列describe-custom-routing-accelerator範例會擷取指定自訂路由加速器的詳細資訊。

```
aws globalaccelerator describe-custom-routing-accelerator \
    --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

輸出：

```
{
    "Accelerator": {
        "AcceleratorArn":
 "arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh",
        "IpAddressType": "IPV4",
        "Name": "ExampleCustomRoutingAccelerator",
        "Enabled": true,
        "Status": "IN_PROGRESS",
        "IpSets": [
            {
                "IpAddresses": [
```

```
                    "192.0.2.250",
                    "198.51.100.52"
                ],
                "IpFamily": "IPv4"
            }
        ],
        "DnsName":"a1234567890abcdef.awsglobalaccelerator.com",
        "CreatedTime": 1542394847,
        "LastModifiedTime": 1542395013
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中的自訂路由加速器。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCustomRoutingAccelerator。

describe-custom-routing-endpoint-group

以下程式碼範例顯示如何使用 describe-custom-routing-endpoint-group。

AWS CLI

描述自訂路由加速器的端點群組

下列describe-custom-routing-endpoint-group範例說明自訂路由加速器的端點群組。

```
aws globalaccelerator describe-custom-routing-endpoint-group \
    --endpoint-group-
arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz/endpoint-group/ab88888example
```

輸出：

```
{
    "EndpointGroup": {
        "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz/endpoint-group/ab88888example",
        "EndpointGroupRegion": "us-east-2",
        "DestinationDescriptions": [
            {
                "FromPort": 5000,
```

```
                "ToPort": 10000,
                "Protocols": [
                    "UDP"
                ]
            }
        ],
        "EndpointDescriptions": [
            {
                "EndpointId": "subnet-1234567890abcdef0"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中自訂路由加速器的端點群組。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCustomRoutingEndpointGroup。

describe-custom-routing-listener

以下程式碼範例顯示如何使用 describe-custom-routing-listener。

AWS CLI

描述自訂路由加速器的接聽程式

下列describe-custom-routing-listener範例說明自訂路由加速器的接聽程式。

```
aws globalaccelerator describe-custom-routing-listener \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

輸出：

```
{
    "Listener": {
        "PortRanges": [
            "FromPort": 5000,
            "ToPort": 10000
        ],
```

```
        "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234"
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中自訂路由加速器的接聽程式。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCustomRoutingListener。

describe-endpoint-group

以下程式碼範例顯示如何使用 describe-endpoint-group。

AWS CLI

描述端點群組

下列describe-endpoint-group範例會擷取具有下列端點的端點群組詳細資訊：Amazon EC2 執行個體、ALB 和 NLB。

```
aws globalaccelerator describe-endpoint-group \
    --endpoint-group-
arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-group/
ab88888example
```

輸出：

```
{
    "EndpointGroup": {
        "TrafficDialPercentage": 100.0,
        "EndpointDescriptions": [
        {
            "Weight": 128,
            "EndpointId": "i-1234567890abcdef0"
        },
        {
            "Weight": 128,
            "EndpointId": "arn:aws:elasticloadbalancing:us-
east-1:000123456789:loadbalancer/app/ALBTesting/alb01234567890xyz"
```

```
        },
        {
            "Weight": 128,
            "EndpointId": "arn:aws:elasticloadbalancing:us-
east-1:000123456789:loadbalancer/net/NLBTesting/alb01234567890qrs"
        }
        ],
        "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-
group/4321abcd-abcd-4321-abcd-4321abcdefg",
        "EndpointGroupRegion": "us-east-1"
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中的端點群組。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEndpointGroup。

describe-listener

以下程式碼範例顯示如何使用 describe-listener。

AWS CLI

描述接聽程式

下列describe-listener範例說明接聽程式。

```
aws globalaccelerator describe-listener \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

輸出：

```
{
    "Listener": {
        "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234",
        "PortRanges": [
            {
```

```
                "FromPort": 80,
                "ToPort": 80
            }
        ],
        "Protocol": "TCP",
        "ClientAffinity": "NONE"
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中的接聽程式。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeListener。

`list-accelerators`

以下程式碼範例顯示如何使用 `list-accelerators`。

AWS CLI

列出您的加速器

下列`list-accelerators`範例列出您 AWS 帳戶中的加速器。此帳戶有兩個加速器。

```
aws globalaccelerator list-accelerators
```

輸出：

```
{
    "Accelerators": [
        {
            "AcceleratorArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/5555abcd-abcd-5555-
abcd-5555EXAMPLE1",
            "Name": "TestAccelerator",
            "IpAddressType": "IPV4",
            "Enabled": true,
            "IpSets": [
                {
                    "IpFamily": "IPv4",
                    "IpAddresses": [
                        "192.0.2.250",
                        "198.51.100.52"
```

```
                    ]
                }
            ],
            "DnsName": "5a5a5a5a5a5a5a5a.awsglobalaccelerator.com",
            "Status": "DEPLOYED",
            "CreatedTime": 1552424416.0,
            "LastModifiedTime": 1569375641.0
        },
        {
            "AcceleratorArn":
 "arn:aws:globalaccelerator::888888888888:accelerator/8888abcd-abcd-8888-
abcd-8888EXAMPLE2",
            "Name": "ExampleAccelerator",
            "IpAddressType": "IPV4",
            "Enabled": true,
            "IpSets": [
                {
                    "IpFamily": "IPv4",
                    "IpAddresses": [
                        "192.0.2.100",
                        "198.51.100.10"
                    ]
                }
            ],
            "DnsName": "6a6a6a6a6a6a6a6a.awsglobalaccelerator.com",
            "Status": "DEPLOYED",
            "CreatedTime": 1575585564.0,
            "LastModifiedTime": 1579809243.0
        },
    ]
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 AWS Global Accelerator。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAccelerators。

`list-byoip-cidr`

以下程式碼範例顯示如何使用 `list-byoip-cidr`。

AWS CLI

列出您的地址範圍

下列list-byoip-cidr範例列出您已佈建用於 Global Accelerator 的自有 IP 地址 (BYOIP) 地址範圍。

```
aws globalaccelerator list-byoip-cidrs
```

輸出：

```
{
    "ByoipCidrs": [
        {
            "Cidr": "198.51.100.0/24",
            "State": "READY"
        }
        {
            "Cidr": "203.0.113.25/24",
            "State": "READY"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的在 Global Accelerator 中使用您自己的 IP 地址。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListByoipCidr。

list-custom-routing-accelerators

以下程式碼範例顯示如何使用 list-custom-routing-accelerators。

AWS CLI

列出您的自訂路由加速器

下列list-custom-routing-accelerators範例列出 AWS 帳戶中的自訂路由加速器。

```
aws globalaccelerator list-custom-routing-accelerators
```

輸出：

```
{
    "Accelerators": [
        {
```

```
            "AcceleratorArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/5555abcd-abcd-5555-
abcd-5555EXAMPLE1",
            "Name": "TestCustomRoutingAccelerator",
            "IpAddressType": "IPV4",
            "Enabled": true,
            "IpSets": [
                {
                    "IpFamily": "IPv4",
                    "IpAddresses": [
                        "192.0.2.250",
                        "198.51.100.52"
                    ]
                }
            ],
            "DnsName": "5a5a5a5a5a5a5a5a.awsglobalaccelerator.com",
            "Status": "DEPLOYED",
            "CreatedTime": 1552424416.0,
            "LastModifiedTime": 1569375641.0
        },
        {
            "AcceleratorArn":
 "arn:aws:globalaccelerator::888888888888:accelerator/8888abcd-abcd-8888-
abcd-8888EXAMPLE2",
            "Name": "ExampleCustomRoutingAccelerator",
            "IpAddressType": "IPV4",
            "Enabled": true,
            "IpSets": [
                {
                    "IpFamily": "IPv4",
                    "IpAddresses": [
                        "192.0.2.100",
                        "198.51.100.10"
                    ]
                }
            ],
            "DnsName": "6a6a6a6a6a6a6a6a.awsglobalaccelerator.com",
            "Status": "DEPLOYED",
            "CreatedTime": 1575585564.0,
            "LastModifiedTime": 1579809243.0
        },
    ]
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中的自訂路由加速器。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCustomRoutingAccelerators。

`list-custom-routing-endpoint-groups`

以下程式碼範例顯示如何使用 `list-custom-routing-endpoint-groups`。

AWS CLI

列出自訂路由加速器中接聽程式的端點群組

下列`list-custom-routing-endpoint-groups`範例列出自訂路由加速器中接聽程式的端點群組。

```
aws globalaccelerator list-custom-routing-endpoint-groups \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

輸出：

```
{
    "EndpointGroups": [
        {
            "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234/endpoint-group/ab88888example",
            "EndpointGroupRegion": "eu-central-1",
            "DestinationDescriptions": [
                {
                    "FromPort": 80,
                    "ToPort": 80,
                    "Protocols": [
                        "TCP",
                        "UDP"
                    ]
                }
            ]
            "EndpointDescriptions": [
                {
                    "EndpointId": "subnet-abcd123example"
                }
```

```
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中自訂路由加速器的端點群組。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCustomRoutingEndpointGroups。

`list-custom-routing-listeners`

以下程式碼範例顯示如何使用 `list-custom-routing-listeners`。

AWS CLI

列出自訂路由加速器的接聽程式

下列`list-custom-routing-listeners`範例列出自訂路由加速器的接聽程式。

```
aws globalaccelerator list-custom-routing-listeners \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

輸出：

```
{
    "Listeners": [
        {
            "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234",
            "PortRanges": [
                {
                    "FromPort": 5000,
                    "ToPort": 10000
                }
            ],
            "Protocol": "TCP"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中自訂路由加速器的接聽程式。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCustomRoutingListeners。

`list-custom-routing-port-mappings-by-destination`

以下程式碼範例顯示如何使用 `list-custom-routing-port-mappings-by-destination`。

AWS CLI

列出特定自訂路由加速器目的地的連接埠映射

下列`list-custom-routing-port-mappings-by-destination`範例提供自訂路由加速器特定目的地 EC2 伺服器（位於目的地地址）的連接埠映射。

```
aws globalaccelerator list-custom-routing-port-mappings-by-destination \
    --endpoint-id subnet-abcd123example \
    --destination-address 198.51.100.52
```

輸出：

```
{
    "DestinationPortMappings": [
        {
            "AcceleratorArn":
 "arn:aws:globalaccelerator::402092451327:accelerator/24ea29b8-
d750-4489-8919-3095f3c4b0a7",
                "AcceleratorSocketAddresses": [
                    {
                        "IpAddress": "192.0.2.250",
                        "Port": 65514
                    },
                    {
                        "IpAddress": "192.10.100.99",
                        "Port": 65514
                    }
                ],
                "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/ab88888example",
                "EndpointId": "subnet-abcd123example",
                "EndpointGroupRegion": "us-west-2",
```

```
                "DestinationSocketAddress": {
                    "IpAddress": "198.51.100.52",
                    "Port": 80
                },
                "IpAddressType": "IPv4",
                "DestinationTrafficState": "ALLOW"
            }
        ]
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的自訂路由加速器在 Global Accelerator 中的運作方式。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCustomRoutingPortMappingsByDestination。

`list-custom-routing-port-mappings`

以下程式碼範例顯示如何使用 `list-custom-routing-port-mappings`。

AWS CLI

列出自訂路由加速器中的連接埠映射

下列`list-custom-routing-port-mappings`範例提供自訂路由加速器中連接埠映射的部分清單。

```
aws globalaccelerator list-custom-routing-port-mappings \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

輸出：

```
{
    "PortMappings": [
        {
            "AcceleratorPort": 40480,
            "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/098765zyxwvu",
            "EndpointId": "subnet-1234567890abcdef0",
            "DestinationSocketAddress": {
                "IpAddress": "192.0.2.250",
```

```
                "Port": 80
            },
            "Protocols": [
                "TCP",
                "UDP"
            ],
            "DestinationTrafficState": "ALLOW"
        }
        {
            "AcceleratorPort": 40481,
            "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/098765zyxwvu",
            "EndpointId": "subnet-1234567890abcdef0",
            "DestinationSocketAddress": {
               "IpAddress": "192.0.2.251",
              "Port": 80
            },
            "Protocols": [
                "TCP",
                "UDP"
            ],
            "DestinationTrafficState": "ALLOW"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的自訂路由加速器在 Global Accelerator 中的運作方式。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCustomRoutingPortMappings。

`list-endpoint-groups`

以下程式碼範例顯示如何使用 `list-endpoint-groups`。

AWS CLI

列出端點群組

下列`list-endpoint-groups`範例列出接聽程式的端點群組。此接聽程式有兩個端點群組。

```
aws globalaccelerator --region us-west-2 list-endpoint-groups \
```

```
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

輸出：

```
{
    "EndpointGroups": [
        {
            "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234/endpoint-group/ab88888example",
            "EndpointGroupRegion": "eu-central-1",
            "EndpointDescriptions": [],
            "TrafficDialPercentage": 100.0,
            "HealthCheckPort": 80,
            "HealthCheckProtocol": "TCP",
            "HealthCheckIntervalSeconds": 30,
            "ThresholdCount": 3
        }
        {
            "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234/endpoint-group/ab99999example",
            "EndpointGroupRegion": "us-east-1",
            "EndpointDescriptions": [],
            "TrafficDialPercentage": 50.0,
            "HealthCheckPort": 80,
            "HealthCheckProtocol": "TCP",
            "HealthCheckIntervalSeconds": 30,
            "ThresholdCount": 3
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中的端點群組。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEndpointGroups。

list-listeners

以下程式碼範例顯示如何使用 list-listeners。

AWS CLI

列出接聽程式

下列list-listeners範例列出 加速器的接聽程式。

```
aws globalaccelerator list-listeners \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

輸出：

```
{
    "Listeners": [
        {
            "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234",
            "PortRanges": [
                {
                    "FromPort": 80,
                    "ToPort": 80
                }
            ],
            "Protocol": "TCP",
            "ClientAffinity": "NONE"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中的接聽程式。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListListeners。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出加速器的標籤

下列list-tags-for-resource範例列出特定加速器的標籤。

```
aws globalaccelerator list-tags-for-resource \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

輸出：

```
{
    "Tags": [
        {
            "Key": "Project",
            "Value": "A123456"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的在 Global Accelerator 中標記。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

provision-byoip-cidr

以下程式碼範例顯示如何使用 provision-byoip-cidr。

AWS CLI

佈建地址範圍

下列provision-byoip-cidr範例會佈建要與 AWS 資源搭配使用的指定地址範圍。

```
aws globalaccelerator provision-byoip-cidr \
    --cidr 192.0.2.250/24 \
    --cidr-authorization-context Message="$text_message",Signature="$signed_message"
```

輸出：

```
{
    "ByoipCidr": {
        "Cidr": "192.0.2.250/24",
```

```
        "State": "PENDING_PROVISIONING"
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的在 Global Accelerator 中使用您自己的 IP 地址。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ProvisionByoipCidr。

`tag-resource`

以下程式碼範例顯示如何使用 `tag-resource`。

AWS CLI

標記加速器

下列`tag-resource`範例會將標籤名稱和專案新增至加速器，以及每個項目的對應值。

```
aws globalaccelerator tag-resource \
    --resource-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --tags Key="Name",Value="Example Name" Key="Project",Value="Example Project"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的在 Global Accelerator 中標記。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

`untag-resource`

以下程式碼範例顯示如何使用 `untag-resource`。

AWS CLI

從加速器移除標籤

下列`untag-resource`範例會從加速器移除標籤名稱和專案。

```
aws globalaccelerator untag-resource \
```

```
    --resource-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --tag-keys Key="Name" Key="Project"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的在 Global Accelerator 中標記。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-accelerator-attributes

以下程式碼範例顯示如何使用 update-accelerator-attributes。

AWS CLI

更新加速器的屬性

下列update-accelerator-attributes範例會更新 加速器以啟用流程日誌。您必須指定要建立或更新加速器屬性US-West-2 (Oregon)的區域。

```
aws globalaccelerator update-accelerator-attributes \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --flow-logs-enabled \
    --flow-logs-s3-bucket flowlogs-abc \
    --flow-logs-s3-prefix bucketprefix-abc
```

輸出：

```
{
    "AcceleratorAttributes": {
        "FlowLogsEnabled": true
        "FlowLogsS3Bucket": flowlogs-abc
        "FlowLogsS3Prefix": bucketprefix-abc
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 AWS Global Accelerator。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAcceleratorAttributes。

update-accelerator

以下程式碼範例顯示如何使用 update-accelerator。

AWS CLI

更新加速器

下列update-accelerator範例會修改加速器，將加速器名稱變更為ExampleAcceleratorNew。您必須指定要建立或更新加速器US-West-2 (Oregon)的區域。

```
aws globalaccelerator update-accelerator \
    --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --name ExampleAcceleratorNew
```

輸出：

```
{
    "Accelerator": {
        "AcceleratorArn":
 "arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh",
        "IpAddressType": "IPV4",
        "Name": "ExampleAcceleratorNew",
        "Enabled": true,
        "Status": "IN_PROGRESS",
        "IpSets": [
            {
                "IpAddresses": [
                    "192.0.2.250",
                    "198.51.100.52"
                ],
                "IpFamily": "IPv4"
            }
        ],
        "DnsName":"a1234567890abcdef.awsglobalaccelerator.com",
        "CreatedTime": 1232394847,
        "LastModifiedTime": 1232395654
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 AWS Global Accelerator。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAccelerator。

update-custom-routing-accelerator-attributes

以下程式碼範例顯示如何使用 update-custom-routing-accelerator-attributes。

AWS CLI

更新自訂路由加速器的屬性

下列update-custom-routing-accelerator-attributes範例會更新自訂路由加速器以啟用流程日誌。

```
aws globalaccelerator update-custom-routing-accelerator-attributes \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --flow-logs-enabled \
    --flow-logs-s3-bucket flowlogs-abc \
    --flow-logs-s3-prefix bucketprefix-abc
```

輸出：

```
{
    "AcceleratorAttributes": {
        "FlowLogsEnabled": true
        "FlowLogsS3Bucket": flowlogs-abc
        "FlowLogsS3Prefix": bucketprefix-abc
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中的自訂路由加速器。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateCustomRoutingAcceleratorAttributes。

update-custom-routing-accelerator

以下程式碼範例顯示如何使用 update-custom-routing-accelerator。

AWS CLI

更新自訂路由加速器

下列update-custom-routing-accelerator範例會修改自訂路由加速器以變更加速器名稱。

```
aws globalaccelerator --region us-west-2 update-custom-routing-accelerator \
    --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --name ExampleCustomRoutingAcceleratorNew
```

輸出：

```
{
    "Accelerator": {
        "AcceleratorArn":
 "arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh",
        "IpAddressType": "IPV4",
        "Name": "ExampleCustomRoutingAcceleratorNew",
        "Enabled": true,
        "Status": "IN_PROGRESS",
        "IpSets": [
            {
                "IpAddresses": [
                    "192.0.2.250",
                    "198.51.100.52"
                ],
                "IpFamily": "IPv4"
            }
        ],
        "DnsName":"a1234567890abcdef.awsglobalaccelerator.com",
        "CreatedTime": 1232394847,
        "LastModifiedTime": 1232395654
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中的自訂路由加速器。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateCustomRoutingAccelerator。

update-custom-routing-listener

以下程式碼範例顯示如何使用 update-custom-routing-listener。

AWS CLI

更新自訂路由加速器的接聽程式

下列update-custom-routing-listener範例會更新接聽程式以變更連接埠範圍。

```
aws globalaccelerator update-custom-routing-listener \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \
    --port-ranges FromPort=10000,ToPort=20000
```

輸出：

```
{
    "Listener": {
        "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz
        "PortRanges": [
            {
                "FromPort": 10000,
                "ToPort": 20000
            }
        ],
        "Protocol": "TCP"
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中自訂路由加速器的接聽程式。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateCustomRoutingListener。

update-endpoint-group

以下程式碼範例顯示如何使用 update-endpoint-group。

AWS CLI

更新端點群組

下列update-endpoint-group範例會將三個端點新增至端點群組：彈性 IP 地址、ALB 和 NLB。

```
aws globalaccelerator update-endpoint-group \
    --endpoint-group-
arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-group/
ab88888example \
    --endpoint-configurations \
        EndpointId=eipalloc-eip01234567890abc,Weight=128 \
        EndpointId=arn:aws:elasticloadbalancing:us-east-1:000123456789:loadbalancer/
app/ALBTesting/alb01234567890xyz,Weight=128 \
        EndpointId=arn:aws:elasticloadbalancing:us-east-1:000123456789:loadbalancer/
net/NLBTesting/alb01234567890qrs,Weight=128
```

輸出：

```
{
    "EndpointGroup": {
        "TrafficDialPercentage": 100,
        "EndpointDescriptions": [
            {
                "Weight": 128,
                "EndpointId": "eip01234567890abc"
            },
            {
                "Weight": 128,
                "EndpointId": "arn:aws:elasticloadbalancing:us-
east-1:000123456789:loadbalancer/app/ALBTesting/alb01234567890xyz"
            },
            {
                "Weight": 128,
                "EndpointId": "arn:aws:elasticloadbalancing:us-
east-1:000123456789:loadbalancer/net/NLBTesting/alb01234567890qrs"
            }
        ],
        "EndpointGroupArn":
 "arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-
group/4321abcd-abcd-4321-abcd-4321abcdefg",
```

```
        "EndpointGroupRegion": "us-east-1"
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中的端點群組。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateEndpointGroup。

update-listener

以下程式碼範例顯示如何使用 update-listener。

AWS CLI

更新接聽程式

下列update-listener範例會更新接聽程式，將連接埠變更為 100。

```
aws globalaccelerator update-listener \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \
    --port-ranges FromPort=100,ToPort=100
```

輸出：

```
{
    "Listener": {
        "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz
        "PortRanges": [
            {
                "FromPort": 100,
                "ToPort": 100
            }
        ],
        "Protocol": "TCP",
        "ClientAffinity": "NONE"
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的 Global Accelerator 中的接聽程式。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateListener。

`withdraw-byoip-cidr`

以下程式碼範例顯示如何使用 `withdraw-byoip-cidr`。

AWS CLI

撤銷地址範圍

下列`withdraw-byoip-cidr`範例會從您先前公告用於 AWS 資源的 AWS Global Accelerator 中撤銷地址範圍。

```
aws globalaccelerator withdraw-byoip-cidr \
    --cidr 192.0.2.250/24
```

輸出：

```
{
    "ByoipCidr": {
        "Cidr": "192.0.2.250/24",
        "State": "PENDING_WITHDRAWING"
    }
}
```

如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》中的在 Global Accelerator 中使用您自己的 IP 地址。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 WithdrawByoipCidr。

AWS Glue 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS Glue。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

batch-stop-job-run

以下程式碼範例顯示如何使用 batch-stop-job-run。

AWS CLI

停止任務執行

下列batch-stop-job-run範例會停止任務執行。

```
aws glue batch-stop-job-run \
    --job-name "my-testing-job" \
    --job-run-id jr_852f1de1f29fb62e0ba4166c33970803935d87f14f96cfdee5089d5274a61d3f
```

輸出：

```
{
    "SuccessfulSubmissions": [
        {
            "JobName": "my-testing-job",
            "JobRunId":
 "jr_852f1de1f29fb62e0ba4166c33970803935d87f14f96cfdee5089d5274a61d3f"
        }
    ],
    "Errors": [],
    "ResponseMetadata": {
        "RequestId": "66bd6b90-01db-44ab-95b9-6aeff0e73d88",
        "HTTPStatusCode": 200,
        "HTTPHeaders": {
            "date": "Fri, 16 Oct 2020 20:54:51 GMT",
            "content-type": "application/x-amz-json-1.1",
            "content-length": "148",
            "connection": "keep-alive",
            "x-amzn-requestid": "66bd6b90-01db-44ab-95b9-6aeff0e73d88"
```

```
        },
        "RetryAttempts": 0
    }
}
```

如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的工作執行。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchStopJobRun。

`create-connection`

以下程式碼範例顯示如何使用 `create-connection`。

AWS CLI

建立 Glue AWS 資料存放區的連線

下列`create-connection`範例會在 AWS Glue Data Catalog 中建立連線，以提供 Kafka 資料存放區的連線資訊。

```
aws glue create-connection \
    --connection-input '{ \
        "Name":"conn-kafka-custom", \
        "Description":"kafka connection with ssl to custom kafka", \
        "ConnectionType":"KAFKA",  \
        "ConnectionProperties":{  \
            "KAFKA_BOOTSTRAP_SERVERS":"<Kafka-broker-server-url>:<SSL-Port>", \
            "KAFKA_SSL_ENABLED":"true", \
            "KAFKA_CUSTOM_CERT": "s3://bucket/prefix/cert-file.pem" \
        }, \
        "PhysicalConnectionRequirements":{ \
            "SubnetId":"subnet-1234", \
            "SecurityGroupIdList":["sg-1234"], \
            "AvailabilityZone":"us-east-1a"} \
    }' \
    --region us-east-1
    --endpoint https://glue.us-east-1.amazonaws.com
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Glue AWS 開發人員指南》中的在 Glue Data Catalog 中定義連線。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateConnection。

create-database

以下程式碼範例顯示如何使用 create-database。

AWS CLI

建立資料庫

下列create-database範例會在 Glue Data Catalog AWS 中建立資料庫。

```
aws glue create-database \
    --database-input "{\"Name\":\"tempdb\"}" \
    --profile my_profile \
    --endpoint https://glue.us-east-1.amazonaws.com
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的在 Data Catalog 中定義資料庫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDatabase。

create-job

以下程式碼範例顯示如何使用 create-job。

AWS CLI

建立工作以轉換資料

下列 create-job 範例會建立串流工作，它可執行存放在 S3 中的指令碼。

```
aws glue create-job \
    --name my-testing-job \
    --role AWSGlueServiceRoleDefault \
    --command '{ \
        "Name": "gluestreaming", \
        "ScriptLocation": "s3://amzn-s3-demo-bucket/folder/" \
    }' \
    --region us-east-1 \
    --output json \
    --default-arguments '{ \
        "--job-language":"scala", \
        "--class":"GlueApp" \
```

```
    }' \
    --profile my-profile \
    --endpoint https://glue.us-east-1.amazonaws.com
```

test_script.scala 的內容：

```
import com.amazonaws.services.glue.ChoiceOption
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
import com.amazonaws.services.glue.ResolveSpec
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._

object GlueApp {
    def main(sysArgs: Array[String]) {
        val spark: SparkContext = new SparkContext()
        val glueContext: GlueContext = new GlueContext(spark)
        // @params: [JOB_NAME]
        val args = GlueArgParser.getResolvedOptions(sysArgs,
 Seq("JOB_NAME").toArray)
        Job.init(args("JOB_NAME"), glueContext, args.asJava)
        // @type: DataSource
        // @args: [database = "tempdb", table_name = "s3-source", transformation_ctx
 = "datasource0"]
        // @return: datasource0
        // @inputs: []
        val datasource0 = glueContext.getCatalogSource(database = "tempdb",
 tableName = "s3-source", redshiftTmpDir = "", transformationContext =
 "datasource0").getDynamicFrame()
        // @type: ApplyMapping
        // @args: [mapping = [("sensorid", "int", "sensorid", "int"),
 ("currenttemperature", "int", "currenttemperature", "int"), ("status", "string",
 "status", "string")], transformation_ctx = "applymapping1"]
        // @return: applymapping1
        // @inputs: [frame = datasource0]
        val applymapping1 = datasource0.applyMapping(mappings = Seq(("sensorid",
 "int", "sensorid", "int"), ("currenttemperature", "int", "currenttemperature",
 "int"), ("status", "string", "status", "string")), caseSensitive = false,
 transformationContext = "applymapping1")
```

```
        // @type: SelectFields
        // @args: [paths = ["sensorid", "currenttemperature", "status"],
 transformation_ctx = "selectfields2"]
        // @return: selectfields2
        // @inputs: [frame = applymapping1]
        val selectfields2 = applymapping1.selectFields(paths = Seq("sensorid",
 "currenttemperature", "status"), transformationContext = "selectfields2")
        // @type: ResolveChoice
        // @args: [choice = "MATCH_CATALOG", database = "tempdb", table_name = "my-
s3-sink", transformation_ctx = "resolvechoice3"]
        // @return: resolvechoice3
        // @inputs: [frame = selectfields2]
        val resolvechoice3 = selectfields2.resolveChoice(choiceOption =
 Some(ChoiceOption("MATCH_CATALOG")), database = Some("tempdb"), tableName =
 Some("my-s3-sink"), transformationContext = "resolvechoice3")
        // @type: DataSink
        // @args: [database = "tempdb", table_name = "my-s3-sink",
 transformation_ctx = "datasink4"]
        // @return: datasink4
        // @inputs: [frame = resolvechoice3]
        val datasink4 = glueContext.getCatalogSink(database = "tempdb",
 tableName = "my-s3-sink", redshiftTmpDir = "", transformationContext =
 "datasink4").writeDynamicFrame(resolvechoice3)
        Job.commit()
    }
}
```

輸出：

```
{
    "Name": "my-testing-job"
}
```

如需詳細資訊，請參閱《Glue AWS 開發人員指南》中的在 Glue 中編寫任務。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateJob。

create-table

以下程式碼範例顯示如何使用 create-table。

AWS CLI

範例 1：建立 Kinesis 資料串流的資料表

下列create-table範例會在 AWS Glue Data Catalog 中建立描述 Kinesis 資料串流的資料表。

```
aws glue create-table \
    --database-name tempdb \
    --table-input  '{"Name":"test-kinesis-input", "StorageDescriptor":{ \
            "Columns":[ \
                {"Name":"sensorid", "Type":"int"}, \
                {"Name":"currenttemperature", "Type":"int"}, \
                {"Name":"status", "Type":"string"} \
            ], \
            "Location":"my-testing-stream", \
            "Parameters":{ \
                "typeOfData":"kinesis","streamName":"my-testing-stream", \
                "kinesisUrl":"https://kinesis.us-east-1.amazonaws.com" \
            }, \
            "SerdeInfo":{ \
                "SerializationLibrary":"org.openx.data.jsonserde.JsonSerDe"} \
        }, \
        "Parameters":{ \
            "classification":"json"} \
        }' \
    --profile my-profile \
    --endpoint https://glue.us-east-1.amazonaws.com
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Glue AWS 開發人員指南》中的 Glue Data Catalog 中的定義資料表。AWS

範例 2：建立 Kafka 資料存放區的資料表

下列create-table範例會在 Glue Data Catalog AWS 中建立描述 Kafka 資料存放區的資料表。

```
aws glue create-table \
    --database-name tempdb \
    --table-input  '{"Name":"test-kafka-input", "StorageDescriptor":{ \
            "Columns":[ \
                {"Name":"sensorid", "Type":"int"}, \
```

```
                {"Name":"currenttemperature", "Type":"int"}, \
                {"Name":"status", "Type":"string"}
            ], \
            "Location":"glue-topic", \
            "Parameters":{ \
                "typeOfData":"kafka","topicName":"glue-topic", \
                "connectionName":"my-kafka-connection"
            }, \
            "SerdeInfo":{ \
                "SerializationLibrary":"org.apache.hadoop.hive.serde2.OpenCSVSerde"}
 \
        }, \
        "Parameters":{ \
            "separatorChar":","} \
        }' \
    --profile my-profile \
    --endpoint https://glue.us-east-1.amazonaws.com
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Glue AWS 開發人員指南》中的 Glue Data Catalog 中的定義資料表。AWS

範例 3：建立 a AWS S3 資料存放區的資料表

下列`create-table`範例會在 AWS Glue Data Catalog 中建立描述 AWS Simple Storage Service (AWS S3) 資料存放區的資料表。

```
aws glue create-table \
    --database-name tempdb \
    --table-input  '{"Name":"s3-output", "StorageDescriptor":{ \
            "Columns":[ \
                {"Name":"s1", "Type":"string"}, \
                {"Name":"s2", "Type":"int"}, \
                {"Name":"s3", "Type":"string"}
            ], \
            "Location":"s3://bucket-path/", \
            "SerdeInfo":{ \
                "SerializationLibrary":"org.openx.data.jsonserde.JsonSerDe"} \
        }, \
        "Parameters":{ \
            "classification":"json"} \
        }' \
```

```
    --profile my-profile \
    --endpoint https://glue.us-east-1.amazonaws.com
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Glue AWS 開發人員指南》中的 Glue Data Catalog 中的定義資料表。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateTable。

delete-job

以下程式碼範例顯示如何使用 delete-job。

AWS CLI

若要刪除工作

下列 delete-job 範例會刪除不再需要的工作。

```
aws glue delete-job \
    --job-name my-testing-job
```

輸出：

```
{
    "JobName": "my-testing-job"
}
```

如需詳細資訊，請參閱《Glue AWS 開發人員指南》中的在 Glue 主控台上使用任務。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteJob。

get-databases

以下程式碼範例顯示如何使用 get-databases。

AWS CLI

在 Glue Data Catalog AWS 中列出部分或全部資料庫的定義

下列 get-databases 範例會傳回 Data Catalog 中的資料庫相關資訊。

```
aws glue get-databases
```

輸出：

```
{
    "DatabaseList": [
        {
            "Name": "default",
            "Description": "Default Hive database",
            "LocationUri": "file:/spark-warehouse",
            "CreateTime": 1602084052.0,
            "CreateTableDefaultPermissions": [
                {
                    "Principal": {
                        "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
                    },
                    "Permissions": [
                        "ALL"
                    ]
                }
            ],
            "CatalogId": "111122223333"
        },
        {
            "Name": "flights-db",
            "CreateTime": 1587072847.0,
            "CreateTableDefaultPermissions": [
                {
                    "Principal": {
                        "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
                    },
                    "Permissions": [
                        "ALL"
                    ]
                }
            ],
            "CatalogId": "111122223333"
        },
        {
            "Name": "legislators",
            "CreateTime": 1601415625.0,
```

```
            "CreateTableDefaultPermissions": [
                {
                    "Principal": {
                        "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
                    },
                    "Permissions": [
                        "ALL"
                    ]
                }
            ],
            "CatalogId": "111122223333"
        },
        {
            "Name": "tempdb",
            "CreateTime": 1601498566.0,
            "CreateTableDefaultPermissions": [
                {
                    "Principal": {
                        "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
                    },
                    "Permissions": [
                        "ALL"
                    ]
                }
            ],
            "CatalogId": "111122223333"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的在 Data Catalog 中定義資料庫。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetDatabases。

get-job-run

以下程式碼範例顯示如何使用 get-job-run。

AWS CLI

取得工作執行的相關資訊

以下 get-job-run 範例會擷取工作執行的相關資訊。

```
aws glue get-job-run \
    --job-name "Combine legistators data" \
    --run-id jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e
```

輸出：

```
{
    "JobRun": {
        "Id": "jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e",
        "Attempt": 0,
        "JobName": "Combine legistators data",
        "StartedOn": 1602873931.255,
        "LastModifiedOn": 1602874075.985,
        "CompletedOn": 1602874075.985,
        "JobRunState": "SUCCEEDED",
        "Arguments": {
            "--enable-continuous-cloudwatch-log": "true",
            "--enable-metrics": "",
            "--enable-spark-ui": "true",
            "--job-bookmark-option": "job-bookmark-enable",
            "--spark-event-logs-path": "s3://aws-glue-assets-111122223333-us-east-1/
sparkHistoryLogs/"
        },
        "PredecessorRuns": [],
        "AllocatedCapacity": 10,
        "ExecutionTime": 117,
        "Timeout": 2880,
        "MaxCapacity": 10.0,
        "WorkerType": "G.1X",
        "NumberOfWorkers": 10,
        "LogGroupName": "/aws-glue/jobs",
        "GlueVersion": "2.0"
    }
}
```

如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的工作執行。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetJobRun。

get-job-runs

以下程式碼範例顯示如何使用 get-job-runs。

AWS CLI

取得工作的全部工作執行相關資訊

以下 get-job-runs 範例會擷取工作的工作執行相關資訊。

```
aws glue get-job-runs \
    --job-name "my-testing-job"
```

輸出：

```
{
    "JobRuns": [
        {
            "Id":
 "jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e",
            "Attempt": 0,
            "JobName": "my-testing-job",
            "StartedOn": 1602873931.255,
            "LastModifiedOn": 1602874075.985,
            "CompletedOn": 1602874075.985,
            "JobRunState": "SUCCEEDED",
            "Arguments": {
                "--enable-continuous-cloudwatch-log": "true",
                "--enable-metrics": "",
                "--enable-spark-ui": "true",
                "--job-bookmark-option": "job-bookmark-enable",
                "--spark-event-logs-path": "s3://aws-glue-assets-111122223333-us-
east-1/sparkHistoryLogs/"
            },
            "PredecessorRuns": [],
            "AllocatedCapacity": 10,
            "ExecutionTime": 117,
            "Timeout": 2880,
            "MaxCapacity": 10.0,
            "WorkerType": "G.1X",
            "NumberOfWorkers": 10,
            "LogGroupName": "/aws-glue/jobs",
            "GlueVersion": "2.0"
        },
        {
            "Id":
 "jr_03cc19ddab11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f_attempt_2",
```

```
            "Attempt": 2,
            "PreviousRunId":
 "jr_03cc19ddab11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f_attempt_1",
            "JobName": "my-testing-job",
            "StartedOn": 1602811168.496,
            "LastModifiedOn": 1602811282.39,
            "CompletedOn": 1602811282.39,
            "JobRunState": "FAILED",
            "ErrorMessage": "An error occurred while calling
 o122.pyWriteDynamicFrame.
                Access Denied (Service: Amazon S3; Status Code: 403; Error Code:
 AccessDenied;
                Request ID: 021AAB703DB20A2D;
                S3 Extended Request ID: teZk24Y09TkXzBvMPG502L5VJBhe9DJuWA9/
TXtuGOqfByajkfL/Tlqt5JBGdEGpigAqzdMDM/U=)",
            "PredecessorRuns": [],
            "AllocatedCapacity": 10,
            "ExecutionTime": 110,
            "Timeout": 2880,
            "MaxCapacity": 10.0,
            "WorkerType": "G.1X",
            "NumberOfWorkers": 10,
            "LogGroupName": "/aws-glue/jobs",
            "GlueVersion": "2.0"
        },
        {
            "Id":
 "jr_03cc19ddab11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f_attempt_1",
            "Attempt": 1,
            "PreviousRunId":
 "jr_03cc19ddab11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f",
            "JobName": "my-testing-job",
            "StartedOn": 1602811020.518,
            "LastModifiedOn": 1602811138.364,
            "CompletedOn": 1602811138.364,
            "JobRunState": "FAILED",
            "ErrorMessage": "An error occurred while calling
 o122.pyWriteDynamicFrame.
                 Access Denied (Service: Amazon S3; Status Code: 403; Error Code:
 AccessDenied;
                 Request ID: 2671D37856AE7ABB;
                 S3 Extended Request ID: RLJCJw20brV
+PpC6GpORahyF2fp9flB5SSb2bTGPnUSPVizLXRl1PN3QZldb+v1o9qRVktNYbW8=)",
            "PredecessorRuns": [],
```

```
            "AllocatedCapacity": 10,
            "ExecutionTime": 113,
            "Timeout": 2880,
            "MaxCapacity": 10.0,
            "WorkerType": "G.1X",
            "NumberOfWorkers": 10,
            "LogGroupName": "/aws-glue/jobs",
            "GlueVersion": "2.0"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的工作執行。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetJobRuns。

get-job

以下程式碼範例顯示如何使用 get-job。

AWS CLI

擷取工作相關資訊

以下 get-job 範例會擷取工作相關資訊。

```
aws glue get-job \
    --job-name my-testing-job
```

輸出：

```
{
    "Job": {
        "Name": "my-testing-job",
        "Role": "Glue_DefaultRole",
        "CreatedOn": 1602805698.167,
        "LastModifiedOn": 1602805698.167,
        "ExecutionProperty": {
            "MaxConcurrentRuns": 1
        },
        "Command": {
            "Name": "gluestreaming",
            "ScriptLocation": "s3://janetst-bucket-01/Scripts/test_script.scala",
```

```
            "PythonVersion": "2"
        },
        "DefaultArguments": {
            "--class": "GlueApp",
            "--job-language": "scala"
        },
        "MaxRetries": 0,
        "AllocatedCapacity": 10,
        "MaxCapacity": 10.0,
        "GlueVersion": "1.0"
    }
}
```

如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的工作。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetJob。

get-plan

以下程式碼範例顯示如何使用 get-plan。

AWS CLI

取得產生的程式碼，以將來源資料表中的資料映射至目標資料表

以下內容會get-plan擷取產生的程式碼，用於將資料欄從資料來源映射至資料目標。

```
aws glue get-plan --mapping '[ \
    { \
        "SourcePath":"sensorid", \
        "SourceTable":"anything", \
        "SourceType":"int", \
        "TargetPath":"sensorid", \
        "TargetTable":"anything", \
        "TargetType":"int" \
    }, \
    { \
        "SourcePath":"currenttemperature", \
        "SourceTable":"anything", \
        "SourceType":"int", \
        "TargetPath":"currenttemperature", \
        "TargetTable":"anything", \
        "TargetType":"int" \
```

```
    }, \
    { \
        "SourcePath":"status", \
        "SourceTable":"anything", \
        "SourceType":"string", \
        "TargetPath":"status", \
        "TargetTable":"anything", \
        "TargetType":"string" \
    }]' \
    --source '{ \
        "DatabaseName":"tempdb", \
        "TableName":"s3-source" \
    }' \
    --sinks '[ \
        { \
            "DatabaseName":"tempdb", \
            "TableName":"my-s3-sink" \
        }]' 
    --language "scala"
    --endpoint https://glue.us-east-1.amazonaws.com
    --output "text"
```

輸出：

```
import com.amazonaws.services.glue.ChoiceOption
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
import com.amazonaws.services.glue.ResolveSpec
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._

object GlueApp {
  def main(sysArgs: Array[String]) {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    // @params: [JOB_NAME]
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)
    // @type: DataSource
```

```
    // @args: [database = "tempdb", table_name = "s3-source", transformation_ctx =
 "datasource0"]
    // @return: datasource0
    // @inputs: []
    val datasource0 = glueContext.getCatalogSource(database = "tempdb",
 tableName = "s3-source", redshiftTmpDir = "", transformationContext =
 "datasource0").getDynamicFrame()
    // @type: ApplyMapping
    // @args: [mapping = [("sensorid", "int", "sensorid", "int"),
 ("currenttemperature", "int", "currenttemperature", "int"), ("status", "string",
 "status", "string")], transformation_ctx = "applymapping1"]
    // @return: applymapping1
    // @inputs: [frame = datasource0]
    val applymapping1 = datasource0.applyMapping(mappings = Seq(("sensorid",
 "int", "sensorid", "int"), ("currenttemperature", "int", "currenttemperature",
 "int"), ("status", "string", "status", "string")), caseSensitive = false,
 transformationContext = "applymapping1")
    // @type: SelectFields
    // @args: [paths = ["sensorid", "currenttemperature", "status"],
 transformation_ctx = "selectfields2"]
    // @return: selectfields2
    // @inputs: [frame = applymapping1]
    val selectfields2 = applymapping1.selectFields(paths = Seq("sensorid",
 "currenttemperature", "status"), transformationContext = "selectfields2")
    // @type: ResolveChoice
    // @args: [choice = "MATCH_CATALOG", database = "tempdb", table_name = "my-s3-
sink", transformation_ctx = "resolvechoice3"]
    // @return: resolvechoice3
    // @inputs: [frame = selectfields2]
    val resolvechoice3 = selectfields2.resolveChoice(choiceOption =
 Some(ChoiceOption("MATCH_CATALOG")), database = Some("tempdb"), tableName =
 Some("my-s3-sink"), transformationContext = "resolvechoice3")
    // @type: DataSink
    // @args: [database = "tempdb", table_name = "my-s3-sink", transformation_ctx =
 "datasink4"]
    // @return: datasink4
    // @inputs: [frame = resolvechoice3]
    val datasink4 = glueContext.getCatalogSink(database = "tempdb",
 tableName = "my-s3-sink", redshiftTmpDir = "", transformationContext =
 "datasink4").writeDynamicFrame(resolvechoice3)
    Job.commit()
  }
}
```

如需詳細資訊，請參閱《Glue AWS 開發人員指南》中的在 Glue 中編輯指令碼。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPlan。

get-tables

以下程式碼範例顯示如何使用 get-tables。

AWS CLI

列出指定資料庫中部分或全部資料表的定義

下列 get-tables 範例會傳回指定資料庫中資料表的相關資訊。

```
aws glue get-tables --database-name 'tempdb'
```

輸出：

```
{
    "TableList": [
        {
            "Name": "my-s3-sink",
            "DatabaseName": "tempdb",
            "CreateTime": 1602730539.0,
            "UpdateTime": 1602730539.0,
            "Retention": 0,
            "StorageDescriptor": {
                "Columns": [
                    {
                        "Name": "sensorid",
                        "Type": "int"
                    },
                    {
                        "Name": "currenttemperature",
                        "Type": "int"
                    },
                    {
                        "Name": "status",
                        "Type": "string"
                    }
                ],
                "Location": "s3://janetst-bucket-01/test-s3-output/",
                "Compressed": false,
```

```
                "NumberOfBuckets": 0,
                "SerdeInfo": {
                    "SerializationLibrary": "org.openx.data.jsonserde.JsonSerDe"
                },
                "SortColumns": [],
                "StoredAsSubDirectories": false
            },
            "Parameters": {
                "classification": "json"
            },
            "CreatedBy": "arn:aws:iam::007436865787:user/JRSTERN",
            "IsRegisteredWithLakeFormation": false,
            "CatalogId": "007436865787"
        },
        {
            "Name": "s3-source",
            "DatabaseName": "tempdb",
            "CreateTime": 1602730658.0,
            "UpdateTime": 1602730658.0,
            "Retention": 0,
            "StorageDescriptor": {
                "Columns": [
                    {
                        "Name": "sensorid",
                        "Type": "int"
                    },
                    {
                        "Name": "currenttemperature",
                        "Type": "int"
                    },
                    {
                        "Name": "status",
                        "Type": "string"
                    }
                ],
                "Location": "s3://janetst-bucket-01/",
                "Compressed": false,
                "NumberOfBuckets": 0,
                "SortColumns": [],
                "StoredAsSubDirectories": false
            },
            "Parameters": {
                "classification": "json"
            },
```

```
            "CreatedBy": "arn:aws:iam::007436865787:user/JRSTERN",
            "IsRegisteredWithLakeFormation": false,
            "CatalogId": "007436865787"
        },
        {
            "Name": "test-kinesis-input",
            "DatabaseName": "tempdb",
            "CreateTime": 1601507001.0,
            "UpdateTime": 1601507001.0,
            "Retention": 0,
            "StorageDescriptor": {
                "Columns": [
                    {
                        "Name": "sensorid",
                        "Type": "int"
                    },
                    {
                        "Name": "currenttemperature",
                        "Type": "int"
                    },
                    {
                        "Name": "status",
                        "Type": "string"
                    }
                ],
                "Location": "my-testing-stream",
                "Compressed": false,
                "NumberOfBuckets": 0,
                "SerdeInfo": {
                    "SerializationLibrary": "org.openx.data.jsonserde.JsonSerDe"
                },
                "SortColumns": [],
                "Parameters": {
                    "kinesisUrl": "https://kinesis.us-east-1.amazonaws.com",
                    "streamName": "my-testing-stream",
                    "typeOfData": "kinesis"
                },
                "StoredAsSubDirectories": false
            },
            "Parameters": {
                "classification": "json"
            },
            "CreatedBy": "arn:aws:iam::007436865787:user/JRSTERN",
            "IsRegisteredWithLakeFormation": false,
```

```
            "CatalogId": "007436865787"
        }
    ]
}
```

如需詳細資訊，請參閱《Glue AWS 開發人員指南》中的 Glue Data Catalog 中的定義資料表。AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetTables。

start-crawler

以下程式碼範例顯示如何使用 start-crawler。

AWS CLI

啟動爬蟲程式

以下 start-crawler 範例會啟動爬蟲程式。

```
aws glue start-crawler --name my-crawler
```

輸出：

```
None
```

如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的定義爬蟲程式。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 StartCrawler。

start-job-run

以下程式碼範例顯示如何使用 start-job-run。

AWS CLI

開始執行工作

以下 start-job-run 範例會啟動工作。

```
aws glue start-job-run \
    --job-name my-job
```

輸出：

```
{
    "JobRunId":
 "jr_22208b1f44eb5376a60569d4b21dd20fcb8621e1a366b4e7b2494af764b82ded"
}
```

如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的授權工作。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 StartJobRun。

使用 的 GuardDuty 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 GuardDuty 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`accept-invitation`

以下程式碼範例顯示如何使用 `accept-invitation`。

AWS CLI

若要接受成為目前區域中 GuardDuty 成員帳戶的邀請

下列`accept-invitation`範例顯示如何接受成為目前區域中 GuardDuty 成員帳戶的邀請。

```
aws guardduty accept-invitation  \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --master-id 123456789111 \
    --invitation-id d6b94fb03a66ff665f7db8764example
```

此命令不會產生輸出。

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的透過邀請管理 GuardDuty 帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptInvitation。

archive-findings

以下程式碼範例顯示如何使用 archive-findings。

AWS CLI

在目前區域中封存問題清單

此archive-findings範例示範如何在目前區域中封存問題清單。

```
aws guardduty archive-findings \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --finding-ids d6b94fb03a66ff665f7db8764example 3eb970e0de00c16ec14e6910fexample
```

此命令不會產生輸出。

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的建立禁止規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ArchiveFindings。

create-detector

以下程式碼範例顯示如何使用 create-detector。

AWS CLI

在目前區域中啟用 GuardDuty

此範例示範如何在目前區域中建立新的偵測器，以啟用 GuardDuty：

```
aws guardduty create-detector \
    --enable
```

輸出：

```
{
    "DetectorId": "b6b992d6d2f48e64bc59180bfexample"
```

```
}
```

如需詳細資訊，請參閱Amazon GuardDuty》中的啟用 Amazon GuardDuty。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDetector。

create-filter

以下程式碼範例顯示如何使用 create-filter。

AWS CLI

範例 1：在目前區域中建立新的篩選條件

下列create-filter範例會建立篩選條件，以符合從特定映像建立之執行個體的所有 Portscan 調查結果。這不會隱藏這些調查結果。

```
aws guardduty create-filter \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --name myFilterExample \
    --finding-criteria '{"Criterion": {"type": {"Eq": ["Recon:EC2/
Portscan"]},"resource.instanceDetails.imageId": {"Eq": ["ami-0a7a207083example"]}}}'
```

輸出：

```
{
    "Name": "myFilterExample"
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的篩選 GuardDuty 調查結果。 GuardDuty

範例 2：在目前區域中建立新的篩選條件和隱藏問題清單

下列create-filter範例會建立篩選條件，以符合從特定映像建立之執行個體的所有 Portscan 調查結果。此篩選條件會封存這些調查結果，使其不會出現在您目前的調查結果中。

```
aws guardduty create-filter \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --action ARCHIVE \
    --name myFilterSecondExample \
    --finding-criteria '{"Criterion": {"type": {"Eq": ["Recon:EC2/
Portscan"]},"resource.instanceDetails.imageId": {"Eq": ["ami-0a7a207083example"]}}}'
```

輸出：

```
{
    "Name": "myFilterSecondExample"
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的篩選 GuardDuty 調查結果。 GuardDuty

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFilter。

create-ip-set

以下程式碼範例顯示如何使用 create-ip-set。

AWS CLI

建立和啟用信任的 IP 組

下列create-ip-set範例會建立並啟用目前區域中的信任 IP 組。

```
aws guardduty create-ip-set \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --name new-ip-set-example \
    --format TXT \
    --location s3://amzn-s3-demo-bucket/customtrustlist.csv \
    --activate
```

輸出：

```
{
    "IpSetId": "d4b94fc952d6912b8f3060768example"
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的使用信任 IP 清單和威脅清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateIpSet。

create-members

以下程式碼範例顯示如何使用 create-members。

AWS CLI

在目前區域中將新成員與您的 GuardDuty 主帳戶建立關聯。

此範例說明如何將要由目前帳戶管理的成員帳戶關聯為 GuardDuty 主帳戶。

```
aws guardduty create-members
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --account-details AccountId=111122223333,Email=first
+member@example.com AccountId=111111111111 ,Email=another+member@example.com
```

輸出：

```
{
   "UnprocessedAccounts": []
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的管理多個帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateMembers。

create-publishing-destination

以下程式碼範例顯示如何使用 create-publishing-destination。

AWS CLI

建立發佈目的地，以將目前區域中的 GuardDuty 問題清單匯出至其中。

下列create-publishing-destination範例示範如何設定發佈目的地以匯出目前 （未封存） GuardDuty 調查結果，以追蹤歷史調查結果資料。

```
aws guardduty create-publishing-destination \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --destination-type S3 \
    --destination-properties 'DestinationArn=arn:aws:s3:::amzn-s3-demo-
bucket,KmsKeyArn=arn:aws:kms:us-west-1:111122223333:key/84cee9c5-dea1-401a-ab6d-
e1de7example'
```

輸出：

```
{
```

```
    "DestinationId": "46b99823849e1bbc242dfbe3cexample"
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的將產生的 GuardDuty 調查結果匯出至 Amazon S3 儲存貯體。 GuardDuty

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePublishingDestination。

create-sample-findings

以下程式碼範例顯示如何使用 create-sample-findings。

AWS CLI

在目前區域中建立範例 GuardDuty 調查結果。

此範例示範如何建立所提供類型的範例調查結果。

```
aws guardduty create-sample-findings \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --finding-types UnauthorizedAccess:EC2/TorClient UnauthorizedAccess:EC2/TorRelay
```

此命令不會產生輸出。

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的問題清單範例。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSampleFindings。

create-threat-intel-set

以下程式碼範例顯示如何使用 create-threat-intel-set。

AWS CLI

建立和啟用新的威脅 intel 集

下列create-threat-intel-set範例會建立並啟用目前區域中的威脅 intel 集。

```
aws guardduty create-threat-intel-set \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --name myThreatSet-example \
    --format TXT \
    --location s3://amzn-s3-demo-bucket/threatlist.csv \
```

```
    --activate
```

輸出：

```
{
    "ThreatIntelSetId": "20b9a4691aeb33506b808878cexample"
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的使用信任 IP 清單和威脅清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateThreatIntelSet。

`decline-invitations`

以下程式碼範例顯示如何使用 `decline-invitations`。

AWS CLI

拒絕目前區域中另一個帳戶管理 Guardduty 的邀請。

此範例示範如何拒絕成員資格邀請。

```
aws guardduty decline-invitations \
    --account-ids 111122223333
```

輸出：

```
{
    "UnprocessedAccounts": []
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的透過邀請管理 GuardDuty 帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeclineInvitations。

`delete-detector`

以下程式碼範例顯示如何使用 `delete-detector`。

AWS CLI

在目前區域中刪除偵測器並停用 GuardDuty。

此範例顯示如何刪除偵測器，如果成功，這將在與該偵測器相關聯的區域中停用 GuardDuty。

```
aws guardduty delete-detector \
    --detector-id b6b992d6d2f48e64bc59180bfexample
```

此命令不會產生輸出。

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的暫停或停用 GuardDuty。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDetector。

delete-filter

以下程式碼範例顯示如何使用 delete-filter。

AWS CLI

刪除目前區域中的現有篩選條件

此範例說明如何建立刪除篩選條件。

```
aws guardduty delete-filter \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --filter-name byebyeFilter
```

此命令不會產生輸出。

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的篩選問題清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFilter。

disable-organization-admin-account

以下程式碼範例顯示如何使用 disable-organization-admin-account。

AWS CLI

移除 帳戶做為組織中 GuardDuty 的委派管理員

此範例說明如何移除 帳戶做為 GuardDuty 的委派管理員。

```
aws guardduty disable-organization-admin-account \
```

```
    --admin-account-id 111122223333
```

此命令不會產生輸出。

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的管理 AWS 組織帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableOrganizationAdminAccount。

disassociate-from-master-account

以下程式碼範例顯示如何使用 disassociate-from-master-account。

AWS CLI

在目前區域中取消與目前管理員帳戶的關聯

下列disassociate-from-master-account範例會將您的帳戶與目前 AWS 區域中目前的 GuardDuty 管理員帳戶取消關聯。

```
aws guardduty disassociate-from-master-account \
    --detector-id d4b040365221be2b54a6264dcexample
```

此命令不會產生輸出。

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的了解 GuardDuty 管理員帳戶與成員帳戶之間的關係。 GuardDuty

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateFromMasterAccount。

get-detector

以下程式碼範例顯示如何使用 get-detector。

AWS CLI

擷取特定偵測器的詳細資訊

下列get-detector範例顯示指定偵測器的組態詳細資訊。

```
aws guardduty get-detector \
    --detector-id 12abc34d567e8fa901bc2d34eexample
```

輸出：

```
{
    "Status": "ENABLED",
    "ServiceRole": "arn:aws:iam::111122223333:role/aws-service-role/
guardduty.amazonaws.com/AWSServiceRoleForAmazonGuardDuty",
    "Tags": {},
    "FindingPublishingFrequency": "SIX_HOURS",
    "UpdatedAt": "2018-11-07T03:24:22.938Z",
    "CreatedAt": "2017-12-22T22:51:31.940Z"
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的概念和術語。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDetector。

get-findings

以下程式碼範例顯示如何使用 get-findings。

AWS CLI

範例 1：擷取特定問題清單的詳細資訊

下列get-findings範例會擷取指定問題清單的完整 JSON 問題清單詳細資訊。

```
aws guardduty get-findings \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --finding-id 1ab92989eaf0e742df4a014d5example
```

輸出：

```
{
    "Findings": [
        {
            "Resource": {
                "ResourceType": "AccessKey",
                "AccessKeyDetails": {
                    "UserName": "testuser",
                    "UserType": "IAMUser",
                    "PrincipalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "AccessKeyId": "ASIASZ4SI7REEEXAMPLE"
                }
            },
```

```
            "Description": "APIs commonly used to discover the users, groups,
 policies and permissions in an account, was invoked by IAM principal testuser under
 unusual circumstances. Such activity is not typically seen from this principal.",
            "Service": {
                "Count": 5,
                "Archived": false,
                "ServiceName": "guardduty",
                "EventFirstSeen": "2020-05-26T22:02:24Z",
                "ResourceRole": "TARGET",
                "EventLastSeen": "2020-05-26T22:33:55Z",
                "DetectorId": "d4b040365221be2b54a6264dcexample",
                "Action": {
                    "ActionType": "AWS_API_CALL",
                    "AwsApiCallAction": {
                        "RemoteIpDetails": {
                            "GeoLocation": {
                                "Lat": 51.5164,
                                "Lon": -0.093
                            },
                            "City": {
                                "CityName": "London"
                            },
                            "IpAddressV4": "52.94.36.7",
                            "Organization": {
                                "Org": "Amazon.com",
                                "Isp": "Amazon.com",
                                "Asn": "16509",
                                "AsnOrg": "AMAZON-02"
                            },
                            "Country": {
                                "CountryName": "United Kingdom"
                            }
                        },
                        "Api": "ListPolicyVersions",
                        "ServiceName": "iam.amazonaws.com",
                        "CallerType": "Remote IP"
                    }
                }
            },
            "Title": "Unusual user permission reconnaissance activity by testuser.",
            "Type": "Recon:IAMUser/UserPermissions",
            "Region": "us-east-1",
            "Partition": "aws",
```

```
            "Arn": "arn:aws:guardduty:us-east-1:111122223333:detector/
d4b040365221be2b54a6264dcexample/finding/1ab92989eaf0e742df4a014d5example",
            "UpdatedAt": "2020-05-26T22:55:21.703Z",
            "SchemaVersion": "2.0",
            "Severity": 5,
            "Id": "1ab92989eaf0e742df4a014d5example",
            "CreatedAt": "2020-05-26T22:21:48.385Z",
            "AccountId": "111122223333"
        }
    ]
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的調查結果。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFindings。

get-ip-set

以下程式碼範例顯示如何使用 get-ip-set。

AWS CLI

列出取得指定信任 IP 集的詳細資訊

下列get-ip-set範例顯示指定信任 IP 集的狀態和詳細資訊。

```
aws guardduty get-ip-set \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --ip-set-id d4b94fc952d6912b8f3060768example
```

輸出：

```
{
    "Status": "ACTIVE",
    "Location": "s3://amzn-s3-demo-bucket.s3-us-west-2.amazonaws.com/
customlist.csv",
    "Tags": {},
    "Format": "TXT",
    "Name": "test-ip-set-example"
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的使用信任 IP 清單和威脅清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIpSet。

`get-master-account`

以下程式碼範例顯示如何使用 `get-master-account`。

AWS CLI

擷取目前區域中主帳戶的詳細資訊

下列`get-master-account`範例顯示目前區域中與偵測器相關聯的主帳戶狀態和詳細資訊。

```
aws guardduty get-master-account \
    --detector-id 12abc34d567e8fa901bc2d34eexample
```

輸出：

```
{
    "Master": {
        "InvitationId": "04b94d9704854a73f94e061e8example",
        "InvitedAt": "2020-06-09T22:23:04.970Z",
        "RelationshipStatus": "Enabled",
        "AccountId": "111122223333"
    }
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的了解 GuardDuty 管理員帳戶與成員帳戶之間的關係。 GuardDuty

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMasterAccount。

`list-detectors`

以下程式碼範例顯示如何使用 `list-detectors`。

AWS CLI

列出目前區域中可用的偵測器

下列`list-detectors`範例列出您目前 AWS 區域中可用的偵測器。

```
aws guardduty list-detectors
```

輸出：

```
{
    "DetectorIds": [
        "12abc34d567e8fa901bc2d34eexample"
    ]
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的概念和術語。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDetectors。

`list-findings`

以下程式碼範例顯示如何使用 `list-findings`。

AWS CLI

範例 1：列出目前區域的所有問題清單

下列`list-findings`範例顯示目前區域的所有 findingIds清單，依嚴重性從最高到最低排序。

```
aws guardduty list-findings \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --sort-criteria '{"AttributeName": "severity","OrderBy":"DESC"}'
```

輸出：

```
{
    "FindingIds": [
        "04b8ab50fd29c64fc771b232dexample",
        "5ab8ab50fd21373735c826d3aexample",
        "90b93de7aba69107f05bbe60bexample",
        ...
    ]
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的調查結果。

範例 2：列出符合特定調查結果條件之目前區域的調查結果

下列`list-findings`範例顯示符合指定findingIds的清單。

```
aws guardduty list-findings \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --finding-criteria  '{"Criterion":{"type": {"Eq":["UnauthorizedAccess:EC2/
SSHBruteForce"]}}}'
```

輸出：

```
{
    "FindingIds": [
        "90b93de7aba69107f05bbe60bexample",
        "6eb9430d7023d30774d6f05e3example",
        "2eb91a2d060ac9a21963a5848example",
        "44b8ab50fd2b0039a9e48f570example",
        "9eb8ab4cd2b7e5b66ba4f5e96example",
        "e0b8ab3a38e9b0312cc390ceeexample"
    ]
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的調查結果。

範例 3：列出符合 JSON 檔案中定義之一組特定調查結果條件的目前區域調查結果

以下list-findings範例顯示所有未封存的 findingIds清單，並涉及名為 "testuser" 的 IAM 使用者，如 JSON 檔案中所指定。

```
aws guardduty list-findings \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --finding-criteria  file://myfile.json
```

myfile.json 的內容：

```
{"Criterion": {
    "resource.accessKeyDetails.userName":{
                "Eq":[
                    "testuser"
                    ]
                },
    "service.archived": {
                "Eq": [
                    "false"
                ]
            }
```

```
        }
}
```

輸出：

```
{
    "FindingIds": [
        "1ab92989eaf0e742df4a014d5example"
    ]
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的調查結果。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFindings。

`list-invitations`

以下程式碼範例顯示如何使用 `list-invitations`。

AWS CLI

列出您邀請成為目前區域中成員帳戶的詳細資訊

下列`list-invitations`範例會列出邀請成為目前區域中 GuardDuty 成員帳戶的詳細資訊和狀態。

```
aws guardduty list-invitations
```

輸出：

```
{
    "Invitations": [
        {
            "InvitationId": "d6b94fb03a66ff665f7db8764example",
            "InvitedAt": "2020-06-10T17:56:38.221Z",
            "RelationshipStatus": "Invited",
            "AccountId": "123456789111"
        }
    ]
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的透過邀請管理 GuardDuty 帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListInvitations。

list-ip-sets

以下程式碼範例顯示如何使用 list-ip-sets。

AWS CLI

列出目前區域中的信任 IP 集

下列list-ip-sets範例列出您目前區域中的信任 IP 集 AWS 。

```
aws guardduty list-ip-sets \
    --detector-id 12abc34d567e8fa901bc2d34eexample
```

輸出：

```
{
    "IpSetIds": [
        "d4b94fc952d6912b8f3060768example"
    ]
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的使用信任 IP 清單和威脅清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListIpSets。

list-members

以下程式碼範例顯示如何使用 list-members。

AWS CLI

範例 1：僅列出目前區域中的目前成員

下列list-members範例只會列出目前區域中與 GuardDuty 管理員帳戶相關聯的目前成員帳戶的詳細資訊。

```
aws guardduty list-members \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --only-associated="true"
```

輸出：

```
{
    "Members": [
        {
            "RelationshipStatus": "Enabled",
            "InvitedAt": "2020-06-09T22:49:00.910Z",
            "MasterId": "111122223333",
            "DetectorId": "7ab8b2f61b256c87f793f6a86example",
            "UpdatedAt": "2020-06-09T23:08:22.512Z",
            "Email": "your+member@example.com",
            "AccountId": "123456789012"
        }
    ]
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的了解 GuardDuty 管理員帳戶與成員帳戶之間的關係。 GuardDuty

範例 2：列出目前區域中的所有成員

下列list-members範例列出並提供所有成員帳戶的詳細資訊，包括在目前區域中已取消關聯或尚未接受來自 GuardDuty 管理員的邀請。

```
aws guardduty list-members \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --only-associated="false"
```

輸出：

```
{
    "Members": [
        {
            "RelationshipStatus": "Enabled",
            "InvitedAt": "2020-06-09T22:49:00.910Z",
            "MasterId": "111122223333",
            "DetectorId": "7ab8b2f61b256c87f793f6a86example",
            "UpdatedAt": "2020-06-09T23:08:22.512Z",
            "Email": "your+other+member@example.com",
            "AccountId": "555555555555"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的了解 GuardDuty 管理員帳戶與成員帳戶之間的關係。 GuardDuty

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMembers。

update-ip-set

以下程式碼範例顯示如何使用 update-ip-set。

AWS CLI

更新信任的 IP 集

下列update-ip-set範例示範如何更新信任 IP 集的詳細資訊。

```
aws guardduty update-ip-set \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --ip-set-id d4b94fc952d6912b8f3060768example \
    --location https://amzn-s3-demo-bucket.s3-us-west-2.amazonaws.com/
customtrustlist2.csv
```

此命令不會產生輸出。

如需詳細資訊，請參閱《GuardDuty 使用者指南》中的使用信任 IP 清單和威脅清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateIpSet。

AWS Health 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS Health。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

describe-affected-entities

以下程式碼範例顯示如何使用 describe-affected-entities。

AWS CLI

列出受指定 AWS 運作狀態事件影響的實體

下列describe-affected-entities範例列出受指定 AWS 運作狀態事件影響的實體。此事件是 AWS 帳戶的帳單通知。

```
aws health describe-affected-entities \
    --filter "eventArns=arn:aws:health:global::event/BILLING/
AWS_BILLING_NOTIFICATION/AWS_BILLING_NOTIFICATION_6ce1d874-e995-40e2-99cd-
EXAMPLE11145" \
    --region us-east-1
```

輸出：

```
{
    "entities": [
        {
            "entityArn": "arn:aws:health:global:123456789012:entity/
EXAMPLEimSMoULmWHpb",
            "eventArn": "arn:aws:health:global::event/BILLING/
AWS_BILLING_NOTIFICATION/AWS_BILLING_NOTIFICATION_6ce1d874-e995-40e2-99cd-
EXAMPLE11145",
            "entityValue": "AWS_ACCOUNT",
            "awsAccountId": "123456789012",
            "lastUpdatedTime": 1588356454.08
        }
    ]
}
```

如需詳細資訊，請參閱 AWS 運作狀態使用者指南中的事件日誌。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAffectedEntities。

describe-event-details

以下程式碼範例顯示如何使用 describe-event-details。

AWS CLI

列出 AWS 運作狀態事件的相關資訊

下列describe-event-details範例列出指定 AWS 運作狀態事件的相關資訊。

```
aws health describe-event-details \
    --event-arns "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/
AWS_EC2_OPERATIONAL_ISSUE_VKTXI_EXAMPLE111" \
    --region us-east-1
```

輸出：

```
{
    "successfulSet": [
        {
            "event": {
                "arn": "arn:aws:health:us-east-1::event/EC2/
AWS_EC2_OPERATIONAL_ISSUE/AWS_EC2_OPERATIONAL_ISSUE_VKTXI_EXAMPLE111",
                "service": "EC2",
                "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
                "eventTypeCategory": "issue",
                "region": "us-east-1",
                "startTime": 1587462325.096,
                "endTime": 1587464204.774,
                "lastUpdatedTime": 1587464204.865,
                "statusCode": "closed"
            },
            "eventDescription": {
                "latestDescription": "[RESOLVED] Increased API Error Rates and
 Latencies\n\n[02:45 AM PDT] We are investigating increased API error rates and
 latencies in the US-EAST-1 Region.\n\n[03:16 AM PDT] Between 2:10 AM and 2:59 AM
 PDT we experienced increased API error rates and latencies in the US-EAST-1 Region.
 The issue has been resolved and the service is operating normally."
            }
        }
    ],
    "failedSet": []
}
```

如需詳細資訊，請參閱《 AWS 運作狀態使用者指南》中的事件詳細資訊窗格。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEventDetails。

describe-events

以下程式碼範例顯示如何使用 describe-events。

AWS CLI

範例 1：列出 AWS 運作狀態事件

下列describe-events範例列出最近的 AWS 運作狀態事件。

```
aws health describe-events \
    --region us-east-1
```

輸出：

```
{
    "events": [
        {
            "arn": "arn:aws:health:us-west-1::event/ECS/AWS_ECS_OPERATIONAL_ISSUE/
AWS_ECS_OPERATIONAL_ISSUE_KWQPY_EXAMPLE111",
            "service": "ECS",
            "eventTypeCode": "AWS_ECS_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "us-west-1",
            "startTime": 1589077890.53,
            "endTime": 1589086345.597,
            "lastUpdatedTime": 1589086345.905,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:global::event/BILLING/AWS_BILLING_NOTIFICATION/
AWS_BILLING_NOTIFICATION_6ce1d874-e995-40e2-99cd-EXAMPLE1118b",
            "service": "BILLING",
            "eventTypeCode": "AWS_BILLING_NOTIFICATION",
            "eventTypeCategory": "accountNotification",
            "region": "global",
            "startTime": 1588356000.0,
            "lastUpdatedTime": 1588356524.358,
            "statusCode": "open",
            "eventScopeCode": "ACCOUNT_SPECIFIC"
        },
        {
```

```
            "arn": "arn:aws:health:us-west-2::event/
CLOUDFORMATION/AWS_CLOUDFORMATION_OPERATIONAL_ISSUE/
AWS_CLOUDFORMATION_OPERATIONAL_ISSUE_OHTWY_EXAMPLE111",
            "service": "CLOUDFORMATION",
            "eventTypeCode": "AWS_CLOUDFORMATION_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "us-west-2",
            "startTime": 1588279630.761,
            "endTime": 1588284650.0,
            "lastUpdatedTime": 1588284691.941,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:ap-northeast-1::event/LAMBDA/
AWS_LAMBDA_OPERATIONAL_ISSUE/AWS_LAMBDA_OPERATIONAL_ISSUE_JZDND_EXAMPLE111",
            "service": "LAMBDA",
            "eventTypeCode": "AWS_LAMBDA_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "ap-northeast-1",
            "startTime": 1587379534.08,
            "endTime": 1587391771.0,
            "lastUpdatedTime": 1587395689.316,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/
AWS_EC2_OPERATIONAL_ISSUE_COBXJ_EXAMPLE111",
            "service": "EC2",
            "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "us-east-1",
            "startTime": 1586473044.284,
            "endTime": 1586479706.091,
            "lastUpdatedTime": 1586479706.153,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:global::event/SECURITY/AWS_SECURITY_NOTIFICATION/
AWS_SECURITY_NOTIFICATION_42007387-8129-42da-8c88-EXAMPLE11139",
            "service": "SECURITY",
            "eventTypeCode": "AWS_SECURITY_NOTIFICATION",
```

```
            "eventTypeCategory": "accountNotification",
            "region": "global",
            "startTime": 1585674000.0,
            "lastUpdatedTime": 1585674004.132,
            "statusCode": "open",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:global::event/CLOUDFRONT/
AWS_CLOUDFRONT_OPERATIONAL_ISSUE/AWS_CLOUDFRONT_OPERATIONAL_ISSUE_FRQXG_EXAMPLE111",
            "service": "CLOUDFRONT",
            "eventTypeCode": "AWS_CLOUDFRONT_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "global",
            "startTime": 1585610898.589,
            "endTime": 1585617671.0,
            "lastUpdatedTime": 1585620638.869,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:us-east-1::event/SES/AWS_SES_OPERATIONAL_ISSUE/
AWS_SES_OPERATIONAL_ISSUE_URNDF_EXAMPLE111",
            "service": "SES",
            "eventTypeCode": "AWS_SES_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "us-east-1",
            "startTime": 1585342008.46,
            "endTime": 1585344017.0,
            "lastUpdatedTime": 1585344355.989,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:global::event/IAM/
AWS_IAM_OPERATIONAL_NOTIFICATION/
AWS_IAM_OPERATIONAL_NOTIFICATION_b6771c34-6ecd-4aea-9d3e-EXAMPLE1117e",
            "service": "IAM",
            "eventTypeCode": "AWS_IAM_OPERATIONAL_NOTIFICATION",
            "eventTypeCategory": "accountNotification",
            "region": "global",
            "startTime": 1584978300.0,
            "lastUpdatedTime": 1584978553.572,
            "statusCode": "open",
```

```
            "eventScopeCode": "ACCOUNT_SPECIFIC"
        },
        {
            "arn": "arn:aws:health:ap-southeast-2::event/EC2/
AWS_EC2_OPERATIONAL_ISSUE/AWS_EC2_OPERATIONAL_ISSUE_HNGHE_EXAMPLE111",
            "service": "EC2",
            "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "ap-southeast-2",
            "startTime": 1583881487.483,
            "endTime": 1583885056.785,
            "lastUpdatedTime": 1583885057.052,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        }
    ]
}
```

如需詳細資訊，請參閱運作狀態使用者指南中的 AWS 個人運作狀態儀表板入門。 AWS

範例 2：依服務和事件狀態碼列出 AWS 運作狀態事件

下列`describe-events`範例列出事件狀態為關閉的 Amazon Elastic Compute Cloud (Amazon EC2) AWS 運作狀態事件。

```
aws health describe-events \
    --filter "services=EC2,eventStatusCodes=closed"
```

輸出：

```
{
    "events": [
        {
            "arn": "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/
AWS_EC2_OPERATIONAL_ISSUE_VKTXI_EXAMPLE111",
            "service": "EC2",
            "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "us-east-1",
            "startTime": 1587462325.096,
            "endTime": 1587464204.774,
            "lastUpdatedTime": 1587464204.865,
```

```
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/
AWS_EC2_OPERATIONAL_ISSUE_COBXJ_EXAMPLE111",
            "service": "EC2",
            "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "us-east-1",
            "startTime": 1586473044.284,
            "endTime": 1586479706.091,
            "lastUpdatedTime": 1586479706.153,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:ap-southeast-2::event/EC2/
AWS_EC2_OPERATIONAL_ISSUE/AWS_EC2_OPERATIONAL_ISSUE_HNGHE_EXAMPLE111",
            "service": "EC2",
            "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "ap-southeast-2",
            "startTime": 1583881487.483,
            "endTime": 1583885056.785,
            "lastUpdatedTime": 1583885057.052,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        }
    ]
}
```

如需詳細資訊，請參閱運作狀態使用者指南中的 AWS 個人運作狀態儀表板入門。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEvents。

HealthImaging 範例使用 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 HealthImaging 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`copy-image-set`

以下程式碼範例顯示如何使用 copy-image-set。

AWS CLI

範例 1：複製沒有目的地的影像集。

下列copy-image-set範例會複製沒有目的地的影像集複本。

```
aws medical-imaging copy-image-set \
    --datastore-id 12345678901234567890123456789012 \
    --source-image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --copy-image-set-information '{"sourceImageSet": {"latestVersionId": "1" } }'
```

輸出：

```
{
    "destinationImageSetProperties": {
        "latestVersionId": "2",
        "imageSetWorkflowStatus": "COPYING",
        "updatedAt": 1680042357.432,
        "imageSetId": "b9a06fef182a5f992842f77f8e0868e5",
        "imageSetState": "LOCKED",
        "createdAt": 1680042357.432
    },
    "sourceImageSetProperties": {
        "latestVersionId": "1",
        "imageSetWorkflowStatus": "COPYING_WITH_READ_ONLY_ACCESS",
        "updatedAt": 1680042357.432,
        "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
        "imageSetState": "LOCKED",
        "createdAt": 1680027126.436
    },
    "datastoreId": "12345678901234567890123456789012"
```

```
}
```

範例 2：複製具有目的地的映像集。

下列copy-image-set範例會複製具有目的地的映像集。

```
aws medical-imaging copy-image-set \
    --datastore-id 12345678901234567890123456789012 \
    --source-image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --copy-image-set-information '{"sourceImageSet": {"latestVersionId": "1" },
 "destinationImageSet": { "imageSetId": "b9a06fef182a5f992842f77f8e0868e5",
 "latestVersionId": "1"} }'
```

輸出：

```
{
    "destinationImageSetProperties": {
        "latestVersionId": "2",
        "imageSetWorkflowStatus": "COPYING",
        "updatedAt": 1680042505.135,
        "imageSetId": "b9a06fef182a5f992842f77f8e0868e5",
        "imageSetState": "LOCKED",
        "createdAt": 1680042357.432
    },
    "sourceImageSetProperties": {
        "latestVersionId": "1",
        "imageSetWorkflowStatus": "COPYING_WITH_READ_ONLY_ACCESS",
        "updatedAt": 1680042505.135,
        "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
        "imageSetState": "LOCKED",
        "createdAt": 1680027126.436
    },
    "datastoreId": "12345678901234567890123456789012"
}
```

範例 3：將來源映像集的執行個體子集複製到目的地映像集。

下列copy-image-set範例會將來源映像集的一個 DICOM 執行個體複製到目的地映像集。提供強制參數來覆寫病患、檢查和序列層級屬性中的不一致。

```
aws medical-imaging copy-image-set \
    --datastore-id 12345678901234567890123456789012 \
```

```
    --source-image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --copy-image-set-information '{"sourceImageSet": {"latestVersionId":
 "1","DICOMCopies": {"copiableAttributes": "{\"SchemaVersion\":\"1.1\",\"Study\":
{\"Series\":{\"1.3.6.1.4.1.5962.99.1.3673257865.2104868982.1369432891697.3666.0\":
{\"Instances\":
{\"1.3.6.1.4.1.5962.99.1.3673257865.2104868982.1369432891697.3669.0\":
{}}}}}"}},"destinationImageSet": {"imageSetId":
 "b9eb50d8ee682eb9fcf4acbf92f62bb7","latestVersionId": "1"}}' \
    --force
```

輸出：

```
{
    "destinationImageSetProperties": {
        "latestVersionId": "2",
        "imageSetWorkflowStatus": "COPYING",
        "updatedAt": 1680042505.135,
        "imageSetId": "b9eb50d8ee682eb9fcf4acbf92f62bb7",
        "imageSetState": "LOCKED",
        "createdAt": 1680042357.432
    },
    "sourceImageSetProperties": {
        "latestVersionId": "1",
        "imageSetWorkflowStatus": "COPYING_WITH_READ_ONLY_ACCESS",
        "updatedAt": 1680042505.135,
        "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
        "imageSetState": "LOCKED",
        "createdAt": 1680027126.436
    },
    "datastoreId": "12345678901234567890123456789012"
}
```

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的複製映像集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CopyImageSet。

create-datastore

以下程式碼範例顯示如何使用 create-datastore。

AWS CLI

建立資料存放區

下列create-datastore程式碼範例會建立名為 的資料存放區my-datastore。

```
aws medical-imaging create-datastore \
    --datastore-name "my-datastore"
```

輸出：

```
{
    "datastoreId": "12345678901234567890123456789012",
    "datastoreStatus": "CREATING"
}
```

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的建立資料存放區。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateDatastore。

delete-datastore

以下程式碼範例顯示如何使用 delete-datastore。

AWS CLI

刪除資料存放區

下列delete-datastore程式碼範例會刪除資料存放區。

```
aws medical-imaging delete-datastore \
    --datastore-id "12345678901234567890123456789012"
```

輸出：

```
{
    "datastoreId": "12345678901234567890123456789012",
    "datastoreStatus": "DELETING"
}
```

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的刪除資料存放區。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteDatastore。

delete-image-set

以下程式碼範例顯示如何使用 delete-image-set。

AWS CLI

刪除映像集

下列delete-image-set程式碼範例會刪除映像集。

```
aws medical-imaging delete-image-set \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e
```

輸出：

```
{
    "imageSetWorkflowStatus": "DELETING",
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "datastoreId": "12345678901234567890123456789012"
}
```

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的刪除映像集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteImageSet。

get-datastore

以下程式碼範例顯示如何使用 get-datastore。

AWS CLI

取得資料存放區的屬性

下列get-datastore程式碼範例會取得資料存放區的屬性。

```
aws medical-imaging get-datastore \
    --datastore-id 12345678901234567890123456789012
```

輸出：

```
{
    "datastoreProperties": {
        "datastoreId": "12345678901234567890123456789012",
        "datastoreName": "TestDatastore123",
        "datastoreStatus": "ACTIVE",
        "datastoreArn": "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012",
        "createdAt": "2022-11-15T23:33:09.643000+00:00",
        "updatedAt": "2022-11-15T23:33:09.643000+00:00"
    }
}
```

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的取得資料存放區屬性。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetDatastore。

get-dicom-import-job

以下程式碼範例顯示如何使用 get-dicom-import-job。

AWS CLI

若要取得 dicom 匯入任務的屬性

下列get-dicom-import-job程式碼範例會取得 dicom 匯入任務的屬性。

```
aws medical-imaging get-dicom-import-job \
    --datastore-id "12345678901234567890123456789012" \
    --job-id "09876543210987654321098765432109"
```

輸出：

```
{
    "jobProperties": {
        "jobId": "09876543210987654321098765432109",
        "jobName": "my-job",
        "jobStatus": "COMPLETED",
        "datastoreId": "12345678901234567890123456789012",
        "dataAccessRoleArn": "arn:aws:iam::123456789012:role/
ImportJobDataAccessRole",
        "endedAt": "2022-08-12T11:29:42.285000+00:00",
        "submittedAt": "2022-08-12T11:28:11.152000+00:00",
```

```
        "inputS3Uri": "s3://medical-imaging-dicom-input/dicom_input/",
        "outputS3Uri": "s3://medical-imaging-output/
job_output/12345678901234567890123456789012-
DicomImport-09876543210987654321098765432109/"
    }
}
```

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的取得匯入任務屬性。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDICOMImportJob。

get-image-frame

以下程式碼範例顯示如何使用 get-image-frame。

AWS CLI

若要取得影像集像素資料

下列get-image-frame程式碼範例會取得影像影格。

```
aws medical-imaging get-image-frame \
    --datastore-id "12345678901234567890123456789012" \
    --image-set-id "98765412345612345678907890789012" \
    --image-frame-information imageFrameId=3abf5d5d7ae72f80a0ec81b2c0de3ef4 \
    imageframe.jph
```

注意：此程式碼範例不包含輸出，因為 GetImageFrame 動作會將像素資料串流傳回至 imageframe.jph 檔案。如需解碼和檢視影像影格的資訊，請參閱 HTJ2K 解碼程式庫。

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的取得影像集像素資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetImageFrame。

get-image-set-metadata

以下程式碼範例顯示如何使用 get-image-set-metadata。

AWS CLI

範例 1：取得不含 版本的映像集中繼資料

下列get-image-set-metadata程式碼範例會取得映像集的中繼資料，而不指定版本。

注意： outfile 是必要的參數

```
aws medical-imaging get-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    studymetadata.json.gz
```

傳回的中繼資料會以 gzip 壓縮，並存放在 studymetadata.json.gz 檔案中。若要檢視傳回 JSON 物件的內容，您必須先解壓縮它。

輸出：

```
{
    "contentType": "application/json",
    "contentEncoding": "gzip"
}
```

範例 2：使用 版本取得影像集中繼資料

下列get-image-set-metadata程式碼範例會取得具有指定版本之映像集的中繼資料。

注意： outfile 是必要的參數

```
aws medical-imaging get-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --version-id 1 \
    studymetadata.json.gz
```

傳回的中繼資料會以 gzip 壓縮，並存放在 studymetadata.json.gz 檔案中。若要檢視傳回 JSON 物件的內容，您必須先解壓縮它。

輸出：

```
{
    "contentType": "application/json",
    "contentEncoding": "gzip"
}
```

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的取得映像集中繼資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetImageSetMetadata。

get-image-set

以下程式碼範例顯示如何使用 get-image-set。

AWS CLI

若要取得影像集屬性

下列get-image-set程式碼範例會取得影像集的屬性。

```
aws medical-imaging get-image-set \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id 18f88ac7870584f58d56256646b4d92b \
    --version-id 1
```

輸出：

```
{
    "versionId": "1",
    "imageSetWorkflowStatus": "COPIED",
    "updatedAt": 1680027253.471,
    "imageSetId": "18f88ac7870584f58d56256646b4d92b",
    "imageSetState": "ACTIVE",
    "createdAt": 1679592510.753,
    "datastoreId": "12345678901234567890123456789012"
}
```

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的取得映像集屬性。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetImageSet。

list-datastores

以下程式碼範例顯示如何使用 list-datastores。

AWS CLI

列出資料存放區

下列list-datastores程式碼範例列出可用的資料存放區。

```
aws medical-imaging list-datastores
```

輸出：

```
{
    "datastoreSummaries": [
        {
            "datastoreId": "12345678901234567890123456789012",
            "datastoreName": "TestDatastore123",
            "datastoreStatus": "ACTIVE",
            "datastoreArn": "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012",
            "createdAt": "2022-11-15T23:33:09.643000+00:00",
            "updatedAt": "2022-11-15T23:33:09.643000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的列出資料存放區。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListDatastores。

`list-dicom-import-jobs`

以下程式碼範例顯示如何使用 `list-dicom-import-jobs`。

AWS CLI

列出 dicom 匯入任務

下列`list-dicom-import-jobs`程式碼範例列出 dicom 匯入任務。

```
aws medical-imaging list-dicom-import-jobs \
    --datastore-id "12345678901234567890123456789012"
```

輸出：

```
{
    "jobSummaries": [
        {
            "jobId": "09876543210987654321098765432109",
            "jobName": "my-job",
            "jobStatus": "COMPLETED",
            "datastoreId": "12345678901234567890123456789012",
```

```
            "dataAccessRoleArn": "arn:aws:iam::123456789012:role/
ImportJobDataAccessRole",
            "endedAt": "2022-08-12T11:21:56.504000+00:00",
            "submittedAt": "2022-08-12T11:20:21.734000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的列出匯入任務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDICOMImportJobs。

list-image-set-versions

以下程式碼範例顯示如何使用 list-image-set-versions。

AWS CLI

列出映像集版本

下列list-image-set-versions程式碼範例會列出映像集的版本歷史記錄。

```
aws medical-imaging list-image-set-versions \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e
```

輸出：

```
{
    "imageSetPropertiesList": [
        {
            "ImageSetWorkflowStatus": "UPDATED",
            "versionId": "4",
            "updatedAt": 1680029436.304,
            "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
            "imageSetState": "ACTIVE",
            "createdAt": 1680027126.436
        },
        {
            "ImageSetWorkflowStatus": "UPDATED",
            "versionId": "3",
            "updatedAt": 1680029163.325,
```

```
            "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
            "imageSetState": "ACTIVE",
            "createdAt": 1680027126.436
        },
        {
            "ImageSetWorkflowStatus": "COPY_FAILED",
            "versionId": "2",
            "updatedAt": 1680027455.944,
            "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
            "imageSetState": "ACTIVE",
            "message": "INVALID_REQUEST:  Series of SourceImageSet and
 DestinationImageSet don't match.",
            "createdAt": 1680027126.436
        },
        {
            "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
            "imageSetState": "ACTIVE",
            "versionId": "1",
            "ImageSetWorkflowStatus": "COPIED",
            "createdAt": 1680027126.436
        }
    ]
}
```

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的列出映像集版本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListImageSetVersions。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

範例 1：列出資料存放區的資源標籤

下列list-tags-for-resource程式碼範例會列出資料存放區的標籤。

```
aws medical-imaging list-tags-for-resource \
    --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012"
```

輸出：

```
{
    "tags":{
        "Deployment":"Development"
    }
}
```

範例 2：列出影像集的資源標籤

下列`list-tags-for-resource`程式碼範例會列出影像集的標籤。

```
aws medical-imaging list-tags-for-resource \
    --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012/
imageset/18f88ac7870584f58d56256646b4d92b"
```

輸出：

```
{
    "tags":{
        "Deployment":"Development"
    }
}
```

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的使用 HealthImaging 標記資源。

AWS HealthImaging

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

search-image-sets

以下程式碼範例顯示如何使用 `search-image-sets`。

AWS CLI

範例 1：使用 EQUAL 運算子搜尋影像集

下列`search-image-sets`程式碼範例使用 EQUAL 運算子，根據特定值搜尋影像集。

```
aws medical-imaging search-image-sets \
    --datastore-id 12345678901234567890123456789012 \
    --search-criteria file://search-criteria.json
```

search-criteria.json 的內容

```
{
    "filters": [{
        "values": [{"DICOMPatientId" : "SUBJECT08701"}],
        "operator": "EQUAL"
    }]
}
```

輸出：

```
{
    "imageSetsMetadataSummaries": [{
        "imageSetId": "09876543210987654321098765432109",
        "createdAt": "2022-12-06T21:40:59.429000+00:00",
        "version": 1,
        "DICOMTags": {
            "DICOMStudyId": "2011201407",
            "DICOMStudyDate": "19991122",
             "DICOMPatientSex": "F",
             "DICOMStudyInstanceUID": "1.2.840.99999999.84710745.943275268089",
             "DICOMPatientBirthDate": "19201120",
             "DICOMStudyDescription": "UNKNOWN",
             "DICOMPatientId": "SUBJECT08701",
             "DICOMPatientName": "Melissa844 Huel628",
             "DICOMNumberOfStudyRelatedInstances": 1,
             "DICOMStudyTime": "140728",
             "DICOMNumberOfStudyRelatedSeries": 1
            },
        "updatedAt": "2022-12-06T21:40:59.429000+00:00"
    }]
}
```

範例 2：使用 DICOMStudyDate 和 DICOMStudyTime 以 BETWEEN 運算子搜尋影像集

下列search-image-sets程式碼範例會搜尋 1990 年 1 月 1 日至 2023 年 1 月 1 日 (12：00 AM) 之間產生 DICOM 檢查的影像集。

注意：DICOMStudyTime 是選用的。如果不存在，上午 12：00 （一天的開始） 是提供用於篩選日期的時間值。

```
aws medical-imaging search-image-sets \
```

```
    --datastore-id 12345678901234567890123456789012 \
    --search-criteria file://search-criteria.json
```

search-criteria.json 的內容

```
{
    "filters": [{
        "values": [{
            "DICOMStudyDateAndTime": {
                "DICOMStudyDate": "19900101",
                "DICOMStudyTime": "000000"
            }
        },
        {
            "DICOMStudyDateAndTime": {
                "DICOMStudyDate": "20230101",
                "DICOMStudyTime": "000000"
            }
        }],
        "operator": "BETWEEN"
    }]
}
```

輸出：

```
{
    "imageSetsMetadataSummaries": [{
        "imageSetId": "09876543210987654321098765432109",
        "createdAt": "2022-12-06T21:40:59.429000+00:00",
        "version": 1,
        "DICOMTags": {
            "DICOMStudyId": "2011201407",
            "DICOMStudyDate": "19991122",
            "DICOMPatientSex": "F",
            "DICOMStudyInstanceUID": "1.2.840.99999999.84710745.943275268089",
            "DICOMPatientBirthDate": "19201120",
            "DICOMStudyDescription": "UNKNOWN",
            "DICOMPatientId": "SUBJECT08701",
            "DICOMPatientName": "Melissa844 Huel628",
            "DICOMNumberOfStudyRelatedInstances": 1,
            "DICOMStudyTime": "140728",
            "DICOMNumberOfStudyRelatedSeries": 1
        },
```

```
        "updatedAt": "2022-12-06T21:40:59.429000+00:00"
    }]
}
```

範例 3：使用 createdAt 使用 BETWEEN 運算子搜尋影像集 （ 先前保留研究的時間 ）

下列search-image-sets程式碼範例會搜尋 DICOM 檢查在 UTC 時區的時間範圍之間保留在 HealthImaging 中的影像集。

注意：以範例格式提供 createdAt ("1985-04-12T23：20：50.52Z")。

```
aws medical-imaging search-image-sets \
    --datastore-id 12345678901234567890123456789012 \
    --search-criteria  file://search-criteria.json
```

search-criteria.json 的內容

```
{
    "filters": [{
        "values": [{
            "createdAt": "1985-04-12T23:20:50.52Z"
        },
        {
            "createdAt": "2022-04-12T23:20:50.52Z"
        }],
        "operator": "BETWEEN"
    }]
}
```

輸出：

```
{
    "imageSetsMetadataSummaries": [{
        "imageSetId": "09876543210987654321098765432109",
        "createdAt": "2022-12-06T21:40:59.429000+00:00",
        "version": 1,
        "DICOMTags": {
            "DICOMStudyId": "2011201407",
            "DICOMStudyDate": "19991122",
            "DICOMPatientSex": "F",
            "DICOMStudyInstanceUID": "1.2.840.99999999.84710745.943275268089",
            "DICOMPatientBirthDate": "19201120",
```

```
            "DICOMStudyDescription": "UNKNOWN",
            "DICOMPatientId": "SUBJECT08701",
            "DICOMPatientName": "Melissa844 Huel628",
            "DICOMNumberOfStudyRelatedInstances": 1,
            "DICOMStudyTime": "140728",
            "DICOMNumberOfStudyRelatedSeries": 1
        },
        "lastUpdatedAt": "2022-12-06T21:40:59.429000+00:00"
    }]
}
```

範例 4：在 DICOMSeriesInstanceUID 上使用 EQUAL 運算子搜尋映像集，並在 updatedAt updatedAt 上以 ASC 順序排序回應

下列search-image-sets程式碼範例會搜尋 DICOMSeriesInstanceUID 上的 EQUAL 運算子和 updatedAt 上的 BETWEEN 的影像集，並在 updatedAt 欄位中依 ASC 順序排序回應。

注意：以範例格式 ("1985-04-12T23：20：50.52Z") 提供 updatedAt。

```
aws medical-imaging search-image-sets \
    --datastore-id 12345678901234567890123456789012 \
    --search-criteria  file://search-criteria.json
```

search-criteria.json 的內容

```
{
    "filters": [{
        "values": [{
            "updatedAt": "2024-03-11T15:00:05.074000-07:00"
        }, {
            "updatedAt": "2024-03-11T16:00:05.074000-07:00"
        }],
        "operator": "BETWEEN"
    }, {
        "values": [{
            "DICOMSeriesInstanceUID": "1.2.840.99999999.84710745.943275268089"
        }],
        "operator": "EQUAL"
    }],
    "sort": {
        "sortField": "updatedAt",
        "sortOrder": "ASC"
    }
```

```
}
```

輸出：

```
{
    "imageSetsMetadataSummaries": [{
        "imageSetId": "09876543210987654321098765432109",
        "createdAt": "2022-12-06T21:40:59.429000+00:00",
        "version": 1,
        "DICOMTags": {
            "DICOMStudyId": "2011201407",
            "DICOMStudyDate": "19991122",
            "DICOMPatientSex": "F",
            "DICOMStudyInstanceUID": "1.2.840.99999999.84710745.943275268089",
            "DICOMPatientBirthDate": "19201120",
            "DICOMStudyDescription": "UNKNOWN",
            "DICOMPatientId": "SUBJECT08701",
            "DICOMPatientName": "Melissa844 Huel628",
            "DICOMNumberOfStudyRelatedInstances": 1,
            "DICOMStudyTime": "140728",
            "DICOMNumberOfStudyRelatedSeries": 1
        },
        "lastUpdatedAt": "2022-12-06T21:40:59.429000+00:00"
    }]
}
```

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的搜尋映像集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchImageSets。

start-dicom-import-job

以下程式碼範例顯示如何使用 start-dicom-import-job。

AWS CLI

啟動 dicom 匯入任務

下列start-dicom-import-job程式碼範例會啟動 dicom 匯入任務。

```
aws medical-imaging start-dicom-import-job \
    --job-name "my-job" \
    --datastore-id "12345678901234567890123456789012" \
```

```
    --input-s3-uri "s3://medical-imaging-dicom-input/dicom_input/" \
    --output-s3-uri "s3://medical-imaging-output/job_output/" \
    --data-access-role-arn "arn:aws:iam::123456789012:role/ImportJobDataAccessRole"
```

輸出：

```
{
    "datastoreId": "12345678901234567890123456789012",
    "jobId": "09876543210987654321098765432109",
    "jobStatus": "SUBMITTED",
    "submittedAt": "2022-08-12T11:28:11.152000+00:00"
}
```

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的啟動匯入任務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartDICOMImportJob。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

範例 1：標記資料存放區

下列tag-resource程式碼範例會標記資料存放區。

```
aws medical-imaging tag-resource \
  --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012" \
  --tags '{"Deployment":"Development"}'
```

此命令不會產生輸出。

範例 2：標記映像集

下列tag-resource程式碼範例會標記影像集。

```
aws medical-imaging tag-resource \
    --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012/
imageset/18f88ac7870584f58d56256646b4d92b" \
```

```
    --tags '{"Deployment":"Development"}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的使用 HealthImaging 標記資源。
AWS HealthImaging

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

範例 1：取消標記資料存放區

下列untag-resource程式碼範例會取消標記資料存放區。

```
aws medical-imaging untag-resource \
    --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012" \
    --tag-keys '["Deployment"]'
```

此命令不會產生輸出。

範例 2：取消標記映像集

下列untag-resource程式碼範例會取消標記影像集。

```
aws medical-imaging untag-resource \
    --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012/
imageset/18f88ac7870584f58d56256646b4d92b" \
    --tag-keys '["Deployment"]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的使用 HealthImaging 標記資源。
AWS HealthImaging

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-image-set-metadata

以下程式碼範例顯示如何使用 update-image-set-metadata。

AWS CLI

範例 1：在影像集中繼資料中插入或更新屬性

下列update-image-set-metadata範例會在影像集中繼資料中插入或更新屬性。

```
aws medical-imaging update-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --latest-version-id 1 \
    --cli-binary-format raw-in-base64-out \
    --update-image-set-metadata-updates file://metadata-updates.json
```

metadata-updates.json 的內容

```
{
    "DICOMUpdates": {
        "updatableAttributes": "{\"SchemaVersion\":1.1,\"Patient\":{\"DICOM\":
{\"PatientName\":\"MX^MX\"}}}"
    }
}
```

輸出：

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
    "updatedAt": 1680042257.908,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
}
```

範例 2：從影像集中繼資料中移除屬性

下列update-image-set-metadata範例會從影像集中繼資料中移除屬性。

```
aws medical-imaging update-image-set-metadata \
```

```
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --latest-version-id 1 \
    --cli-binary-format raw-in-base64-out \
    --update-image-set-metadata-updates file://metadata-updates.json
```

metadata-updates.json 的內容

```
{
    "DICOMUpdates": {
        "removableAttributes": "{\"SchemaVersion\":1.1,\"Study\":{\"DICOM\":
{\"StudyDescription\":\"CHEST\"}}}"
    }
}
```

輸出：

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
    "updatedAt": 1680042257.908,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
}
```

範例 3：從映像集中繼資料中移除執行個體

下列update-image-set-metadata範例會從映像集中繼資料中移除執行個體。

```
aws medical-imaging update-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --latest-version-id 1 \
    --cli-binary-format raw-in-base64-out \
    --update-image-set-metadata-updates file://metadata-updates.json
```

metadata-updates.json 的內容

```
{
    "DICOMUpdates": {
```

```
        "removableAttributes": "{\"SchemaVersion\": 1.1,\"Study\": {\"Series
\": {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"Instances\":
 {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {}}}}}}"
    }
}
```

輸出：

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
    "updatedAt": 1680042257.908,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
}
```

範例 4：將映像集還原至先前的版本

下列update-image-set-metadata範例示範如何將映像集還原為舊版。CopyImageSet 和 UpdateImageSetMetadata 動作會建立新的映像集版本。

```
aws medical-imaging update-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
    --latest-version-id 3 \
    --cli-binary-format raw-in-base64-out \
    --update-image-set-metadata-updates '{"revertToVersionId": "1"}'
```

輸出：

```
{
    "datastoreId": "12345678901234567890123456789012",
    "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
    "latestVersionId": "4",
    "imageSetState": "LOCKED",
    "imageSetWorkflowStatus": "UPDATING",
    "createdAt": 1680027126.436,
    "updatedAt": 1680042257.908
}
```

範例 5：將私有 DICOM 資料元素新增至執行個體

下列update-image-set-metadata範例顯示如何將私有元素新增至映像集中的指定執行個體。DICOM 標準允許私有資料元素進行無法包含在標準資料元素中的資訊通訊。您可以使用 UpdateImageSetMetadata 動作建立、更新和刪除私有資料元素。

```
aws medical-imaging update-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
    --latest-version-id 1 \
    --cli-binary-format raw-in-base64-out \
    --force \
    --update-image-set-metadata-updates file://metadata-updates.json
```

metadata-updates.json 的內容

```
{
    "DICOMUpdates": {
        "updatableAttributes": "{\"SchemaVersion\": 1.1,\"Study\": {\"Series
\": {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"Instances
\": {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"DICOM\":
 {\"001910F9\": \"97\"},\"DICOMVRs\": {\"001910F9\": \"DS\"}}}}}}"
    }
}
```

輸出：

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
    "updatedAt": 1680042257.908,
    "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
}
```

範例 6：將私有 DICOM 資料元素更新至執行個體

下列update-image-set-metadata範例顯示如何更新屬於映像集中執行個體之私有資料元素的值。

```
aws medical-imaging update-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
    --latest-version-id 1 \
    --cli-binary-format raw-in-base64-out \
    --force \
    --update-image-set-metadata-updates file://metadata-updates.json
```

metadata-updates.json 的內容

```
{
    "DICOMUpdates": {
        "updatableAttributes": "{\"SchemaVersion\": 1.1,\"Study\": {\"Series
\": {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"Instances
\": {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"DICOM\":
 {\"00091001\": \"GE_GENESIS_DD\"}}}}}}"
    }
}
```

輸出：

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
    "updatedAt": 1680042257.908,
    "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
}
```

範例 7：使用 強制參數更新 SOPInstanceUID

下列update-image-set-metadata範例示範如何使用 強制參數來覆寫 DICOM 中繼資料限制條件，以更新 SOPInstanceUID。

```
aws medical-imaging update-image-set-metadata \
        --datastore-id 12345678901234567890123456789012 \
        --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
        --latest-version-id 1 \
        --cli-binary-format raw-in-base64-out \
```

```
        --force \
        --update-image-set-metadata-updates file://metadata-updates.json
```

metadata-updates.json 的內容

```
{
    "DICOMUpdates": {
        "updatableAttributes": "{\"SchemaVersion\":1.1,\"Study\":{\"Series\":
{\"1.3.6.1.4.1.5962.99.1.3633258862.2104868982.1369432891697.3656.0\":{\"Instances
\":{\"1.3.6.1.4.1.5962.99.1.3633258862.2104868982.1369432891697.3659.0\":{\"DICOM\":
{\"SOPInstanceUID\":
\"1.3.6.1.4.1.5962.99.1.3633258862.2104868982.1369432891697.3659.9\"}}}}}}"
    }
}
```

輸出：

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
    "updatedAt": 1680042257.908,
    "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
}
```

如需詳細資訊，請參閱 AWS HealthImaging 開發人員指南中的更新映像集中繼資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateImageSetMetadata。

使用 的 HealthLake 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 HealthLake 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-fhir-datastore`

以下程式碼範例顯示如何使用 `create-fhir-datastore`。

AWS CLI

範例 1：建立SigV4-enabled HealthLake 資料存放區

下列`create-fhir-datastore`範例示範如何在 AWS HealthLake 中建立新的資料存放區。

```
aws healthlake create-fhir-datastore \
    --datastore-type-version R4 \
    --datastore-name "FhirTestDatastore"
```

輸出：

```
{
    "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/(Data
 store ID)/r4/",
    "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/(Data
 store ID)",
    "DatastoreStatus": "CREATING",
    "DatastoreId": "(Data store ID)"
}
```

範例 2：在啟用 FHIR 的 HealthLake 資料存放區上建立 SMART

下列`create-fhir-datastore`範例示範如何在啟用 FHIR 的資料存放區 in AWS HealthLake 上建立新的 SMART。

```
aws healthlake create-fhir-datastore \
    --datastore-name "your-data-store-name" \
    --datastore-type-version R4 \
    --preload-data-config PreloadDataType="SYNTHEA" \
    --sse-configuration '{ "KmsEncryptionConfig": {  "CmkType":
 "CUSTOMER_MANAGED_KMS_KEY", "KmsKeyId": "arn:aws:kms:us-east-1:your-account-id:key/
your-key-id" } }' \
    --identity-provider-configuration  file://identity_provider_configuration.json
```

identity_provider_configuration.json 的內容：

```
{
    "AuthorizationStrategy": "SMART_ON_FHIR_V1",
    "FineGrainedAuthorizationEnabled": true,
    "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-id:function:your-
lambda-name",
    "Metadata": "{\"issuer\":\"https://ehr.example.com\", \"jwks_uri\":\"https://
ehr.example.com/.well-known/jwks.json\",\"authorization_endpoint\":\"https://
ehr.example.com/auth/authorize\",\"token_endpoint\":\"https://ehr.token.com/auth/
token\",\"token_endpoint_auth_methods_supported\":[\"client_secret_basic\",\"foo\"],
\"grant_types_supported\":[\"client_credential\",\"foo\"],\"registration_endpoint\":
\"https://ehr.example.com/auth/register\",\"scopes_supported\":[\"openId\",\"profile
\",\"launch\"],\"response_types_supported\":[\"code\"],\"management_endpoint\":
\"https://ehr.example.com/user/manage\",\"introspection_endpoint\":\"https://
ehr.example.com/user/introspect\",\"revocation_endpoint\":\"https://ehr.example.com/
user/revoke\",\"code_challenge_methods_supported\":[\"S256\"],\"capabilities\":
[\"launch-ehr\",\"sso-openid-connect\",\"client-public\"]}"
}
```

輸出：

```
{
    "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/(Data
 store ID)/r4/",
    "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/(Data
 store ID)",
    "DatastoreStatus": "CREATING",
    "DatastoreId": "(Data store ID)"
}
```

如需詳細資訊，請參閱 AWS HealthLake 開發人員指南中的建立和監控 FHIR 資料存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFHIRDatastore。

delete-fhir-datastore

以下程式碼範例顯示如何使用 delete-fhir-datastore。

AWS CLI

刪除 FHIR 資料存放區

下列delete-fhir-datastore範例示範如何在 AWS HealthLake 中刪除資料存放區及其所有內容。

```
aws healthlake delete-fhir-datastore \
    --datastore-id (Data store ID)
```

輸出：

```
{
    "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/(Data
 store ID)/r4/",
    "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/(Data
 store ID)",
    "DatastoreStatus": "DELETING",
    "DatastoreId": "(Data store ID)"
}
```

如需詳細資訊，請參閱 AWS HealthLake 開發人員指南中的建立和監控 FHIR 資料存放區 <https://docs.aws.amazon.com/healthlake/latest/devguide/working-with-FHIR-healthlake.html>。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFHIRDatastore。

describe-fhir-datastore

以下程式碼範例顯示如何使用 describe-fhir-datastore。

AWS CLI

描述 FHIR 資料存放區

下列describe-fhir-datastore範例示範如何尋找資料存放區 in AWS HealthLake 的屬性。

```
aws healthlake describe-fhir-datastore \
    --datastore-id "1f2f459836ac6c513ce899f9e4f66a59"
```

輸出：

```
{
    "DatastoreProperties": {
        "PreloadDataConfig": {
            "PreloadDataType": "SYNTHEA"
        },
```

```
        "SseConfiguration": {
            "KmsEncryptionConfig": {
                "CmkType": "CUSTOMER_MANAGED_KMS_KEY",
                "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
            }
        },
        "DatastoreName": "Demo",
        "DatastoreArn": "arn:aws:healthlake:us-east-1:<AWS Account ID>:datastore/
<Data store ID>",
        "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/
<Data store ID>/r4/",
        "DatastoreStatus": "ACTIVE",
        "DatastoreTypeVersion": "R4",
        "CreatedAt": 1603761064.881,
        "DatastoreId": "<Data store ID>",
        "IdentityProviderConfiguration": {
            "AuthorizationStrategy": "AWS_AUTH",
            "FineGrainedAuthorizationEnabled": false
        }
    }
}
```

如需詳細資訊，請參閱 AWS HealthLake 開發人員指南中的建立和監控 FHIR 資料存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFHIRDatastore。

describe-fhir-export-job

以下程式碼範例顯示如何使用 describe-fhir-export-job。

AWS CLI

描述 FHIR 匯出任務

下列describe-fhir-export-job範例顯示如何在 AWS HealthLake 中尋找 FHIR 匯出任務的屬性。

```
aws healthlake describe-fhir-export-job \
    --datastore-id (Data store ID) \
    --job-id 9b9a51943afaedd0a8c0c26c49135a31
```

輸出：

```
{
    "ExportJobProperties": {
        "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",
        "JobStatus": "IN_PROGRESS",
        "JobId": "9009813e9d69ba7cf79bcb3468780f16",
        "SubmitTime": "2024-11-20T11:31:46.672000-05:00",
        "EndTime": "2024-11-20T11:34:01.636000-05:00",
        "OutputDataConfig": {
            "S3Configuration": {
            "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",
            "KmsKeyId": "arn:aws:kms:us-east-1:012345678910:key/d330e7fc-b56c-4216-
a250-f4c43ef46e83"
        }

        },
        "DatastoreId": "(Data store ID)"
    }
}
```

如需詳細資訊，請參閱 AWS HealthLake 開發人員指南中的從 FHIR 資料存放區匯出檔案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFHIRExportJob。

describe-fhir-import-job

以下程式碼範例顯示如何使用 describe-fhir-import-job。

AWS CLI

描述 FHIR 匯入任務

下列describe-fhir-import-job範例顯示如何使用 AWS HealthLake 學習 FHIR 匯入任務的屬性。

```
aws healthlake describe-fhir-import-job \
    --datastore-id (Data store ID) \
    --job-id c145fbb27b192af392f8ce6e7838e34f
```

輸出：

```
{
```

```
    "ImportJobProperties": {
    "InputDataConfig": {
        "S3Uri": "s3://(Bucket Name)/(Prefix Name)/"
        { "arrayitem2": 2 }
    },
    "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",
    "JobStatus": "COMPLETED",
    "JobId": "c145fbb27b192af392f8ce6e7838e34f",
    "SubmitTime": 1606272542.161,
    "EndTime": 1606272609.497,
    "DatastoreId": "(Data store ID)"
    }
}
```

如需詳細資訊，請參閱 AWS HealthLake 開發人員指南中的將檔案匯入 FHIR 資料存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFHIRImportJob。

`list-fhir-datastores`

以下程式碼範例顯示如何使用 `list-fhir-datastores`。

AWS CLI

列出 FHIR 資料存放區

下列`list-fhir-datastores`範例顯示如何使用 命令，以及使用者如何根據 AWS HealthLake 中的資料存放區狀態篩選結果。

```
aws healthlake list-fhir-datastores \
    --filter DatastoreStatus=ACTIVE
```

輸出：

```
{
    "DatastorePropertiesList": [
    {
        "PreloadDataConfig": {
            "PreloadDataType": "SYNTHEA"
        },
        "SseConfiguration": {
            "KmsEncryptionConfig": {
```

```
                "CmkType": "CUSTOMER_MANAGED_KMS_KEY",
                "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
            }
        },
        "DatastoreName": "Demo",
        "DatastoreArn": "arn:aws:healthlake:us-east-1:<AWS Account ID>:datastore/
<Data store ID>",
        "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/
<Data store ID>/r4/",
        "DatastoreStatus": "ACTIVE",
        "DatastoreTypeVersion": "R4",
        "CreatedAt": 1603761064.881,
        "DatastoreId": "<Data store ID>",
        "IdentityProviderConfiguration": {
            "AuthorizationStrategy": "AWS_AUTH",
            "FineGrainedAuthorizationEnabled": false
        }
    }
    ]
}
```

如需詳細資訊，請參閱 AWS HealthLake 開發人員指南中的建立和監控 FHIR 資料存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFHIRDatastores。

`list-fhir-export-jobs`

以下程式碼範例顯示如何使用 `list-fhir-export-jobs`。

AWS CLI

列出所有 FHIR 匯出任務

下列`list-fhir-export-jobs`範例示範如何使用 命令來檢視與 帳戶相關聯的匯出任務清單。

```
aws healthlake list-fhir-export-jobs \
    --datastore-id (Data store ID) \
    --submitted-before (DATE like 2024-10-13T19:00:00Z)\
    --submitted-after (DATE like 2020-10-13T19:00:00Z )\
    --job-name "FHIR-EXPORT" \
    --job-status SUBMITTED  \
    --max-results (Integer between 1 and 500)
```

輸出：

```
{
    "ExportJobPropertiesList": [
        {
            "ExportJobProperties": {
                "OutputDataConfig": {
                    "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",
                    "S3Configuration": {
                        "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",
                        "KmsKeyId": "(KmsKey Id)"
                    }
                },
                "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role
 Name)",
                "JobStatus": "COMPLETED",
                "JobId": "c145fbb27b192af392f8ce6e7838e34f",
                "JobName": "FHIR-EXPORT",
                "SubmitTime": "2024-11-20T11:31:46.672000-05:00",
                "EndTime": "2024-11-20T11:34:01.636000-05:00",
                "DatastoreId": "(Data store ID)"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《 AWS HealthLake 開發人員指南》中的從 FHIR 資料存放區匯出檔案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFHIRExportJobs。

`list-fhir-import-jobs`

以下程式碼範例顯示如何使用 `list-fhir-import-jobs`。

AWS CLI

列出所有 FHIR 匯入任務

下列`list-fhir-import-jobs`範例示範如何使用 命令來檢視與 帳戶相關聯的所有匯入任務清單。

```
aws healthlake list-fhir-import-jobs \
```

```
    --datastore-id (Data store ID) \
    --submitted-before (DATE like 2024-10-13T19:00:00Z) \
    --submitted-after (DATE like 2020-10-13T19:00:00Z ) \
    --job-name "FHIR-IMPORT" \
    --job-status SUBMITTED  \
    -max-results (Integer between 1 and 500)
```

輸出：

```
{
    "ImportJobPropertiesList": [
        {
            "JobId": "c0fddbf76f238297632d4aebdbfc9ddf",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2024-11-20T10:08:46.813000-05:00",
            "EndTime": "2024-11-20T10:10:09.093000-05:00",
            "DatastoreId": "(Data store ID)",
            "InputDataConfig": {
                "S3Uri": "s3://(Bucket Name)/(Prefix Name)/"
            },
            "JobOutputDataConfig": {
                "S3Configuration": {
                    "S3Uri": "s3://(Bucket Name)/
import/6407b9ae4c2def3cb6f1a46a0c599ec0-FHIR_IMPORT-
c0fddbf76f238297632d4aebdbfc9ddf/",
                    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/b7f645cb-
e564-4981-8672-9e012d1ff1a0"
                }
            },
            "JobProgressReport": {
                "TotalNumberOfScannedFiles": 1,
                "TotalSizeOfScannedFilesInMB": 0.001798,
                "TotalNumberOfImportedFiles": 1,
                "TotalNumberOfResourcesScanned": 1,
                "TotalNumberOfResourcesImported": 1,
                "TotalNumberOfResourcesWithCustomerError": 0,
                "TotalNumberOfFilesReadWithCustomerError": 0,
                "Throughput": 0.0
            },
            "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)"
        }
    ]
}
```

如需詳細資訊，請參閱《 AWS HealthLake 開發人員指南》中的將檔案匯入 FHIR 資料存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFHIRImportJobs。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出資料存放區的標籤

下列`list-tags-for-resource`範例列出與指定資料存放區相關聯的標籤：

```
aws healthlake list-tags-for-resource \
    --resource-arn "arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/0725c83f4307f263e16fd56b6d8ebdbe"
```

輸出：

```
{
    "tags": {
        "key": "value",
        "key1": "value1"
    }
}
```

如需詳細資訊，請參閱《AWS HealthLake 開發人員指南》中的在HealthLake 中標記資源。 AWS HealthLake

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`start-fhir-export-job`

以下程式碼範例顯示如何使用 `start-fhir-export-job`。

AWS CLI

啟動 FHIR 匯出任務

下列`start-fhir-export-job`範例示範如何使用 AWS HealthLake 啟動 FHIR 匯出任務。

```
aws healthlake start-fhir-export-job \
    --output-data-config '{"S3Configuration": {"S3Uri":"s3://(Bucket Name)/(Prefix
 Name)/","KmsKeyId":"arn:aws:kms:us-east-1:012345678910:key/d330e7fc-b56c-4216-a250-
f4c43ef46e83"}}' \
    --datastore-id (Data store ID) \
    --data-access-role-arn arn:aws:iam::(AWS Account ID):role/(Role Name)
```

輸出：

```
{
    "DatastoreId": "(Data store ID)",
    "JobStatus": "SUBMITTED",
    "JobId": "9b9a51943afaedd0a8c0c26c49135a31"
}
```

如需詳細資訊，請參閱 AWS HealthLake 開發人員指南中的從 FHIR 資料存放區匯出檔案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartFHIRExportJob。

start-fhir-import-job

以下程式碼範例顯示如何使用 start-fhir-import-job。

AWS CLI

啟動 FHIR 匯入任務

下列start-fhir-import-job範例示範如何使用 AWS HealthLake 啟動 FHIR 匯入任務。

```
aws healthlake start-fhir-import-job \
    --input-data-config S3Uri="s3://(Bucket Name)/(Prefix Name)/" \
    --job-output-data-config '{"S3Configuration": {"S3Uri":"s3://(Bucket Name)/
(Prefix Name)/","KmsKeyId":"arn:aws:kms:us-east-1:012345678910:key/d330e7fc-
b56c-4216-a250-f4c43ef46e83"}}' \
    --datastore-id (Data store ID) \
    --data-access-role-arn "arn:aws:iam::(AWS Account ID):role/(Role Name)"
```

輸出：

```
{
    "DatastoreId": "(Data store ID)",
```

```
    "JobStatus": "SUBMITTED",
    "JobId": "c145fbb27b192af392f8ce6e7838e34f"
}
```

如需詳細資訊，請參閱 AWS HealthLake 開發人員指南中的將檔案匯入 FHIR 資料存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartFHIRImportJob。

`tag-resource`

以下程式碼範例顯示如何使用 `tag-resource`。

AWS CLI

將標籤新增至資料存放區

下列`tag-resource`範例示範如何將標籤新增至資料存放區。

```
aws healthlake tag-resource \
    --resource-arn "arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/0725c83f4307f263e16fd56b6d8ebdbe" \
    --tags '[{"Key": "key1", "Value": "value1"}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 HealthLake 開發人員指南中的將標籤新增至資料存放區。 AWS HealthLake

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

`untag-resource`

以下程式碼範例顯示如何使用 `untag-resource`。

AWS CLI

從資料存放區移除標籤。

下列`untag-resource`範例示範如何從資料存放區移除標籤。

```
aws healthlake untag-resource \
    --resource-arn "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/
b91723d65c6fdeb1d26543a49d2ed1fa" \
```

```
    --tag-keys '["key1"]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS HealthLake 開發人員指南中的從資料存放區移除標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

使用 的 HealthOmics 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 HealthOmics 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`abort-multipart-read-set-upload`

以下程式碼範例顯示如何使用 `abort-multipart-read-set-upload`。

AWS CLI

停止分段讀取集上傳

下列`abort-multipart-read-set-upload`範例會停止分段讀取集上傳到您的 HealthOmics 序列存放區。

```
aws omics abort-multipart-read-set-upload \
    --sequence-store-id 0123456789 \
    --upload-id 1122334455
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的直接上傳至序列存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AbortMultipartReadSetUpload。

accept-share

以下程式碼範例顯示如何使用 accept-share。

AWS CLI

接受分析存放區資料的份額

下列accept-share範例接受 HealthOmics 分析存放區資料的共用。

```
aws omics accept-share \
    ----share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

輸出：

```
{
    "status": "ACTIVATING"
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的跨帳戶共用。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptShare。

batch-delete-read-set

以下程式碼範例顯示如何使用 batch-delete-read-set。

AWS CLI

刪除多個讀取集

下列batch-delete-read-set範例會刪除兩個讀取集。

```
aws omics batch-delete-read-set \
    --sequence-store-id 1234567890 \
    --ids 1234567890 0123456789
```

如果刪除任何指定的讀取集時發生錯誤，服務會傳回錯誤清單。

```
{
    "errors": [
        {
            "code": "",
            "id": "0123456789",
            "message": "The specified readset does not exist."
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDeleteReadSet。

cancel-annotation-import-job

以下程式碼範例顯示如何使用 cancel-annotation-import-job。

AWS CLI

若要取消註釋匯入任務

下列cancel-annotation-import-job範例會取消 ID 為 的註釋匯入任務04f57618-xmpl-4fd0-9349-e5a85aefb997。

```
aws omics cancel-annotation-import-job \
    --job-id 04f57618-xmpl-4fd0-9349-e5a85aefb997
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelAnnotationImportJob。

cancel-run

以下程式碼範例顯示如何使用 cancel-run。

AWS CLI

取消執行

下列cancel-run範例會取消 ID 為 的執行1234567。

```
aws omics cancel-run \
    --id 1234567
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的在工作流程中執行生命週期。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelRun。

cancel-variant-import-job

以下程式碼範例顯示如何使用 cancel-variant-import-job。

AWS CLI

若要取消變體匯入任務

下列cancel-variant-import-job範例會取消 ID 為 的變體匯入任務69cb65d6-xmpl-4a4a-9025-4565794b684e。

```
aws omics cancel-variant-import-job \
    --job-id 69cb65d6-xmpl-4a4a-9025-4565794b684e
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelVariantImportJob。

complete-multipart-read-set-upload

以下程式碼範例顯示如何使用 complete-multipart-read-set-upload。

AWS CLI

上傳所有元件後，完成分段上傳。

所有元件上傳完成後，下列complete-multipart-read-set-upload範例會結束分段上傳至序列存放區。

```
aws omics complete-multipart-read-set-upload \
    --sequence-store-id 0123456789 \
    --upload-id 1122334455 \
    --parts '[{"checksum":"gaCBQMe+rpCFZxLpoP6gydBoXaKKDA/
Vobh5zBDb4W4=","partNumber":1,"partSource":"SOURCE1"}]'
```

輸出：

```
{
    "readSetId": "0000000001"
    "readSetId": "0000000002"
    "readSetId": "0000000003"
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的直接上傳至序列存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CompleteMultipartReadSetUpload。

create-annotation-store-version

以下程式碼範例顯示如何使用 create-annotation-store-version。

AWS CLI

建立新版本的註釋存放區

下列create-annotation-store-version範例會建立新的 註釋存放區版本。

```
aws omics create-annotation-store-version \
    --name my_annotation_store \
    --version-name my_version
```

輸出：

```
{
    "creationTime": "2023-07-21T17:15:49.251040+00:00",
    "id": "3b93cdef69d2",
    "name": "my_annotation_store",
    "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:555555555555:referenceStore/6505293348/reference/5987565360"
    },
    "status": "CREATING",
    "versionName": "my_version"
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的建立新版本的註釋存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAnnotationStoreVersion。

create-annotation-store

以下程式碼範例顯示如何使用 create-annotation-store。

AWS CLI

範例 1：建立 VCF 註釋存放區

下列create-annotation-store範例會建立 VCF 格式註釋存放區。

```
aws omics create-annotation-store \
    --name my_ann_store \
    --store-format VCF \
    --reference referenceArn=arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890
```

輸出：

```
{
    "creationTime": "2022-11-23T22:48:39.226492Z",
    "id": "0a91xmplc71f",
    "name": "my_ann_store",
    "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
    },
    "status": "CREATING",
    "storeFormat": "VCF"
}
```

範例 2：建立 TSV 註釋存放區

下列create-annotation-store範例會建立 TSV 格式註釋存放區。

```
aws omics create-annotation-store \
    --name tsv_ann_store \
    --store-format TSV \
    --reference referenceArn=arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890 \
    --store-options file://tsv-store-options.json
```

tsv-store-options.json 設定註釋的格式選項。

```
{
    "tsvStoreOptions": {
        "annotationType": "CHR_START_END_ZERO_BASE",
        "formatToHeader": {
            "CHR": "chromosome",
            "START": "start",
            "END": "end"
        },
        "schema": [
            {
                "chromosome": "STRING"
            },
            {
                "start": "LONG"
            },
            {
                "end": "LONG"
            },
            {
                "name": "STRING"
            }
        ]
    }
}
```

輸出：

```
{
    "creationTime": "2022-11-30T01:28:08.525586Z",
    "id": "861cxmpl96b0",
    "name": "tsv_ann_store",
    "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
    },
    "status": "CREATING",
    "storeFormat": "TSV",
    "storeOptions": {
        "tsvStoreOptions": {
            "annotationType": "CHR_START_END_ZERO_BASE",
            "formatToHeader": {
```

```
                "CHR": "chromosome",
                "END": "end",
                "START": "start"
            },
            "schema": [
                {
                    "chromosome": "STRING"
                },
                {
                    "start": "LONG"
                },
                {
                    "end": "LONG"
                },
                {
                    "name": "STRING"
                }
            ]
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAnnotationStore。

create-multipart-read-set-upload

以下程式碼範例顯示如何使用 create-multipart-read-set-upload。

AWS CLI

開始分段讀取集上傳。

下列create-multipart-read-set-upload範例會啟動分段讀取集上傳。

```
aws omics create-multipart-read-set-upload \
    --sequence-store-id 0123456789 \
    --name HG00146 \
    --source-file-type FASTQ \
    --subject-id mySubject\
    --sample-id mySample\
    --description "FASTQ for HG00146"\
```

```
    --generated-from "1000 Genomes"
```

輸出：

```
{
    "creationTime": "2022-07-13T23:25:20Z",
    "description": "FASTQ for HG00146",
    "generatedFrom": "1000 Genomes",
    "name": "HG00146",
    "sampleId": "mySample",
    "sequenceStoreId": "0123456789",
    "sourceFileType": "FASTQ",
    "subjectId": "mySubject",
    "uploadId": "1122334455"
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的直接上傳至序列存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateMultipartReadSetUpload。

create-reference-store

以下程式碼範例顯示如何使用 create-reference-store。

AWS CLI

建立參考存放區

下列create-reference-store範例會建立參考存放區 my-ref-store。

```
aws omics create-reference-store \
    --name my-ref-store
```

輸出：

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",
    "creationTime": "2022-11-22T22:13:25.947Z",
    "id": "1234567890",
    "name": "my-ref-store"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateReferenceStore。

create-run-group

以下程式碼範例顯示如何使用 create-run-group。

AWS CLI

建立執行群組

下列create-run-group範例會建立名為 的執行群組cram-converter。

```
aws omics create-run-group \
    --name cram-converter \
    --max-cpus 20 \
    --max-gpus 10 \
    --max-duration 600 \
    --max-runs 5
```

輸出：

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",
    "id": "1234567",
    "tags": {}
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的建立執行群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRunGroup。

create-sequence-store

以下程式碼範例顯示如何使用 create-sequence-store。

AWS CLI

建立序列存放區

下列create-sequence-store範例會建立序列存放區。

```
aws omics create-sequence-store \
    --name my-seq-store
```

輸出：

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890",
    "creationTime": "2022-11-23T01:24:33.629Z",
    "id": "1234567890",
    "name": "my-seq-store"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSequenceStore。

create-share

以下程式碼範例顯示如何使用 create-share。

AWS CLI

建立 HealthOmics 分析存放區的共用

下列create-share範例顯示如何建立 HealthOmics 分析存放區的共用，可由帳戶外部的訂閱者接受。

```
aws omics create-share \
    --resource-arn "arn:aws:omics:us-west-2:555555555555:variantStore/
omics_dev_var_store" \
    --principal-subscriber "123456789012" \
    --name "my_Share-123"
```

輸出：

```
{
    "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
    "name": "my_Share-123",
    "status": "PENDING"
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的跨帳戶共用。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateShare。

create-variant-store

以下程式碼範例顯示如何使用 create-variant-store。

AWS CLI

建立變體存放區

下列create-variant-store範例會建立名為 的變體存放區my_var_store。

```
aws omics create-variant-store \
    --name my_var_store \
    --reference referenceArn=arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890
```

輸出：

```
{
    "creationTime": "2022-11-23T22:09:07.534499Z",
    "id": "02dexmplcfdd",
    "name": "my_var_store",
    "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
    },
    "status": "CREATING"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVariantStore。

create-workflow

以下程式碼範例顯示如何使用 create-workflow。

AWS CLI

建立工作流程

下列create-workflow範例會建立 WDL 工作流程。

```
aws omics create-workflow \
    --name cram-converter \
    --engine WDL \
    --definition-zip fileb://workflow-crambam.zip \
    --parameter-template file://workflow-params.json
```

workflow-crambam.zip 是包含工作流程定義的 ZIP 封存檔。 會workflow-params.json定義工作流程的執行時間參數。

```
{
    "ref_fasta" : {
        "description": "Reference genome fasta file",
        "optional": false
    },
    "ref_fasta_index" : {
        "description": "Index of the reference genome fasta file",
        "optional": false
    },
    "ref_dict" : {
        "description": "dictionary file for 'ref_fasta'",
        "optional": false
    },
    "input_cram" : {
        "description": "The Cram file to convert to BAM",
        "optional": false
    },
    "sample_name" : {
        "description": "The name of the input sample, used to name the output BAM",
        "optional": false
    }
}
```

輸出：

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
    "id": "1234567",
    "status": "CREATING",
    "tags": {}
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的建立私有工作流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateWorkflow。

delete-annotation-store-versions

以下程式碼範例顯示如何使用 delete-annotation-store-versions。

AWS CLI

刪除註釋存放區版本

下列delete-annotation-store-versions範例會刪除註釋存放區版本。

```
aws omics delete-annotation-store-versions \
    --name my_annotation_store \
    --versions my_version
```

輸出：

```
{
    "errors": []
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的建立新版本的註釋存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAnnotationStoreVersions。

delete-annotation-store

以下程式碼範例顯示如何使用 delete-annotation-store。

AWS CLI

刪除註釋存放區

下列delete-annotation-store範例會刪除名為 的註釋存放區my_vcf_store。

```
aws omics delete-annotation-store \
    --name my_vcf_store
```

輸出：

```
{
    "status": "DELETING"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAnnotationStore。

delete-reference-store

以下程式碼範例顯示如何使用 delete-reference-store。

AWS CLI

刪除參考存放區

下列delete-reference-store範例會刪除 ID 為 的參考存放區1234567890。

```
aws omics delete-reference-store \
    --id 1234567890
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteReferenceStore。

delete-reference

以下程式碼範例顯示如何使用 delete-reference。

AWS CLI

刪除參考

下列delete-reference範例會刪除參考。

```
aws omics delete-reference \
    --reference-store-id 1234567890 \
    --id 1234567890
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 命令參考》中的 DeleteReference。 AWS CLI

delete-run-group

以下程式碼範例顯示如何使用 delete-run-group。

AWS CLI

刪除執行群組

下列delete-run-group範例會刪除 ID 為 的執行群組1234567。

```
aws omics delete-run-group \
    --id 1234567
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的刪除執行和執行群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRunGroup。

delete-run

以下程式碼範例顯示如何使用 delete-run。

AWS CLI

刪除工作流程執行

下列delete-run範例會刪除 ID 為 的執行1234567。

```
aws omics delete-run \
    --id 1234567
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的刪除執行和執行群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRun。

delete-sequence-store

以下程式碼範例顯示如何使用 delete-sequence-store。

AWS CLI

刪除序列存放區

下列delete-sequence-store範例會刪除 ID 為 的序列存放區1234567890。

```
aws omics delete-sequence-store \
    --id 1234567890
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSequenceStore。

delete-share

以下程式碼範例顯示如何使用 delete-share。

AWS CLI

刪除 HealthOmics 分析資料的份額

下列delete-share範例會刪除分析資料的跨帳戶共用。

```
aws omics delete-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

輸出：

```
{
    "status": "DELETING"
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的跨帳戶共用。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteShare。

delete-variant-store

以下程式碼範例顯示如何使用 delete-variant-store。

AWS CLI

刪除變體存放區

下列delete-variant-store範例會刪除名為 的變體存放區my_var_store。

```
aws omics delete-variant-store \
```

```
    --name my_var_store
```

輸出：

```
{
    "status": "DELETING"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVariantStore。

delete-workflow

以下程式碼範例顯示如何使用 delete-workflow。

AWS CLI

刪除工作流程

下列delete-workflow範例會刪除 ID 為 的工作流程1234567。

```
aws omics delete-workflow \
    --id 1234567
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的刪除私有工作流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteWorkflow。

get-annotation-import-job

以下程式碼範例顯示如何使用 get-annotation-import-job。

AWS CLI

檢視註釋匯入任務

下列get-annotation-import-job範例會取得註釋匯入任務的詳細資訊。

```
aws omics get-annotation-import-job \
    --job-id 984162c7-xmpl-4d23-ab47-286f7950bfbf
```

輸出：

```
{
    "creationTime": "2022-11-30T01:40:11.017746Z",
    "destinationName": "tsv_ann_store",
    "id": "984162c7-xmpl-4d23-ab47-286f7950bfbf",
    "items": [
        {
            "jobStatus": "COMPLETED",
            "source": "s3://omics-artifacts-01d6xmpl4e72dd32/targetedregions.bed.gz"
        }
    ],
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ",
    "runLeftNormalization": false,
    "status": "COMPLETED",
    "updateTime": "2022-11-30T01:42:39.134009Z"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAnnotationImportJob。

get-annotation-store-version

以下程式碼範例顯示如何使用 get-annotation-store-version。

AWS CLI

擷取註釋存放區版本的中繼資料

下列get-annotation-store-version範例會擷取所請求註釋存放區版本的中繼資料。

```
aws omics get-annotation-store-version \
    --name my_annotation_store \
    --version-name my_version
```

輸出：

```
{
    "storeId": "4934045d1c6d",
    "id": "2a3f4a44aa7b",
```

```
    "status": "ACTIVE",
    "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/
my_annotation_store/version/my_version",
    "name": "my_annotation_store",
    "versionName": "my_version",
    "creationTime": "2023-07-21T17:15:49.251040+00:00",
    "updateTime": "2023-07-21T17:15:56.434223+00:00",
    "statusMessage": "",
    "versionSizeBytes": 0
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的建立新版本的註釋存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAnnotationStoreVersion。

get-annotation-store

以下程式碼範例顯示如何使用 get-annotation-store。

AWS CLI

檢視註釋存放區

下列get-annotation-store範例會取得名為 的註釋存放區詳細資訊my_ann_store。

```
aws omics get-annotation-store \
    --name my_ann_store
```

輸出：

```
{
    "creationTime": "2022-11-23T22:48:39.226492Z",
    "id": "0a91xmplc71f",
    "name": "my_ann_store",
    "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
    },
    "status": "CREATING",
    "storeArn": "arn:aws:omics:us-west-2:123456789012:annotationStore/my_ann_store",
    "storeFormat": "VCF",
    "storeSizeBytes": 0,
    "tags": {}
```

```
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAnnotationStore。

get-read-set-activation-job

以下程式碼範例顯示如何使用 get-read-set-activation-job。

AWS CLI

檢視讀取集啟用任務

下列get-read-set-activation-job範例會取得讀取集啟用任務的詳細資訊。

```
aws omics get-read-set-activation-job \
    --sequence-store-id 1234567890 \
    --id 1234567890
```

輸出：

```
{
    "completionTime": "2022-12-06T22:33:42.828Z",
    "creationTime": "2022-12-06T22:32:45.213Z",
    "id": "1234567890",
    "sequenceStoreId": "1234567890",
    "sources": [
        {
            "readSetId": "1234567890",
            "status": "FINISHED",
            "statusMessage": "No activation needed as read set is already in
 ACTIVATING or ACTIVE state."
        }
    ],
    "status": "COMPLETED",
    "statusMessage": "The job completed successfully."
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetReadSetActivationJob。

get-read-set-export-job

以下程式碼範例顯示如何使用 get-read-set-export-job。

AWS CLI

檢視讀取集匯出任務

下列get-read-set-export-job範例會取得讀取集匯出任務的詳細資訊。

```
aws omics get-read-set-export-job \
    --sequence-store-id 1234567890 \
    --id 1234567890
```

輸出：

```
{
    "completionTime": "2022-12-06T22:39:14.491Z",
    "creationTime": "2022-12-06T22:37:18.612Z",
    "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",
    "id": "1234567890",
    "sequenceStoreId": "1234567890",
    "status": "COMPLETED",
    "statusMessage": "The job is submitted and will start soon."
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetReadSetExportJob。

get-read-set-import-job

以下程式碼範例顯示如何使用 get-read-set-import-job。

AWS CLI

檢視讀取集匯入任務

下列get-read-set-import-job範例會取得讀取集匯入任務的詳細資訊。

```
aws omics get-read-set-import-job \
    --sequence-store-id 1234567890 \
    --id 1234567890
```

輸出：


```
{
    "creationTime": "2022-11-23T01:36:38.158Z",
    "id": "1234567890",
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ",
    "sequenceStoreId": "1234567890",
    "sources": [
        {
            "name": "HG00100",
            "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890",
            "sampleId": "bam-sample",
            "sourceFileType": "BAM",
            "sourceFiles": {
                "source1": "s3://omics-artifacts-01d6xmpl4e72dd32/
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam",
                "source2": ""
            },
            "status": "IN_PROGRESS",
            "statusMessage": "The source job is currently in progress.",
            "subjectId": "bam-subject",
            "tags": {
                "aws:omics:sampleId": "bam-sample",
                "aws:omics:subjectId": "bam-subject"
            }
        },
        {
            "name": "HG00146",
            "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890",
            "sampleId": "fastq-sample",
            "sourceFileType": "FASTQ",
            "sourceFiles": {
                "source1": "s3://omics-artifacts-01d6xmpl4e72dd32/
SRR233106_1.filt.fastq.gz",
                "source2": "s3://omics-artifacts-01d6xmpl4e72dd32/
SRR233106_2.filt.fastq.gz"
            },
            "status": "IN_PROGRESS",
            "statusMessage": "The source job is currently in progress.",
            "subjectId": "fastq-subject",
            "tags": {
```

```
                "aws:omics:sampleId": "fastq-sample",
                "aws:omics:subjectId": "fastq-subject"
            }
        },
        {
            "name": "HG00096",
            "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890",
            "sampleId": "cram-sample",
            "sourceFileType": "CRAM",
            "sourceFiles": {
                "source1": "s3://omics-artifacts-01d6xmpl4e72dd32/
HG00096.alt_bwamem_GRCh38DH.20150718.GBR.low_coverage.cram",
                "source2": ""
            },
            "status": "IN_PROGRESS",
            "statusMessage": "The source job is currently in progress.",
            "subjectId": "cram-subject",
            "tags": {
                "aws:omics:sampleId": "cram-sample",
                "aws:omics:subjectId": "cram-subject"
            }
        }
    ],
    "status": "IN_PROGRESS",
    "statusMessage": "The job is currently in progress."
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetReadSetImportJob。

get-read-set-metadata

以下程式碼範例顯示如何使用 get-read-set-metadata。

AWS CLI

檢視讀取集

下列get-read-set-metadata範例會取得讀取集檔案的詳細資訊。

```
aws omics get-read-set-metadata \
    --sequence-store-id 1234567890 \
```

```
    --id 1234567890
```

輸出：

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890/
readSet/1234567890",
    "creationTime": "2022-11-23T21:55:00.515Z",
    "fileType": "FASTQ",
    "files": {
        "source1": {
            "contentLength": 310054739,
            "partSize": 104857600,
            "totalParts": 3
        },
        "source2": {
            "contentLength": 307846621,
            "partSize": 104857600,
            "totalParts": 3
        }
    },
    "id": "1234567890",
    "name": "HG00146",
    "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/
reference/1234567890",
    "sampleId": "fastq-sample",
    "sequenceInformation": {
        "alignment": "UNALIGNED",
        "totalBaseCount": 677717384,
        "totalReadCount": 8917334
    },
    "sequenceStoreId": "1234567890",
    "status": "ACTIVE",
    "subjectId": "fastq-subject"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetReadSetMetadata。

get-read-set

以下程式碼範例顯示如何使用 get-read-set。

AWS CLI

下載讀取集

下列get-read-set範例會將讀取集的第 3 部分下載為 1234567890.3.bam。

```
aws omics get-read-set \
    --sequence-store-id 1234567890 \
    --id 1234567890 \
    --part-number 3  1234567890.3.bam
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetReadSet。

get-reference-import-job

以下程式碼範例顯示如何使用 get-reference-import-job。

AWS CLI

檢視參考匯入任務

下列get-reference-import-job範例會取得參考匯入任務的詳細資訊。

```
aws omics get-reference-import-job \
    --reference-store-id 1234567890 \
    --id 1234567890
```

輸出：

```
{
    "creationTime": "2022-11-22T22:25:41.124Z",
    "id": "1234567890",
    "referenceStoreId": "1234567890",
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ",
    "sources": [
        {
            "name": "assembly-38",
            "sourceFile": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.fasta",
```

```
            "status": "IN_PROGRESS",
            "statusMessage": "The source job is currently in progress."
        }
    ],
    "status": "IN_PROGRESS",
    "statusMessage": "The job is currently in progress."
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetReferenceImportJob。

get-reference-metadata

以下程式碼範例顯示如何使用 get-reference-metadata。

AWS CLI

檢視參考

下列get-reference-metadata範例會取得參考的詳細資訊。

```
aws omics get-reference-metadata \
    --reference-store-id 1234567890 \
    --id 1234567890
```

輸出：

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/
reference/1234567890",
    "creationTime": "2022-11-22T22:27:09.033Z",
    "files": {
        "index": {
            "contentLength": 160928,
            "partSize": 104857600,
            "totalParts": 1
        },
        "source": {
            "contentLength": 3249912778,
            "partSize": 104857600,
            "totalParts": 31
        }
```

```
    },
    "id": "1234567890",
    "md5": "7ff134953dcca8c8997453bbb80b6b5e",
    "name": "assembly-38",
    "referenceStoreId": "1234567890",
    "status": "ACTIVE",
    "updateTime": "2022-11-22T22:27:09.033Z"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetReferenceMetadata。

get-reference-store

以下程式碼範例顯示如何使用 get-reference-store。

AWS CLI

檢視參考存放區

下列get-reference-store範例會取得參考存放區的詳細資訊。

```
aws omics get-reference-store \
    --id 1234567890
```

輸出：

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",
    "creationTime": "2022-09-23T23:27:20.364Z",
    "id": "1234567890",
    "name": "my-rstore-0"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetReferenceStore。

get-reference

以下程式碼範例顯示如何使用 get-reference。

AWS CLI

下載基因體參考

下列get-reference範例會將基因體的第 1 部分下載為 hg38.1.fa。

```
aws omics get-reference \
    --reference-store-id 1234567890 \
    --id 1234567890 \
    --part-number 1 hg38.1.fa
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetReference。

get-run-group

以下程式碼範例顯示如何使用 get-run-group。

AWS CLI

檢視執行群組

下列get-run-group範例會取得執行群組的詳細資訊。

```
aws omics get-run-group \
    --id 1234567
```

輸出：

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",
    "creationTime": "2022-12-01T00:58:42.915219Z",
    "id": "1234567",
    "maxCpus": 20,
    "maxDuration": 600,
    "name": "cram-convert",
    "tags": {}
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的建立執行群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRunGroup。

get-run-task

以下程式碼範例顯示如何使用 get-run-task。

AWS CLI

檢視任務

下列get-run-task範例會取得工作流程任務的詳細資訊。

```
aws omics get-run-task \
    --id 1234567 \
    --task-id 1234567
```

輸出：

```
{
    "cpus": 1,
    "creationTime": "2022-11-30T23:13:00.718651Z",
    "logStream": "arn:aws:logs:us-west-2:123456789012:log-group:/aws/omics/
WorkflowLog:log-stream:run/1234567/task/1234567",
    "memory": 15,
    "name": "CramToBamTask",
    "startTime": "2022-11-30T23:17:47.016Z",
    "status": "COMPLETED",
    "stopTime": "2022-11-30T23:18:21.503Z",
    "taskId": "1234567"
}
```

如需詳細資訊，請參閱 HealthOmics 使用者指南中的 HealthOmics 中執行的任務生命週期。 AWS HealthOmics

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRunTask。

get-run

以下程式碼範例顯示如何使用 get-run。

AWS CLI

檢視工作流程執行

下列get-run範例會取得工作流程執行的詳細資訊。

```
aws omics get-run \
    --id 1234567
```

輸出：

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
    "creationTime": "2022-11-30T22:58:22.615865Z",
    "digest":
 "sha256:c54bxmpl742dcc26f7fa1f10e37550ddd8f251f418277c0a58e895b801ed28cf",
    "id": "1234567",
    "name": "cram-to-bam",
    "outputUri": "s3://omics-artifacts-01d6xmpl4e72dd32/workflow-output/",
    "parameters": {
        "ref_dict": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.dict",
        "ref_fasta_index": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.fasta.fai",
        "ref_fasta": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.fasta",
        "sample_name": "NA12878",
        "input_cram": "s3://omics-artifacts-01d6xmpl4e72dd32/NA12878.cram"
    },
    "resourceDigests": {
        "s3://omics-artifacts-01d6xmpl4e72dd32/Homo_sapiens_assembly38.fasta.fai":
 "etag:f76371b113734a56cde236bc0372de0a",
        "s3://omics-artifacts-01d6xmpl4e72dd32/Homo_sapiens_assembly38.dict":
 "etag:3884c62eb0e53fa92459ed9bff133ae6",
        "s3://omics-artifacts-01d6xmpl4e72dd32/Homo_sapiens_assembly38.fasta":
 "etag:e307d81c605fb91b7720a08f00276842-388",
        "s3://omics-artifacts-01d6xmpl4e72dd32/NA12878.cram":
 "etag:a9f52976381286c6143b5cc681671ec6"
    },
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ",
    "startedBy": "arn:aws:iam::123456789012:user/laptop-2020",
    "status": "STARTING",
    "tags": {},
    "workflowId": "1234567",
    "workflowType": "PRIVATE"
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的在工作流程中執行生命週期。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRun。

get-sequence-store

以下程式碼範例顯示如何使用 get-sequence-store。

AWS CLI

檢視序列存放區

下列get-sequence-store範例取得 ID 為 之序列存放區的詳細資訊1234567890。

```
aws omics get-sequence-store \
    --id 1234567890
```

輸出：

```
{
    "arn": "arn:aws:omics:us-east-1:123456789012:sequenceStore/1234567890",
    "creationTime": "2022-11-23T19:55:48.376Z",
    "id": "1234567890",
    "name": "my-seq-store"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSequenceStore。

get-share

以下程式碼範例顯示如何使用 get-share。

AWS CLI

擷取 HealthOmics 分析資料共享的相關中繼資料

下列get-share範例會擷取分析資料跨帳戶共用的中繼資料。

```
aws omics get-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

輸出：

```
{
    "share": {
        "shareId": "495c21bedc889d07d0ab69d710a6841e-
dd75ab7a1a9c384fa848b5bd8e5a7e0a",
        "name": "my_Share-123",
        "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/
omics_dev_var_store",
        "principalSubscriber": "123456789012",
        "ownerId": "555555555555",
        "status": "PENDING"
    }
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的跨帳戶共用。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetShare。

get-variant-import-job

以下程式碼範例顯示如何使用 get-variant-import-job。

AWS CLI

檢視變體匯入任務

下列get-variant-import-job範例會取得變體匯入任務的詳細資訊。

```
aws omics get-variant-import-job \
    --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

輸出：

```
{
    "creationTime": "2022-11-23T22:42:50.037812Z",
    "destinationName": "my_var_store",
    "id": "edd7b8ce-xmpl-47e2-bc99-258cac95a508",
    "items": [
        {
            "jobStatus": "IN_PROGRESS",
```

```
            "source": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.known_indels.vcf.gz"
        }
    ],
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ",
    "runLeftNormalization": false,
    "status": "IN_PROGRESS",
    "updateTime": "2022-11-23T22:43:05.898309Z"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetVariantImportJob。

get-variant-store

以下程式碼範例顯示如何使用 get-variant-store。

AWS CLI

檢視變體存放區

下列get-variant-store範例會取得變體存放區的詳細資訊。

```
aws omics get-variant-store \
    --name my_var_store
```

輸出：

```
{
    "creationTime": "2022-11-23T22:09:07.534499Z",
    "id": "02dexmplcfdd",
    "name": "my_var_store",
    "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
    },
    "status": "CREATING",
    "storeArn": "arn:aws:omics:us-west-2:123456789012:variantStore/my_var_store",
    "storeSizeBytes": 0,
    "tags": {},
    "updateTime": "2022-11-23T22:09:24.931711Z"
```

```
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetVariantStore。

get-workflow

以下程式碼範例顯示如何使用 get-workflow。

AWS CLI

檢視工作流程

下列get-workflow範例取得 ID 為 之工作流程的詳細資訊1234567。

```
aws omics get-workflow \
    --id 1234567
```

輸出：

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
    "creationTime": "2022-11-30T22:33:16.225368Z",
    "digest":
 "sha256:c54bxmpl742dcc26f7fa1f10e37550ddd8f251f418277c0a58e895b801ed28cf",
    "engine": "WDL",
    "id": "1234567",
    "main": "workflow-crambam.wdl",
    "name": "cram-converter",
    "parameterTemplate": {
        "ref_dict": {
            "description": "dictionary file for 'ref_fasta'"
        },
        "ref_fasta_index": {
            "description": "Index of the reference genome fasta file"
        },
        "ref_fasta": {
            "description": "Reference genome fasta file"
        },
        "input_cram": {
            "description": "The Cram file to convert to BAM"
        },
```

```
        "sample_name": {
            "description": "The name of the input sample, used to name the output
 BAM"
        }
    },
    "status": "ACTIVE",
    "statusMessage": "workflow-crambam.wdl\n    workflow CramToBamFlow\n
 call CramToBamTask\n        call ValidateSamFile\n    task CramToBamTask\n    task
 ValidateSamFile\n",
    "tags": {},
    "type": "PRIVATE"
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的建立私有工作流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetWorkflow。

`list-annotation-import-jobs`

以下程式碼範例顯示如何使用 `list-annotation-import-jobs`。

AWS CLI

若要取得註釋匯入任務的清單

以下內容`list-annotation-import-jobs`會取得註釋匯入任務的清單。

```
aws omics list-annotation-import-jobs
```

輸出：

```
{
    "annotationImportJobs": [
        {
            "creationTime": "2022-11-30T01:39:41.478294Z",
            "destinationName": "gff_ann_store",
            "id": "18a9e792-xmpl-4869-a105-e5b602900444",
            "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W8O1XMPL7QZ",
            "runLeftNormalization": false,
            "status": "COMPLETED",
            "updateTime": "2022-11-30T01:47:09.145178Z"
        },
```

```
        {
            "creationTime": "2022-11-30T00:45:58.007838Z",
            "destinationName": "my_ann_store",
            "id": "4e9eafc8-xmpl-431e-a0b2-3bda27cb600a",
            "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W8O1XMPL7QZ",
            "runLeftNormalization": false,
            "status": "FAILED",
            "updateTime": "2022-11-30T00:47:01.706325Z"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAnnotationImportJobs。

`list-annotation-store-versions`

以下程式碼範例顯示如何使用 `list-annotation-store-versions`。

AWS CLI

列出註釋存放區的所有版本。

下列`list-annotation-store-versions`範例列出 註釋存放區存在的所有版本。

```
aws omics list-annotation-store-versions \
    --name my_annotation_store
```

輸出：

```
{
    "annotationStoreVersions": [
        {
        "storeId": "4934045d1c6d",
        "id": "2a3f4a44aa7b",
        "status": "CREATING",
        "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/
my_annotation_store/version/my_version_2",
        "name": "my_annotation_store",
        "versionName": "my_version_2",
        "creation Time": "2023-07-21T17:20:59.380043+00:00",
```

```
        "versionSizeBytes": 0
},
{
     "storeId": "4934045d1c6d",
     "id": "4934045d1c6d",
     "status": "ACTIVE",
     "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/
my_annotation_store/version/my_version_1",
     "name": "my_annotation_store",
     "versionName": "my_version_1",
     "creationTime": "2023-07-21T17:15:49.251040+00:00",
     "updateTime": "2023-07-21T17:15:56.434223+00:00",
     "statusMessage": "",
     "versionSizeBytes": 0
     }

}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的建立新版本的註釋存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAnnotationStoreVersions。

`list-annotation-stores`

以下程式碼範例顯示如何使用 `list-annotation-stores`。

AWS CLI

取得註釋存放區的清單

下列`list-annotation-stores`範例會取得註釋存放區的清單。

```
aws omics list-annotation-stores
```

輸出：

```
{
    "annotationStores": [
        {
            "creationTime": "2022-11-23T22:48:39.226492Z",
            "id": "0a91xmplc71f",
            "name": "my_ann_store",
            "reference": {
```

```
                "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
            },
            "status": "ACTIVE",
            "statusMessage": "",
            "storeArn": "arn:aws:omics:us-west-2:123456789012:annotationStore/
my_ann_store",
            "storeFormat": "VCF",
            "storeSizeBytes": 0,
            "updateTime": "2022-11-23T22:53:27.372840Z"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAnnotationStores。

list-multipart-read-set-uploads

以下程式碼範例顯示如何使用 list-multipart-read-set-uploads。

AWS CLI

列出所有分段讀取集上傳及其狀態。

下列list-multipart-read-set-uploads範例列出所有分段讀取集上傳及其狀態。

```
aws omics list-multipart-read-set-uploads \
    --sequence-store-id 0123456789
```

輸出：

```
{
"uploads":
    [
        {
           "sequenceStoreId": "0123456789",
           "uploadId": "8749584421",
           "sourceFileType": "FASTQ",
            "subjectId": "mySubject",
            "sampleId": "mySample",
            "generatedFrom": "1000 Genomes",
```

```
            "name": "HG00146",
            "description": "FASTQ for HG00146",
            "creationTime": "2023-11-29T19:22:51.349298+00:00"
        },
        {
            "sequenceStoreId": "0123456789",
            "uploadId": "5290538638",
            "sourceFileType": "BAM",
            "subjectId": "mySubject",
            "sampleId": "mySample",
            "generatedFrom": "1000 Genomes",
            "referenceArn": "arn:aws:omics:us-
west-2:845448930428:referenceStore/8168613728/reference/2190697383",
            "name": "HG00146",
            "description": "BAM for HG00146",
            "creationTime": "2023-11-29T19:23:33.116516+00:00"
        },
        {
            "sequenceStoreId": "0123456789",
            "uploadId": "4174220862",
            "sourceFileType": "BAM",
            "subjectId": "mySubject",
            "sampleId": "mySample",
            "generatedFrom": "1000 Genomes",
            "referenceArn": "arn:aws:omics:us-
west-2:845448930428:referenceStore/8168613728/reference/2190697383",
            "name": "HG00147",
            "description": "BAM for HG00147",
            "creationTime": "2023-11-29T19:23:47.007866+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的直接上傳至序列存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMultipartReadSetUploads。

list-read-set-activation-jobs

以下程式碼範例顯示如何使用 `list-read-set-activation-jobs`。

AWS CLI

取得讀取集啟用任務的清單

下列list-read-set-activation-jobs範例會取得 ID 為 之序列存放區的啟用任務清單1234567890。

```
aws omics list-read-set-activation-jobs \
    --sequence-store-id 1234567890
```

輸出：

```
{
    "activationJobs": [
        {
            "completionTime": "2022-12-06T22:33:42.828Z",
            "creationTime": "2022-12-06T22:32:45.213Z",
            "id": "1234567890",
            "sequenceStoreId": "1234567890",
            "status": "COMPLETED"
        },
        {
            "creationTime": "2022-12-06T22:35:10.100Z",
            "id": "1234567890",
            "sequenceStoreId": "1234567890",
            "status": "IN_PROGRESS"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListReadSetActivationJobs。

list-read-set-export-jobs

以下程式碼範例顯示如何使用 list-read-set-export-jobs。

AWS CLI

取得讀取集匯出任務的清單

下列list-read-set-export-jobs範例會取得 ID 為 之序列存放區的匯出任務清單1234567890。

```
aws omics list-read-set-export-jobs \
```

```
    --sequence-store-id 1234567890
```

輸出：

```
{
    "exportJobs": [
        {
            "completionTime": "2022-12-06T22:39:14.491Z",
            "creationTime": "2022-12-06T22:37:18.612Z",
            "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",
            "id": "1234567890",
            "sequenceStoreId": "1234567890",
            "status": "COMPLETED"
        },
        {
            "creationTime": "2022-12-06T22:38:04.871Z",
            "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",
            "id": "1234567890",
            "sequenceStoreId": "1234567890",
            "status": "IN_PROGRESS"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListReadSetExportJobs。

list-read-set-import-jobs

以下程式碼範例顯示如何使用 list-read-set-import-jobs。

AWS CLI

若要取得讀取集匯入任務的清單

下列list-read-set-import-jobs範例會取得 ID 為 之序列存放區的匯入任務清單1234567890。

```
aws omics list-read-set-import-jobs \
    --sequence-store-id 1234567890
```

輸出：

```
{
    "importJobs": [
        {
            "completionTime": "2022-11-29T18:17:49.244Z",
            "creationTime": "2022-11-29T17:32:47.700Z",
            "id": "1234567890",
            "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W8O1XMPL7QZ",
            "sequenceStoreId": "1234567890",
            "status": "COMPLETED"
        },
        {
            "completionTime": "2022-11-23T22:01:34.090Z",
            "creationTime": "2022-11-23T21:52:43.289Z",
            "id": "1234567890",
            "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W8O1XMPL7QZ",
            "sequenceStoreId": "1234567890",
            "status": "COMPLETED_WITH_FAILURES"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListReadSetImportJobs。

list-read-set-upload-parts

以下程式碼範例顯示如何使用 list-read-set-upload-parts。

AWS CLI

列出序列存放區所請求分段上傳中的所有組件。

下列list-read-set-upload-parts範例列出序列存放區所請求分段上傳中的所有組件。

```
aws omics list-read-set-upload-parts \
    --sequence-store-id 0123456789 \
    --upload-id 1122334455 \
    --part-source SOURCE1
```

輸出：

```
{
    "parts": [
        {
            "partNumber": 1,
            "partSize": 94371840,
            "file": "SOURCE1",
            "checksum":
 "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635",
            "lastUpdatedTime": "2023-02-02T20:14:47.533000+00:00"
        }
        {
            "partNumber": 2,
            "partSize": 10471840,
            "file": "SOURCE1",
            "checksum":
 "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635",
            "lastUpdatedTime": "2023-02-02T20:14:47.533000+00:00"
        }
      ]

}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的直接上傳至序列存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListReadSetUploadParts。

list-read-sets

以下程式碼範例顯示如何使用 list-read-sets。

AWS CLI

取得讀取集的清單

下列list-read-sets範例會取得 ID 為 之序列存放區的讀取集清單1234567890。

```
aws omics list-read-sets \
    --sequence-store-id 1234567890
```

輸出：

```
{
    "readSets": [
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890/
readSet/1234567890",
            "creationTime": "2022-11-23T21:55:00.515Z",
            "fileType": "FASTQ",
            "id": "1234567890",
            "name": "HG00146",
            "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890",
            "sampleId": "fastq-sample",
            "sequenceStoreId": "1234567890",
            "status": "ACTIVE",
            "subjectId": "fastq-subject"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListReadSets。

list-reference-import-jobs

以下程式碼範例顯示如何使用 list-reference-import-jobs。

AWS CLI

若要取得參考匯入任務的清單

下列list-reference-import-jobs範例會取得 ID 為 之參考存放區的參考匯入任務清單1234567890。

```
aws omics list-reference-import-jobs \
    --reference-store-id 1234567890
```

輸出：

```
{
    "importJobs": [
        {
```

```
            "completionTime": "2022-11-23T19:54:58.204Z",
            "creationTime": "2022-11-23T19:53:20.729Z",
            "id": "1234567890",
            "referenceStoreId": "1234567890",
            "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W8O1XMPL7QZ",
            "status": "COMPLETED"
        },
        {
            "creationTime": "2022-11-23T20:34:03.250Z",
            "id": "1234567890",
            "referenceStoreId": "1234567890",
            "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W8O1XMPL7QZ",
            "status": "IN_PROGRESS"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListReferenceImportJobs。

list-reference-stores

以下程式碼範例顯示如何使用 list-reference-stores。

AWS CLI

取得參考存放區的清單

下列list-reference-stores範例會取得參考存放區的清單。

```
aws omics list-reference-stores
```

輸出：

```
{
    "referenceStores": [
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",
            "creationTime": "2022-11-22T22:13:25.947Z",
            "id": "1234567890",
```

```
            "name": "my-ref-store"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListReferenceStores。

`list-references`

以下程式碼範例顯示如何使用 `list-references`。

AWS CLI

取得參考清單

下列`list-references`範例會取得 ID 為 之參考存放區的基因體參考清單`1234567890`。

```
aws omics list-references \
    --reference-store-id 1234567890
```

輸出：

```
{
    "references": [
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/
reference/1234567890",
            "creationTime": "2022-11-22T22:27:09.033Z",
            "id": "1234567890",
            "md5": "7ff134953dcca8c8997453bbb80b6b5e",
            "name": "assembly-38",
            "referenceStoreId": "1234567890",
            "status": "ACTIVE",
            "updateTime": "2022-11-22T22:27:09.033Z"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListReferences。

list-run-groups

以下程式碼範例顯示如何使用 list-run-groups。

AWS CLI

取得執行群組的清單

下列list-run-groups範例會取得執行群組的清單。

```
aws omics list-run-groups
```

輸出：

```
{
    "items": [
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",
            "creationTime": "2022-12-01T00:58:42.915219Z",
            "id": "1234567",
            "maxCpus": 20,
            "maxDuration": 600,
            "name": "cram-convert"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的建立執行群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRunGroups。

list-run-tasks

以下程式碼範例顯示如何使用 list-run-tasks。

AWS CLI

取得任務清單

下列list-run-tasks範例會取得工作流程執行的任務清單。

```
aws omics list-run-tasks \
```

```
    --id 1234567
```

輸出：

```
{
    "items": [
        {
            "cpus": 1,
            "creationTime": "2022-11-30T23:13:00.718651Z",
            "memory": 15,
            "name": "CramToBamTask",
            "startTime": "2022-11-30T23:17:47.016Z",
            "status": "COMPLETED",
            "stopTime": "2022-11-30T23:18:21.503Z",
            "taskId": "1234567"
        },
        {
            "cpus": 1,
            "creationTime": "2022-11-30T23:18:32.315606Z",
            "memory": 4,
            "name": "ValidateSamFile",
            "startTime": "2022-11-30T23:23:40.165Z",
            "status": "COMPLETED",
            "stopTime": "2022-11-30T23:24:14.766Z",
            "taskId": "1234567"
        }
    ]
}
```

如需詳細資訊，請參閱 HealthOmics 使用者指南中的 HealthOmics 中執行的任務生命週期。 AWS HealthOmics

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRunTasks。

`list-runs`

以下程式碼範例顯示如何使用 `list-runs`。

AWS CLI

取得工作流程執行的清單

下列`list-runs`範例會取得工作流程執行的清單。

```
aws omics list-runs
```

輸出：

```
{
    "items": [
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
            "creationTime": "2022-12-02T23:20:01.202074Z",
            "id": "1234567",
            "name": "cram-to-bam",
            "priority": 1,
            "startTime": "2022-12-02T23:29:18.115Z",
            "status": "COMPLETED",
            "stopTime": "2022-12-02T23:57:54.428812Z",
            "storageCapacity": 10,
            "workflowId": "1234567"
        },
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
            "creationTime": "2022-12-03T00:16:57.180066Z",
            "id": "1234567",
            "name": "cram-to-bam",
            "priority": 1,
            "startTime": "2022-12-03T00:26:50.233Z",
            "status": "FAILED",
            "stopTime": "2022-12-03T00:37:21.451340Z",
            "storageCapacity": 10,
            "workflowId": "1234567"
        },
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
            "creationTime": "2022-12-05T17:57:08.444817Z",
            "id": "1234567",
            "name": "cram-to-bam",
            "status": "STARTING",
            "workflowId": "1234567"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的在工作流程中執行生命週期。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRuns。

`list-sequence-stores`

以下程式碼範例顯示如何使用 `list-sequence-stores`。

AWS CLI

取得序列存放區的清單

下列`list-sequence-stores`範例會取得序列存放區的清單。

```
aws omics list-sequence-stores
```

輸出：

```
{
    "sequenceStores": [
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890",
            "creationTime": "2022-11-23T01:24:33.629Z",
            "id": "1234567890",
            "name": "my-seq-store"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSequenceStores。

`list-shares`

以下程式碼範例顯示如何使用 `list-shares`。

AWS CLI

列出 HealthOmics 分析資料的可用共享

下列`list-shares`範例列出為資源擁有者建立的所有共用。

```
aws omics list-shares \
```

```
    --resource-owner SELF
```

輸出：

```
{
    "shares": [
        {
            "shareId": "595c1cbd-a008-4eca-a887-954d30c91c6e",
            "name": "myShare",
            "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/
store_1",
            "principalSubscriber": "123456789012",
            "ownerId": "555555555555",
            "status": "PENDING"
        }
        {
            "shareId": "39b65d0d-4368-4a19-9814-b0e31d73c10a",
            "name": "myShare3456",
            "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/
store_2",
            "principalSubscriber": "123456789012",
            "ownerId": "555555555555",
            "status": "ACTIVE"
        },
        {
            "shareId": "203152f5-eef9-459d-a4e0-a691668d44ef",
            "name": "myShare4",
            "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/
store_3",
            "principalSubscriber": "123456789012",
            "ownerId": "555555555555",
            "status": "ACTIVE"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的跨帳戶共用。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListShares。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

取得標籤清單

下列list-tags-for-resource範例會取得 ID 為 之工作流程的標籤清單1234567。

```
aws omics list-tags-for-resource \
    --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567
```

輸出：

```
{
    "tags": {
        "department": "analytics"
    }
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的在 Amazon Omics 中標記資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

list-variant-import-jobs

以下程式碼範例顯示如何使用 list-variant-import-jobs。

AWS CLI

若要取得變體匯入任務的清單

下列list-variant-import-jobs範例會取得變體匯入任務的清單。

```
aws omics list-variant-import-jobs
```

輸出：

```
{
    "variantImportJobs": [
        {
            "creationTime": "2022-11-23T22:47:02.514002Z",
            "destinationName": "my_var_store",
            "id": "69cb65d6-xmpl-4a4a-9025-4565794b684e",
```

```
            "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W8O1XMPL7QZ",
            "runLeftNormalization": false,
            "status": "COMPLETED",
            "updateTime": "2022-11-23T22:49:17.976597Z"
        },
        {
            "creationTime": "2022-11-23T22:42:50.037812Z",
            "destinationName": "my_var_store",
            "id": "edd7b8ce-xmpl-47e2-bc99-258cac95a508",
            "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W8O1XMPL7QZ",
            "runLeftNormalization": false,
            "status": "COMPLETED",
            "updateTime": "2022-11-23T22:45:26.009880Z"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListVariantImportJobs。

`list-variant-stores`

以下程式碼範例顯示如何使用 `list-variant-stores`。

AWS CLI

取得變體存放區的清單

下列`list-variant-stores`範例會取得變體存放區的清單。

```
aws omics list-variant-stores
```

輸出：

```
{
    "variantStores": [
        {
            "creationTime": "2022-11-23T22:09:07.534499Z",
            "id": "02dexmplcfdd",
```

```
            "name": "my_var_store",
            "reference": {
                "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
            },
            "status": "CREATING",
            "storeArn": "arn:aws:omics:us-west-2:123456789012:variantStore/
my_var_store",
            "storeSizeBytes": 0,
            "updateTime": "2022-11-23T22:09:24.931711Z"
        },
        {
            "creationTime": "2022-09-23T23:00:09.140265Z",
            "id": "8777xmpl1a24",
            "name": "myvstore0",
            "status": "ACTIVE",
            "storeArn": "arn:aws:omics:us-west-2:123456789012:variantStore/
myvstore0",
            "storeSizeBytes": 0,
            "updateTime": "2022-09-23T23:03:26.013220Z"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListVariantStores。

`list-workflows`

以下程式碼範例顯示如何使用 `list-workflows`。

AWS CLI

取得工作流程清單

下列`list-workflows`範例會取得工作流程清單。

```
aws omics list-workflows
```

輸出：

```
{
```

```
    "items": [
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
            "creationTime": "2022-09-23T23:08:22.041227Z",
            "digest": "nSCNo/qMWFxmplXpUdokXJnwgneOaxyyc2YOxVxrJTE=",
            "id": "1234567",
            "name": "my-wkflow-0",
            "status": "ACTIVE",
            "type": "PRIVATE"
        },
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
            "creationTime": "2022-11-30T22:33:16.225368Z",
            "digest":
 "sha256:c54bxmpl742dcc26f7fa1f10e37550ddd8f251f418277c0a58e895b801ed28cf",
            "id": "1234567",
            "name": "cram-converter",
            "status": "ACTIVE",
            "type": "PRIVATE"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的建立私有工作流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListWorkflows。

start-annotation-import-job

以下程式碼範例顯示如何使用 start-annotation-import-job。

AWS CLI

匯入註釋

下列start-annotation-import-job範例會從 Amazon S3 匯入註釋。

```
aws omics start-annotation-import-job \
    --destination-name tsv_ann_store \
    --no-run-left-normalization \
    --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ \
    --items source=s3://omics-artifacts-01d6xmpl4e72dd32/targetedregions.bed.gz
```

輸出：

```
{
    "jobId": "984162c7-xmpl-4d23-ab47-286f7950bfbf"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartAnnotationImportJob。

start-read-set-activation-job

以下程式碼範例顯示如何使用 start-read-set-activation-job。

AWS CLI

啟用封存的讀取集

下列start-read-set-activation-job範例會啟用兩個讀取集。

```
aws omics start-read-set-activation-job \
    --sequence-store-id 1234567890 \
    --sources readSetId=1234567890 readSetId=1234567890
```

輸出：

```
{
    "creationTime": "2022-12-06T22:35:10.100Z",
    "id": "1234567890",
    "sequenceStoreId": "1234567890",
    "status": "SUBMITTED"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartReadSetActivationJob。

start-read-set-export-job

以下程式碼範例顯示如何使用 start-read-set-export-job。

AWS CLI

匯出讀取集

下列start-read-set-export-job範例會將兩個讀取集匯出至 Amazon S3。

```
   aws omics start-read-set-export-job \
       --sequence-store-id 1234567890 \
       --sources readSetId=1234567890 readSetId=1234567890 \
       --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ
\
       --destination s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/
```

輸出：

```
{
    "creationTime": "2022-12-06T22:37:18.612Z",
    "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",
    "id": "1234567890",
    "sequenceStoreId": "1234567890",
    "status": "SUBMITTED"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartReadSetExportJob。

start-read-set-import-job

以下程式碼範例顯示如何使用 start-read-set-import-job。

AWS CLI

匯入讀取集

下列start-read-set-import-job範例會匯入讀取集。

```
aws omics start-read-set-import-job \
    --sequence-store-id 1234567890 \
    --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ \
```

```
    --sources file://readset-sources.json
```

readset-sources.json 是具有下列內容的 JSON 文件。

```
[
    {
        "sourceFiles":
        {
            "source1": "s3://omics-artifacts-01d6xmpl4e72dd32/
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam"
        },
        "sourceFileType": "BAM",
        "subjectId": "bam-subject",
        "sampleId": "bam-sample",
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890",
        "name": "HG00100"
    }
]
```

輸出：

```
{
    "creationTime": "2022-11-23T01:36:38.158Z",
    "id": "1234567890",
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ",
    "sequenceStoreId": "1234567890",
    "status": "SUBMITTED"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartReadSetImportJob。

start-reference-import-job

以下程式碼範例顯示如何使用 start-reference-import-job。

AWS CLI

匯入參考基因體

下列start-reference-import-job範例會從 Amazon S3 匯入參考基因體。

```
aws omics start-reference-import-job \
    --reference-store-id 1234567890 \
    --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-W801XMPL7QZ \
    --sources sourceFile=s3://omics-artifacts-01d6xmpl4e72dd32/Homo_sapiens_assembly38.fasta,name=assembly-38
```

輸出：

```
{
    "creationTime": "2022-11-22T22:25:41.124Z",
    "id": "1234567890",
    "referenceStoreId": "1234567890",
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-W801XMPL7QZ",
    "status": "SUBMITTED"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartReferenceImportJob。

start-run

以下程式碼範例顯示如何使用 start-run。

AWS CLI

執行工作流程

下列start-run範例會執行 ID 為 的工作流程1234567。

```
aws omics start-run \
    --workflow-id 1234567 \
    --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-W801XMPL7QZ \
    --name 'cram-to-bam' \
    --output-uri s3://omics-artifacts-01d6xmpl4e72dd32/workflow-output/ \
    --run-group-id 1234567 \
    --priority 1 \
```

```
    --storage-capacity 10 \
    --log-level ALL \
    --parameters file://workflow-inputs.json
```

workflow-inputs.json 是 JSON 文件，內容如下。

```
{
    "sample_name": "NA12878",
    "input_cram": "s3://omics-artifacts-01d6xmpl4e72dd32/NA12878.cram",
    "ref_dict": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.dict",
    "ref_fasta": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.fasta",
    "ref_fasta_index": "omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.fasta.fai"
}
```

輸出：

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
    "id": "1234567",
    "status": "PENDING",
    "tags": {}
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的開始執行。

從 Amazon Omics 載入來源檔案

您也可以使用服務特定的 URIs，從 Amazon Omics 儲存體載入來源檔案。下列範例 workflow-inputs.json 檔案會將 Amazon Omics URIs 用於讀取集和參考基因體來源。

```
{
    "sample_name": "NA12878",
    "input_cram": "omics://123456789012.storage.us-west-2.amazonaws.com/1234567890/
readSet/1234567890/source1",
    "ref_dict": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.dict",
    "ref_fasta": "omics://123456789012.storage.us-west-2.amazonaws.com/1234567890/
reference/1234567890",
```

```
    "ref_fasta_index": "omics://123456789012.storage.us-
west-2.amazonaws.com/1234567890/reference/1234567890/index"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartRun。

start-variant-import-job

以下程式碼範例顯示如何使用 start-variant-import-job。

AWS CLI

匯入變體檔案

下列start-variant-import-job範例會匯入 VCF 格式變體檔案。

```
aws omics start-variant-import-job \
    --destination-name my_var_store \
    --no-run-left-normalization  \
    --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ \
    --items source=s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.known_indels.vcf.gz
```

輸出：

```
{
    "jobId": "edd7b8ce-xmpl-47e2-bc99-258cac95a508"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartVariantImportJob。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

標記資源

下列tag-resource範例會將department標籤新增至 ID 為 的工作流程1234567。

```
aws omics tag-resource \
    --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567 \
    --tags department=analytics
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的在 Amazon Omics 中標記資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源移除標籤

下列untag-resource範例會從工作流程中移除department標籤。

```
aws omics untag-resource \
    --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567 \
    --tag-keys department
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Storage。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-annotation-store

以下程式碼範例顯示如何使用 update-annotation-store。

AWS CLI

更新註釋存放區

下列update-annotation-store範例會更新名為 的註釋存放區描述my_vcf_store。

```
aws omics update-annotation-store \
    --name my_vcf_store \
```

```
    --description "VCF annotation store"
```

輸出：

```
{
    "creationTime": "2022-12-05T18:00:56.101860Z",
    "description": "VCF annotation store",
    "id": "bd6axmpl2444",
    "name": "my_vcf_store",
    "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
    },
    "status": "ACTIVE",
    "storeFormat": "VCF",
    "updateTime": "2022-12-05T18:13:16.100051Z"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAnnotationStore。

update-run-group

以下程式碼範例顯示如何使用 update-run-group。

AWS CLI

更新執行群組

下列update-run-group範例會更新 ID 為 之執行群組的設定1234567。

```
aws omics update-run-group \
    --id 1234567 \
    --max-cpus 10
```

輸出：

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",
    "creationTime": "2022-12-01T00:58:42.915219Z",
    "id": "1234567",
```

```
    "maxCpus": 10,
    "maxDuration": 600,
    "name": "cram-convert",
    "tags": {}
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics 工作流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRunGroup。

update-variant-store

以下程式碼範例顯示如何使用 update-variant-store。

AWS CLI

更新變體存放區

下列update-variant-store範例會更新名為 的變體存放區描述my_var_store。

```
aws omics update-variant-store \
    --name my_var_store \
    --description "variant store"
```

輸出：

```
{
    "creationTime": "2022-11-23T22:09:07.534499Z",
    "description": "variant store",
    "id": "02dexmplcfdd",
    "name": "my_var_store",
    "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
    },
    "status": "ACTIVE",
    "updateTime": "2022-12-05T18:23:37.686402Z"
}
```

如需詳細資訊，請參閱《Amazon Omics 開發人員指南》中的 Omics Analytics。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateVariantStore。

update-workflow

以下程式碼範例顯示如何使用 update-workflow。

AWS CLI

更新工作流程

下列update-workflow範例會更新 ID 為 之工作流程的描述1234567。

```
aws omics update-workflow \
    --id 1234567 \
    --description "copy workflow"
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的建立或更新工作流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateWorkflow。

upload-read-set-part

以下程式碼範例顯示如何使用 upload-read-set-part。

AWS CLI

上傳讀取集部分。

下列upload-read-set-part範例會上傳讀取集的指定部分。

```
aws omics upload-read-set-part \
    --sequence-store-id 0123456789 \
    --upload-id 1122334455 \
    --part-source SOURCE1 \
    --part-number 1 \
    --payload /path/to/file/read_1_part_1.fastq.gz
```

輸出：

```
{
    "checksum": "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635"
}
```

如需詳細資訊，請參閱 AWS HealthOmics 使用者指南中的直接上傳至序列存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UploadReadSetPart。

使用 的 IAM 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 IAM 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`add-client-id-to-open-id-connect-provider`

以下程式碼範例顯示如何使用 `add-client-id-to-open-id-connect-provider`。

AWS CLI

若要將用戶端 ID (對象) 新增至 Open-ID Connect (OIDC) 提供者

下列 `add-client-id-to-open-id-connect-provider` 命令會將用戶端 ID `my-application-ID` 新增至名為 `server.example.com` 的 OIDC 提供者。

```
aws iam add-client-id-to-open-id-connect-provider \
    --client-id my-application-ID \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com
```

此命令不會產生輸出。

若要建立 OIDC 提供者，請使用 `create-open-id-connect-provider` 命令。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Creating OpenID Connect (OIDC) identity providers。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 AddClientIdToOpenIdConnectProvider。

add-role-to-instance-profile

以下程式碼範例顯示如何使用 add-role-to-instance-profile。

AWS CLI

若要將角色新增至執行個體設定檔

下列 add-role-to-instance-profile 命令會將名為 S3Access 的角色新增至名為 Webserver 的執行個體設定檔。

```
aws iam add-role-to-instance-profile \
    --role-name S3Access \
    --instance-profile-name Webserver
```

此命令不會產生輸出。

若要建立執行個體設定檔，請使用 create-instance-profile 命令。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用 IAM 角色為在 Amazon EC2 執行個體上執行的應用程式授予許可。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 AddRoleToInstanceProfile。

add-user-to-group

以下程式碼範例顯示如何使用 add-user-to-group。

AWS CLI

將使用者新增至 IAM 群組

下列 add-user-to-group 命令會將名為 Bob 的 IAM 使用者新增至名為 Admins 的 IAM 群組。

```
aws iam add-user-to-group \
    --user-name Bob \
    --group-name Admins
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的在 IAM 使用者群組中新增和移除使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AddUserToGroup。

attach-group-policy

以下程式碼範例顯示如何使用 attach-group-policy。

AWS CLI

若要將受管政策連接至 IAM 群組

下列attach-group-policy命令會將名為 的 AWS 受管政策連接至名為 ReadOnlyAccess的 IAM 群組Finance。

```
aws iam attach-group-policy \
    --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \
    --group-name Finance
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的受管政策和內嵌政策。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 AttachGroupPolicy。

attach-role-policy

以下程式碼範例顯示如何使用 attach-role-policy。

AWS CLI

將受管政策連接至 IAM 角色

下列attach-role-policy命令會將名為 的 AWS 受管政策連接至名為 ReadOnlyAccess的 IAM 角色ReadOnlyRole。

```
aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \
    --role-name ReadOnlyRole
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的受管政策和內嵌政策。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AttachRolePolicy。

attach-user-policy

以下程式碼範例顯示如何使用 attach-user-policy。

AWS CLI

將受管政策連接至 IAM 使用者

下列attach-user-policy命令會將名為 的 AWS 受管政策連接至名為 AdministratorAccess的 IAM 使用者Alice。

```
aws iam attach-user-policy \
    --policy-arn arn:aws:iam::aws:policy/AdministratorAccess \
    --user-name Alice
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的受管政策和內嵌政策。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AttachUserPolicy。

change-password

以下程式碼範例顯示如何使用 change-password。

AWS CLI

若要變更您的 IAM 使用者的密碼

若要變更 IAM 使用者的密碼，建議使用 --cli-input-json 參數來傳遞包含舊密碼和新密碼的 JSON 檔案。採用此方法時，可以使用含非英數字元的高保護性密碼。當您以命令列參數形式傳遞密碼時，使用含非英數字元的密碼可能會有困難。若要使用 --cli-input-json 參數，請先從使用含 --generate-cli-skeleton 參數的 change-password 命令開始，範例如下所示。

```
aws iam change-password \
    --generate-cli-skeleton > change-password.json
```

先前的命令會建立名為 change-password.json 的 JSON 檔案，可用來填入舊密碼與新密碼。例如，此檔案可能如下所示：

```
{
```

```
    "OldPassword": "3s0K_;xh4~8XXI",
    "NewPassword": "]35d/{pB9Fo9wJ"
}
```

接下來，若要變更密碼，請再次使用 change-password 命令，這次傳遞 --cli-input-json 參數以指定 JSON 檔案。下列 change-password 命令會將 --cli-input-json 參數與名為 change-password.json 的 JSON 檔案搭配使用。

```
aws iam change-password \
    --cli-input-json file://change-password.json
```

此命令不會產生輸出。

此命令只能由 IAM 使用者呼叫。如果使用 AWS 帳戶 （根） 登入資料呼叫此命令，則命令會傳回InvalidUserType錯誤。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 IAM 使用者如何變更自己的密碼。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ChangePassword。

create-access-key

以下程式碼範例顯示如何使用 create-access-key。

AWS CLI

為 IAM 使用者建立存取金鑰

下列 create-access-key 命令會為名為 Bob 的 IAM 使用者建立存取金鑰 (存取金鑰 ID 與私密存取金鑰)。

```
aws iam create-access-key \
    --user-name Bob
```

輸出：

```
{
    "AccessKey": {
        "UserName": "Bob",
        "Status": "Active",
        "CreateDate": "2015-03-09T18:39:23.411Z",
```

```
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"
    }
}
```

請將私密存取金鑰存放在安全之處。遺失的金鑰無法復原，您必須建立新的存取金鑰。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的管理 IAM 使用者的存取金鑰。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateAccessKey。

create-account-alias

以下程式碼範例顯示如何使用 create-account-alias。

AWS CLI

建立帳戶別名

下列create-account-alias命令會examplecorp為您的 AWS 帳戶建立別名。

```
aws iam create-account-alias \
    --account-alias examplecorp
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的AWS 您的帳戶 ID 及其別名。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateAccountAlias。

create-group

以下程式碼範例顯示如何使用 create-group。

AWS CLI

建立 IAM 群組

下列 create-group 命令會建立名為 Admins 的 IAM 群組。

```
aws iam create-group \
    --group-name Admins
```

輸出：

```
{
    "Group": {
        "Path": "/",
        "CreateDate": "2015-03-09T20:30:24.940Z",
        "GroupId": "AIDGPMS9RO4H3FEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:group/Admins",
        "GroupName": "Admins"
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的建立 IAM 使用者群組。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateGroup。

create-instance-profile

以下程式碼範例顯示如何使用 create-instance-profile。

AWS CLI

建立執行個體設定檔

下列 create-instance-profile 命令會建立名為 Webserver 的執行個體設定檔。

```
aws iam create-instance-profile \
    --instance-profile-name Webserver
```

輸出：

```
{
    "InstanceProfile": {
        "InstanceProfileId": "AIPAJMBYC7DLSPEXAMPLE",
        "Roles": [],
        "CreateDate": "2015-03-09T20:33:19.626Z",
        "InstanceProfileName": "Webserver",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:instance-profile/Webserver"
    }
}
```

若要將角色新增至執行個體設定檔，請使用 add-role-to-instance-profile 命令。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用 IAM 角色為在 Amazon EC2 執行個體上執行的應用程式授予許可。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateInstanceProfile。

create-login-profile

以下程式碼範例顯示如何使用 create-login-profile。

AWS CLI

為 IAM 使用者建立密碼

若要為 IAM 使用者建立密碼，建議使用 --cli-input-json 參數傳遞包含密碼的 JSON 檔案。採用此方法時，可以建立含非英數字元的強式密碼。當您以命令列參數形式傳遞密碼時，建立含非英數字元的密碼可能會很困難。

若要使用 --cli-input-json 參數，請先使用 create-login-profile 命令搭配 --generate-cli-skeleton 參數，如下列範例中所示。

```
aws iam create-login-profile \
    --generate-cli-skeleton > create-login-profile.json
```

先前的命令會建立名為 create-login-profile.json 的 JSON 檔案，可用來填入後續 create-login-profile 命令的資訊。例如：

```
{
    "UserName": "Bob",
    "Password": "&1-3a6u:RA0djs",
    "PasswordResetRequired": true
}
```

接下來，若要為 IAM 使用者建立密碼，請再次使用 create-login-profile 命令，這次傳遞 --cli-input-json 參數來指定 JSON 檔案。下列 create-login-profile 命令會將 --cli-input-json 參數搭配名為 create-login-profile.json 的 JSON 檔案使用。

```
aws iam create-login-profile \
    --cli-input-json file://create-login-profile.json
```

輸出：

```
{
    "LoginProfile": {
        "UserName": "Bob",
        "CreateDate": "2015-03-10T20:55:40.274Z",
        "PasswordResetRequired": true
    }
}
```

如果新密碼違反帳戶密碼政策，則命令會傳回 PasswordPolicyViolation 錯誤。

若要為已有密碼的使用者變更密碼，請使用 update-login-profile。若要設定帳戶的密碼政策，請使用 update-account-password-policy 命令。

如果帳戶密碼政策允許，IAM 使用者可以使用 change-password 命令變更自己的密碼。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的管理 IAM 使用者的密碼。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 CreateLoginProfile。

create-open-id-connect-provider

以下程式碼範例顯示如何使用 create-open-id-connect-provider。

AWS CLI

若要建立 IAM OpenID Connect (OIDC) 提供者

若要建立 OpenID Connect (OIDC) 提供者，建議使用 --cli-input-json 參數來傳遞包含必要參數的 JSON 檔案。建立 OIDC 提供者時，必須傳遞提供者的 URL，且 URL 必須以 https:// 開頭。以命令列參數形式傳遞 URL 可能會很困難，因為在某些命令列環境中，冒號 (:) 和正斜線 (/) 字元有特殊含義。使用 --cli-input-json 參數可以避開這個限制。

若要使用 --cli-input-json 參數，請先使用 create-open-id-connect-provider 命令搭配 --generate-cli-skeleton 參數，如下列範例中所示。

```
aws iam create-open-id-connect-provider \
    --generate-cli-skeleton > create-open-id-connect-provider.json
```

上一個命令會建立名為 create-open-id-connect-provider.json 的 JSON 檔案，用來填入後續 create-open-id-connect-provider 命令的資訊。例如：

```
{
    "Url": "https://server.example.com",
    "ClientIDList": [
        "example-application-ID"
    ],
    "ThumbprintList": [
        "c3768084dfb3d2b68b7897bf5f565da8eEXAMPLE"
    ]
}
```

接下來，若要建立 OpenID Connect (OIDC) 提供者，請再次使用 `create-open-id-connect-provider` 命令，這次傳遞 `--cli-input-json` 參數來指定 JSON 檔案。下列 `create-open-id-connect-provider` 命令會將 `--cli-input-json` 參數與名為 create-open-id-connect-provider.json 的 JSON 檔案搭配使用。

```
aws iam create-open-id-connect-provider \
    --cli-input-json file://create-open-id-connect-provider.json
```

輸出：

```
{
    "OpenIDConnectProviderArn": "arn:aws:iam::123456789012:oidc-provider/
server.example.com"
}
```

如需有關 OIDC 提供者的詳細資訊，請參閱《AWS IAM 使用者指南》中的建立 OpenID Connect (OIDC) 身分提供者。

如需如何取得 OIDC 提供者指紋的詳細資訊，請參閱《AWS IAM 使用者指南》中的取得 OpenID Connect 身分提供者的指紋。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 CreateOpenIdConnectProvider。

`create-policy-version`

以下程式碼範例顯示如何使用 `create-policy-version`。

AWS CLI

建立新版本的受管政策

此範例會建立新 v2 版的 IAM 政策 (其 ARN 為 arn:aws:iam::123456789012:policy/MyPolicy)，並將該版本設為預設版本。

```
aws iam create-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --policy-document file://NewPolicyVersion.json \
    --set-as-default
```

輸出：

```
{
    "PolicyVersion": {
        "CreateDate": "2015-06-16T18:56:03.721Z",
        "VersionId": "v2",
        "IsDefaultVersion": true
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 IAM 政策的版本控制。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreatePolicyVersion。

create-policy

以下程式碼範例顯示如何使用 create-policy。

AWS CLI

範例 1：建立客戶管理政策

下列命令會建立名為 my-policy 的客戶管理政策。檔案 policy.json 是目前資料夾中的 JSON 文件，在名為 amzn-s3-demo-bucket 的 Amazon S3 儲存貯體中授予 shared 資料夾的唯讀存取權限。

```
aws iam create-policy \
    --policy-name my-policy \
    --policy-document file://policy.json
```

policy.json 的內容：

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/shared/*"
            ]
        }
    ]
}
```

輸出：

```
{
    "Policy": {
        "PolicyName": "my-policy",
        "CreateDate": "2015-06-01T19:31:18.620Z",
        "AttachmentCount": 0,
        "IsAttachable": true,
        "PolicyId": "ZXR6A36LTYANPAI7NJ5UV",
        "DefaultVersionId": "v1",
        "Path": "/",
        "Arn": "arn:aws:iam::0123456789012:policy/my-policy",
        "UpdateDate": "2015-06-01T19:31:18.620Z"
    }
}
```

如需使用檔案做為字串參數輸入的詳細資訊，請參閱《CLI AWS 使用者指南》中的指定 CLI 的參數值。 AWS

範例 2：建立內含描述的客戶管理政策

下列命令會建立名為 my-policy 的客戶管理政策，其中包含不可變的描述。

檔案 policy.json 是目前資料夾中的 JSON 文件，可針對名為 amzn-s3-demo-bucket 的 Amazon S3 儲存貯體，授予所有 Put、List 和 Get 動作的存取權限。

```
aws iam create-policy \
    --policy-name my-policy \
```

```
    --policy-document file://policy.json \
    --description "This policy grants access to all Put, Get, and List actions for
 amzn-s3-demo-bucket"
```

policy.json 的內容：

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Effect": "Allow",
           "Action": [
                "s3:ListBucket*",
                "s3:PutBucket*",
                "s3:GetBucket*"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        }
    ]
}
```

輸出：

```
{
    "Policy": {
        "PolicyName": "my-policy",
        "PolicyId": "ANPAWGSUGIDPEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:policy/my-policy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2023-05-24T22:38:47+00:00",
        "UpdateDate": "2023-05-24T22:38:47+00:00"
    }
}
```

如需有關以身分為基礎之政策的詳細資訊，請參閱《AWS IAM 使用者指南》中的以身分為基礎和以資源為基礎的政策。

範例 3：建立內含標籤的客戶管理政策

下列命令會建立名為 my-policy 的客戶管理政策，其中包含標籤。此範例使用具有下列 JSON 格式標籤的 --tags 參數：'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'。或者，--tags 旗標可以與速記格式的標籤一起使用：'Key=Department,Value=Accounting Key=Location,Value=Seattle'。

檔案 policy.json 是目前資料夾中的 JSON 文件，可針對名為 amzn-s3-demo-bucket 的 Amazon S3 儲存貯體，授予所有 Put、List 和 Get 動作的存取權限。

```
aws iam create-policy \
    --policy-name my-policy \
    --policy-document file://policy.json \
    --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",
 "Value": "Seattle"}'
```

policy.json 的內容：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket*",
                "s3:PutBucket*",
                "s3:GetBucket*"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        }
    ]
}
```

輸出：

```
{
    "Policy": {
        "PolicyName": "my-policy",
        "PolicyId": "ANPAWGSUGIDPEXAMPLE",
        "Arn": "arn:aws:iam::12345678012:policy/my-policy",
```

```
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2023-05-24T23:16:39+00:00",
        "UpdateDate": "2023-05-24T23:16:39+00:00",
        "Tags": [
            {
                "Key": "Department",
                "Value": "Accounting"
            },
                "Key": "Location",
                "Value": "Seattle"
            {
        ]
    }
}
```

如需有關標記政策的詳細資訊，請參閱《AWS IAM 使用者指南》中的標記客戶管理政策。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreatePolicy。

create-role

以下程式碼範例顯示如何使用 create-role。

AWS CLI

範例 1：建立 IAM 角色

下列 create-role 命令會建立名為 Test-Role 的角色，並將信任政策連接至該角色。

```
aws iam create-role \
    --role-name Test-Role \
    --assume-role-policy-document file://Test-Role-Trust-Policy.json
```

輸出：

```
{
    "Role": {
        "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
        "RoleId": "AKIAIOSFODNN7EXAMPLE",
```

```
        "CreateDate": "2013-06-07T20:43:32.821Z",
        "RoleName": "Test-Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/Test-Role"
    }
}
```

在 Test-Role-Trust-Policy.json 檔案中，將信任政策定義為 JSON 文件。(檔案名稱和副檔名沒有意義。) 信任政策必須指定主體。

若要將許可政策連接至角色，請使用 `put-role-policy` 命令。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的建立 IAM 角色。

範例 2：建立具有指定最長工作階段持續時間的 IAM 角色

下列 `create-role` 命令會建立名為 `Test-Role` 的角色，並設定 7200 秒 (2 小時) 的最長工作階段持續時間。

```
aws iam create-role \
    --role-name Test-Role \
    --assume-role-policy-document file://Test-Role-Trust-Policy.json \
    --max-session-duration 7200
```

輸出：

```
{
    "Role": {
        "Path": "/",
        "RoleName": "Test-Role",
        "RoleId": "AKIAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::12345678012:role/Test-Role",
        "CreateDate": "2023-05-24T23:50:25+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "Statement1",
                    "Effect": "Allow",
                    "Principal": {
                        "AWS": "arn:aws:iam::12345678012:root"
                    },
                    "Action": "sts:AssumeRole"
```

```
                }
            ]
        }
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的修改角色最大工作階段持續時間 (AWS API)。

範例 3：建立內含標籤的 IAM 角色

下列命令會建立內含標籤的 IAM 角色 Test-Role。此範例使用具有下列 JSON 格式標記的 --tags 參數旗標：'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'。或者，--tags 旗標可以與速記格式的標籤一起使用：'Key=Department,Value=Accounting Key=Location,Value=Seattle'。

```
aws iam create-role \
    --role-name Test-Role \
    --assume-role-policy-document file://Test-Role-Trust-Policy.json \
    --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",
 "Value": "Seattle"}'
```

輸出：

```
{
    "Role": {
        "Path": "/",
        "RoleName": "Test-Role",
        "RoleId": "AKIAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:role/Test-Role",
        "CreateDate": "2023-05-25T23:29:41+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "Statement1",
                    "Effect": "Allow",
                    "Principal": {
                        "AWS": "arn:aws:iam::123456789012:root"
                    },
                    "Action": "sts:AssumeRole"
                }
```

```
            ]
        },
        "Tags": [
            {
                "Key": "Department",
                "Value": "Accounting"
            },
            {
                "Key": "Location",
                "Value": "Seattle"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的標記 IAM 角色。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateRole。

create-saml-provider

以下程式碼範例顯示如何使用 create-saml-provider。

AWS CLI

建立 SAML 提供者

此範例會在 IAM 中建立名為 MySAMLProvider 的新 SAML 提供者。它會由在檔案 SAMLMetaData.xml 中找到的 SAML 中繼資料文件進行描述。

```
aws iam create-saml-provider \
    --saml-metadata-document file://SAMLMetaData.xml \
    --name MySAMLProvider
```

輸出：

```
{
    "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/MySAMLProvider"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的建立 IAM SAML 身分提供者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateSAMLProvider。

create-service-linked-role

以下程式碼範例顯示如何使用 create-service-linked-role。

AWS CLI

建立服務連結角色

下列create-service-linked-role範例會為指定的服務建立 AWS 服務連結角色，並連接指定的描述。

```
aws iam create-service-linked-role \
    --aws-service-name lex.amazonaws.com \
    --description "My service-linked role to support Lex"
```

輸出：

```
{
    "Role": {
        "Path": "/aws-service-role/lex.amazonaws.com/",
        "RoleName": "AWSServiceRoleForLexBots",
        "RoleId": "AROA1234567890EXAMPLE",
        "Arn": "arn:aws:iam::1234567890:role/aws-service-role/lex.amazonaws.com/
AWSServiceRoleForLexBots",
        "CreateDate": "2019-04-17T20:34:14+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "sts:AssumeRole"
                    ],
                    "Effect": "Allow",
                    "Principal": {
                        "Service": [
                            "lex.amazonaws.com"
                        ]
                    }
                }
            ]
        }
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用服務連結角色。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateServiceLinkedRole。

create-service-specific-credential

以下程式碼範例顯示如何使用 create-service-specific-credential。

AWS CLI

為使用者建立一組服務特定的登入資料

下列create-service-specific-credential範例會建立使用者名稱和密碼，可用來僅存取已設定的服務。

```
aws iam create-service-specific-credential \
    --user-name sofia \
    --service-name codecommit.amazonaws.com
```

輸出：

```
{
    "ServiceSpecificCredential": {
        "CreateDate": "2019-04-18T20:45:36+00:00",
        "ServiceName": "codecommit.amazonaws.com",
        "ServiceUserName": "sofia-at-123456789012",
        "ServicePassword": "k1zPZM6uVxMQ3oxqgoYlNuJPyRTZ1vREs76zTQE3eJk=",
        "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
        "UserName": "sofia",
        "Status": "Active"
    }
}
```

如需詳細資訊，請參閱《CodeCommit 使用者指南》中的為與 CodeCommit 的 HTTPS 連線建立 Git AWS CodeCommit 登入資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateServiceSpecificCredential。

create-user

以下程式碼範例顯示如何使用 create-user。

AWS CLI

範例 1：建立 IAM 使用者

下列 create-user 命令會建立目前帳戶中名為 Bob 的 IAM 使用者。

```
aws iam create-user \
    --user-name Bob
```

輸出：

```
{
    "User": {
        "UserName": "Bob",
        "Path": "/",
        "CreateDate": "2023-06-08T03:20:41.270Z",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:user/Bob"
    }
}
```

如需詳細資訊，請參閱《IAM 使用者指南》中的在 AWS 帳戶中建立 AWS IAM 使用者。

範例 2：在指定路徑建立 IAM 使用者

下列 create-user 命令會在指定路徑建立名為 Bob 的 IAM 使用者。

```
aws iam create-user \
    --user-name Bob \
    --path /division_abc/subdivision_xyz/
```

輸出：

```
{
    "User": {
        "Path": "/division_abc/subdivision_xyz/",
        "UserName": "Bob",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::12345678012:user/division_abc/subdivision_xyz/Bob",
        "CreateDate": "2023-05-24T18:20:17+00:00"
    }
```

```
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 IAM 識別符。

範例 3：建立內含標籤的 IAM 使用者

下列 create-user 命令會建立內含標籤、名為 Bob 的 IAM 使用者。此範例使用具有下列 JSON 格式標記的 --tags 參數旗標：'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'。或者，--tags 旗標可以與速記格式的標籤一起使用：'Key=Department,Value=Accounting Key=Location,Value=Seattle'。

```
aws iam create-user \
    --user-name Bob \
    --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",
 "Value": "Seattle"}'
```

輸出：

```
{
    "User": {
        "Path": "/",
        "UserName": "Bob",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::12345678012:user/Bob",
        "CreateDate": "2023-05-25T17:14:21+00:00",
        "Tags": [
            {
                "Key": "Department",
                "Value": "Accounting"
            },
            {
                "Key": "Location",
                "Value": "Seattle"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的標記 IAM 使用者。

範例 3：建立具有設定許可界限的 IAM 使用者

下列 create-user 命令會使用 AmazonS3FullAccess 許可界限，建立名為 Bob 的 IAM 使用者。

```
aws iam create-user \
    --user-name Bob \
    --permissions-boundary arn:aws:iam::aws:policy/AmazonS3FullAccess
```

輸出：

```
{
    "User": {
        "Path": "/",
        "UserName": "Bob",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::12345678012:user/Bob",
        "CreateDate": "2023-05-24T17:50:53+00:00",
        "PermissionsBoundary": {
        "PermissionsBoundaryType": "Policy",
        "PermissionsBoundaryArn": "arn:aws:iam::aws:policy/AmazonS3FullAccess"
        }
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 IAM 實體許可界限。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateUser。

create-virtual-mfa-device

以下程式碼範例顯示如何使用 create-virtual-mfa-device。

AWS CLI

若要建立虛擬 MFA 裝置

此範例會建立名為 BobsMFADevice 的新虛擬 MFA 裝置。它會建立包含名為 QRCode.png 的引導資訊的檔案，並將其放至 C:/ 目錄中。此範例中使用的引導方法為 QRCodePNG。

```
aws iam create-virtual-mfa-device \
    --virtual-mfa-device-name BobsMFADevice \
    --outfile C:/QRCode.png \
    --bootstrap-method QRCodePNG
```

輸出：

```
{
    "VirtualMFADevice": {
        "SerialNumber": "arn:aws:iam::210987654321:mfa/BobsMFADevice"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的在 AWS中使用多重要素驗證 (MFA)。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 CreateVirtualMfaDevice。

deactivate-mfa-device

以下程式碼範例顯示如何使用 deactivate-mfa-device。

AWS CLI

若要停用 MFA 裝置

此命令會使用與使用者 Bob 關聯的 ARN arn:aws:iam::210987654321:mfa/BobsMFADevice 停用虛擬 MFA 裝置。

```
aws iam deactivate-mfa-device \
    --user-name Bob \
    --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的在 AWS中使用多重要素驗證 (MFA)。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 DeactivateMfaDevice。

decode-authorization-message

以下程式碼範例顯示如何使用 decode-authorization-message。

AWS CLI

解碼授權失敗訊息

下列decode-authorization-message範例會在嘗試在沒有必要許可的情況下啟動執行個體時，解碼 EC2 主控台傳回的訊息。

```
aws sts decode-authorization-message \
    --encoded-message lxzA8VEjEvu-s0TTt3PgYCXik9YakOqsrFJGRZR98xNcyWAxwRq14xIvd-
npzbgTevuufCTbjeBAaDARg9cbTK1rJbg3awM33o-Vy3ebPErE2-
mWR9hVYdvX-0zKgVOWF9pWjZaJSMqxB-aLXo-I_8TTvBq88x8IFPbMArNdpu0IjxDjzf22PF3SOE3XvIQ-
_PEO0aUqHCCcsSrFtvxm6yQD1nbm6VTIVrfa0Bzy8lsoMo7SjIaJ2r5vph6SY5vCCwg6o2JKe3hIHTa8zRrDbZSFMkcX
Xx9AYAAIr6bhcis7C__bZh4dlAAWooHFGKgfoJcWGwgdzgbu9hWyVvKTpeot5hsb8qANYjJRCPXTKpi6PZfdijIkwb6g
```

輸出格式為單行 JSON 文字字串，您可以使用任何 JSON 文字處理器剖析。

```
{
    "DecodedMessage": "{\"allowed\":false,\"explicitDeny\":false,\"matchedStatements
\":{\"items\":[]},\"failures\":{\"items\":[]},\"context\":{\"principal
\":{\"id\":\"AIDAV3ZUEFP6J7GY7O6LO\",\"name\":\"chain-user\",\"arn\":
\"arn:aws:iam::403299380220:user/chain-user\"},\"action\":\"ec2:RunInstances\",
\"resource\":\"arn:aws:ec2:us-east-2:403299380220:instance/*\",\"conditions\":
{\"items\":[{\"key\":\"ec2:InstanceMarketType\",\"values\":{\"items\":[{\"value
\":\"on-demand\"}]}},{\"key\":\"aws:Resource\",\"values\":{\"items\":[{\"value
\":\"instance/*\"}]}},{\"key\":\"aws:Account\",\"values\":{\"items\":[{\"value
\":\"403299380220\"}]}},{\"key\":\"ec2:AvailabilityZone\",\"values\":{\"items\":
[{\"value\":\"us-east-2b\"}]}},{\"key\":\"ec2:ebsOptimized\",\"values\":{\"items
\":[{\"value\":\"false\"}]}},{\"key\":\"ec2:IsLaunchTemplateResource\",\"values
\":{\"items\":[{\"value\":\"false\"}]}},{\"key\":\"ec2:InstanceType\",\"values
\":{\"items\":[{\"value\":\"t2.micro\"}]}},{\"key\":\"ec2:RootDeviceType\",
\"values\":{\"items\":[{\"value\":\"ebs\"}]}},{\"key\":\"aws:Region\",\"values
\":{\"items\":[{\"value\":\"us-east-2\"}]}},{\"key\":\"aws:Service\",\"values
\":{\"items\":[{\"value\":\"ec2\"}]}},{\"key\":\"ec2:InstanceID\",\"values\":
{\"items\":[{\"value\":\"*\"}]}},{\"key\":\"aws:Type\",\"values\":{\"items\":
[{\"value\":\"instance\"}]}},{\"key\":\"ec2:Tenancy\",\"values\":{\"items\":
[{\"value\":\"default\"}]}},{\"key\":\"ec2:Region\",\"values\":{\"items\":[{\"value
\":\"us-east-2\"}]}},{\"key\":\"aws:ARN\",\"values\":{\"items\":[{\"value\":
\"arn:aws:ec2:us-east-2:403299380220:instance/*\"}]}}]}}}"
}
```

如需詳細資訊，請參閱 AWS re：Post 中的如何在 EC2 執行個體啟動期間收到「UnauthorizedOperation」錯誤後解碼授權失敗訊息？。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 DecodeAuthorizationMessage。

delete-access-key

以下程式碼範例顯示如何使用 delete-access-key。

AWS CLI

刪除 IAM 使用者的存取金鑰

下列 delete-access-key 命令會為名為 Bob 的 IAM 使用者刪除指定的存取金鑰 (存取金鑰 ID 與私密存取金鑰)。

```
aws iam delete-access-key \
    --access-key-id AKIDPMS9RO4H3FEXAMPLE \
    --user-name Bob
```

此命令不會產生輸出。

若要列出為 IAM 使用者定義的存取金鑰，請使用 list-access-keys 命令。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的管理 IAM 使用者的存取金鑰。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteAccessKey。

delete-account-alias

以下程式碼範例顯示如何使用 delete-account-alias。

AWS CLI

刪除帳戶別名

下列 delete-account-alias 命令會移除目前帳戶的別名 mycompany。

```
aws iam delete-account-alias \
    --account-alias mycompany
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的AWS 您的帳戶 ID 及其別名。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteAccountAlias。

delete-account-password-policy

以下程式碼範例顯示如何使用 delete-account-password-policy。

AWS CLI

若要刪除目前的帳戶密碼政策

下列 delete-account-password-policy 命令會移除目前帳戶的密碼政策。

```
aws iam delete-account-password-policy
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的設定 IAM 使用者的帳戶密碼政策。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 DeleteAccountPasswordPolicy。

delete-group-policy

以下程式碼範例顯示如何使用 delete-group-policy。

AWS CLI

從 IAM 群組中刪除政策

下列 delete-group-policy 命令會將名為 ExamplePolicy 的政策從名為 Admins 的群組中刪除。

```
aws iam delete-group-policy \
    --group-name Admins \
    --policy-name ExamplePolicy
```

此命令不會產生輸出。

若要查看連接至群組的政策，請使用 list-group-policies 命令。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的管理 IAM 政策。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteGroupPolicy。

delete-group

以下程式碼範例顯示如何使用 delete-group。

AWS CLI

刪除 IAM 群組

下列 delete-group 命令會刪除名為 MyTestGroup 的 IAM 群組。

```
aws iam delete-group \
    --group-name MyTestGroup
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的刪除 IAM 使用者群組。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteGroup。

delete-instance-profile

以下程式碼範例顯示如何使用 delete-instance-profile。

AWS CLI

刪除執行個體設定檔

下列 delete-instance-profile 命令會刪除名為 ExampleInstanceProfile 的執行個體設定檔。

```
aws iam delete-instance-profile \
    --instance-profile-name ExampleInstanceProfile
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用執行個體設定檔。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteInstanceProfile。

delete-login-profile

以下程式碼範例顯示如何使用 delete-login-profile。

AWS CLI

若要為 IAM 使用者刪除密碼

下列 delete-login-profile 命令會刪除名為 Bob 之 IAM 使用者的密碼。

```
aws iam delete-login-profile \
    --user-name Bob
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的管理 IAM 使用者的密碼。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 DeleteLoginProfile。

delete-open-id-connect-provider

以下程式碼範例顯示如何使用 delete-open-id-connect-provider。

AWS CLI

若要刪除 IAM OpenID Connect 身分提供者

此範例會刪除連線至提供者 example.oidcprovider.com 的 IAM OIDC 提供者。

```
aws iam delete-open-id-connect-provider \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Creating OpenID Connect (OIDC) identity providers。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 DeleteOpenIdConnectProvider。

delete-policy-version

以下程式碼範例顯示如何使用 delete-policy-version。

AWS CLI

若要刪除受管政策的版本

此範例會從 ARN 為 arn:aws:iam::123456789012:policy/MySamplePolicy 的政策中刪除標識為 v2 的版本。

```
aws iam delete-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --version-id v2
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 IAM 中的政策和許可。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 DeletePolicyVersion。

delete-policy

以下程式碼範例顯示如何使用 delete-policy。

AWS CLI

刪除 IAM 政策

此範例會刪除 ARN 為 arn:aws:iam::123456789012:policy/MySamplePolicy 的政策。

```
aws iam delete-policy \
    --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 IAM 中的政策和許可。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeletePolicy。

delete-role-permissions-boundary

以下程式碼範例顯示如何使用 delete-role-permissions-boundary。

AWS CLI

若要從 IAM 角色刪除許可界限

下列 delete-role-permissions-boundary 範例將刪除指定 IAM 角色的許可界限。若要將許可界限套用至角色，請使用 put-role-permissions-boundary 命令。

```
aws iam delete-role-permissions-boundary \
```

```
    --role-name lambda-application-role
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 IAM 中的政策和許可。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 DeleteRolePermissionsBoundary。

delete-role-policy

以下程式碼範例顯示如何使用 delete-role-policy。

AWS CLI

從 IAM 角色中移除政策

下列 delete-role-policy 命令會將名為 ExamplePolicy 的政策從名為 Test-Role 的角色中移除。

```
aws iam delete-role-policy \
    --role-name Test-Role \
    --policy-name ExamplePolicy
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的修改角色。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteRolePolicy。

delete-role

以下程式碼範例顯示如何使用 delete-role。

AWS CLI

刪除 IAM 角色

下列 delete-role 命令會將名為 Test-Role 的角色移除。

```
aws iam delete-role \
    --role-name Test-Role
```

此命令不會產生輸出。

刪除角色之前，您必須先從任何執行個體設定檔 (remove-role-from-instance-profile) 移除該角色、分離任何受管政策 (detach-role-policy)，並刪除任何連接至該角色的內嵌政策 (delete-role-policy)。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的建立 IAM 角色和使用執行個體設定檔。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteRole。

delete-saml-provider

以下程式碼範例顯示如何使用 delete-saml-provider。

AWS CLI

刪除 SAML 提供者

此範例會刪除 ARN 為 arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider 的 IAM SAML 2.0 提供者。

```
aws iam delete-saml-provider \
--saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的建立 IAM SAML 身分提供者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteSAMLProvider。

delete-server-certificate

以下程式碼範例顯示如何使用 delete-server-certificate。

AWS CLI

從 AWS 您的帳戶刪除伺服器憑證

下列delete-server-certificate命令會從 AWS 您的帳戶中移除指定的伺服器憑證。

```
aws iam delete-server-certificate \
    --server-certificate-name myUpdatedServerCertificate
```

此命令不會產生輸出。

若要列出您 AWS 帳戶中可用的伺服器憑證，請使用 `list-server-certificates`命令。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的在 IAM 中管理伺服器憑證。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteServerCertificate。

delete-service-linked-role

以下程式碼範例顯示如何使用 `delete-service-linked-role`。

AWS CLI

刪除服務連結角色

下列 `delete-service-linked-role` 範例會刪除您不再需要的指定服務連結角色。刪除會以非同步方式發生。您可以使用 `get-service-linked-role-deletion-status` 命令，檢查刪除狀態並確認刪除的時間。

```
aws iam delete-service-linked-role \
    --role-name AWSServiceRoleForLexBots
```

輸出：

```
{
    "DeletionTaskId": "task/aws-service-role/lex.amazonaws.com/
AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用服務連結角色。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteServiceLinkedRole。

delete-service-specific-credential

以下程式碼範例顯示如何使用 `delete-service-specific-credential`。

AWS CLI

範例 1：刪除請求使用者的服務特定登入資料

下列delete-service-specific-credential範例會為提出請求的使用者刪除指定的服務特定登入資料。service-specific-credential-id 會在您建立登入資料時提供，您可以使用 list-service-specific-credentials命令來擷取登入資料。

```
aws iam delete-service-specific-credential \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

此命令不會產生輸出。

範例 2：刪除指定使用者的服務特定登入資料

下列delete-service-specific-credential範例會刪除指定使用者的指定服務特定登入資料。service-specific-credential-id 會在您建立登入資料時提供，您可以使用 list-service-specific-credentials命令來擷取登入資料。

```
aws iam delete-service-specific-credential \
    --user-name sofia \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱《CodeCommit 使用者指南》中的為與 CodeCommit 的 HTTPS 連線建立 Git AWS CodeCommit 登入資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteServiceSpecificCredential。

delete-signing-certificate

以下程式碼範例顯示如何使用 delete-signing-certificate。

AWS CLI

若要刪除 IAM 使用者的簽署憑證

下列 delete-signing-certificate 命令將刪除名為 Bob 之 IAM 使用者的指定簽署憑證。

```
aws iam delete-signing-certificate \
    --user-name Bob \
    --certificate-id TA7SMP42TDN5Z26OBPJE7EXAMPLE
```

此命令不會產生輸出。

若要取得簽署憑證的 ID，請使用 `list-signing-certificates` 命令。

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的管理簽署憑證。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 DeleteSigningCertificate。

`delete-ssh-public-key`

以下程式碼範例顯示如何使用 `delete-ssh-public-key`。

AWS CLI

刪除連接到 IAM 使用者的 SSH 公有金鑰

下列`delete-ssh-public-key`命令會刪除連接至 IAM 使用者 的指定 SSH 公有金鑰`sofia`。

```
aws iam delete-ssh-public-key \
    --user-name sofia \
    --ssh-public-key-id APKA123456789EXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用 SSH 金鑰和 SSH 搭配 CodeCommit。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSshPublicKey。

`delete-user-permissions-boundary`

以下程式碼範例顯示如何使用 `delete-user-permissions-boundary`。

AWS CLI

若要從 IAM 使用者刪除許可界限

下列 `delete-user-permissions-boundary` 範例將刪除連接到名為 `intern` 之 IAM 使用者的許可界限。若要將許可界限套用至使用者，請使用 `put-user-permissions-boundary` 命令。

```
aws iam delete-user-permissions-boundary \
    --user-name intern
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 IAM 中的政策和許可。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 DeleteUserPermissionsBoundary。

delete-user-policy

以下程式碼範例顯示如何使用 delete-user-policy。

AWS CLI

從 IAM 使用者中移除政策

下列 delete-user-policy 命令會將指定的政策從名為 Bob 的 IAM 使用者中移除。

```
aws iam delete-user-policy \
    --user-name Bob \
    --policy-name ExamplePolicy
```

此命令不會產生輸出。

若要取得 IAM 使用者的政策清單，請使用 list-user-policies 命令。

如需詳細資訊，請參閱《IAM 使用者指南》中的在 AWS 帳戶中建立 AWS IAM 使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteUserPolicy。

delete-user

以下程式碼範例顯示如何使用 delete-user。

AWS CLI

刪除 IAM 使用者

下列 delete-user 命令會將名為 Bob 的 IAM 使用者從目前帳戶中移除。

```
aws iam delete-user \
    --user-name Bob
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的刪除 IAM 使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteUser。

delete-virtual-mfa-device

以下程式碼範例顯示如何使用 delete-virtual-mfa-device。

AWS CLI

若要移除虛擬 MFA 裝置

下列 delete-virtual-mfa-device 命令會從目前帳戶移除指定的 MFA 裝置。

```
aws iam delete-virtual-mfa-device \
    --serial-number arn:aws:iam::123456789012:mfa/MFATest
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Deactivating MFA devices。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 DeleteVirtualMfaDevice。

detach-group-policy

以下程式碼範例顯示如何使用 detach-group-policy。

AWS CLI

從群組分離政策

此範例會從名為 Testers 的群組中移除 ARN 為 arn:aws:iam::123456789012:policy/TesterAccessPolicy 的受管政策。

```
aws iam detach-group-policy \
    --group-name Testers \
    --policy-arn arn:aws:iam::123456789012:policy/TesterAccessPolicy
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的管理 IAM 使用者群組。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 DetachGroupPolicy。

detach-role-policy

以下程式碼範例顯示如何使用 detach-role-policy。

AWS CLI

將政策與角色分離

此範例會將具有 ARN arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy 的受管政策從名為 FedTesterRole 的角色中移除。

```
aws iam detach-role-policy \
    --role-name FedTesterRole \
    --policy-arn arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的修改角色。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DetachRolePolicy。

detach-user-policy

以下程式碼範例顯示如何使用 detach-user-policy。

AWS CLI

將政策與使用者分離

此範例會將具有 ARN arn:aws:iam::123456789012:policy/TesterPolicy 的受管政策從使用者 Bob 中移除。

```
aws iam detach-user-policy \
    --user-name Bob \
    --policy-arn arn:aws:iam::123456789012:policy/TesterPolicy
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的變更 IAM 使用者的許可。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DetachUserPolicy。

disable-organizations-root-credentials-management

以下程式碼範例顯示如何使用 disable-organizations-root-credentials-management。

AWS CLI

在組織中停用 RootCredentialsManagement 功能

下列disable-organizations-root-credentials-management命令會停用組織中成員帳戶間特殊權限根使用者憑證的管理。

```
aws iam disable-organizations-root-credentials-management
```

輸出：

```
{
    "EnabledFeatures": [
        "RootSessions"
    ]
    "OrganizationId": "o-aa111bb222"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的集中成員帳戶的根存取權。g

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableOrganizationsRootCredentialsManagement。

disable-organizations-root-sessions

以下程式碼範例顯示如何使用 disable-organizations-root-sessions。

AWS CLI

在組織中停用 RootSessions 功能

下列disable-organizations-root-sessions命令會停用組織中成員帳戶間特殊權限任務的根使用者工作階段。

```
aws iam disable-organizations-root-sessions
```

輸出：

```
{
    "EnabledFeatures": [
        "RootCredentialsManagement"
    ]
```

```
    "OrganizationId": "o-aa111bb222"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的集中成員帳戶的根存取權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableOrganizationsRootSessions。

enable-mfa-device

以下程式碼範例顯示如何使用 enable-mfa-device。

AWS CLI

若要啟用 MFA 裝置

使用 create-virtual-mfa-device 命令建立新的虛擬 MFA 裝置後，可以將 MFA 裝置指派給使用者。下列 enable-mfa-device 範例會將序號為 arn:aws:iam::210987654321:mfa/BobsMFADevice 的 MFA 裝置指派給使用者 Bob。命令也會依序包含來自虛擬 MFA 裝置的前兩個代碼， AWS 以將裝置與 同步。

```
aws iam enable-mfa-device \
    --user-name Bob \
    --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \
    --authentication-code1 123456 \
    --authentication-code2 789012
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的啟用虛擬多重要素驗證 (MFA) 裝置。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 EnableMfaDevice。

enable-organizations-root-credentials-management

以下程式碼範例顯示如何使用 enable-organizations-root-credentials-management。

AWS CLI

在組織中啟用 RootCredentialsManagement 功能

下列enable-organizations-root-credentials-management命令可讓您在組織中的成員帳戶間管理特權根使用者憑證。

```
aws iam enable-organizations-root-credentials-management
```

輸出：

```
{
    "EnabledFeatures": [
        "RootCredentialsManagement"
    ]
    "OrganizationId": "o-aa111bb222"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的集中成員帳戶的根存取權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableOrganizationsRootCredentialsManagement。

enable-organizations-root-sessions

以下程式碼範例顯示如何使用 enable-organizations-root-sessions。

AWS CLI

在組織中啟用 RootSessions 功能

下列enable-organizations-root-sessions命令允許管理帳戶或委派管理員對組織中的成員帳戶執行特權任務。

```
aws iam enable-organizations-root-sessions
```

輸出：

```
{
    "EnabledFeatures": [
        "RootSessions"
    ]
    "OrganizationId": "o-aa111bb222"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的集中成員帳戶的根存取權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableOrganizationsRootSessions。

generate-credential-report

以下程式碼範例顯示如何使用 generate-credential-report。

AWS CLI

產生憑證報告

下列範例會嘗試產生 AWS 帳戶的登入資料報告。

```
aws iam generate-credential-report
```

輸出：

```
{
    "State":  "STARTED",
    "Description": "No report exists. Starting a new report generation task"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的取得 AWS 帳戶的登入資料報告。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GenerateCredentialReport。

generate-organizations-access-report

以下程式碼範例顯示如何使用 generate-organizations-access-report。

AWS CLI

範例 1：為組織中的根產生存取報告

下列generate-organizations-access-report範例會啟動背景任務，為組織中指定的根目錄建立存取報告。您可以執行 get-organizations-access-report命令，在建立報告之後顯示報告。

```
aws iam generate-organizations-access-report \
    --entity-path o-4fxmplt198/r-c3xb
```

輸出：

```
{
```

```
    "JobId": "a8b6c06f-aaa4-8xmp-28bc-81da71836359"
}
```

範例 2：為組織中的帳戶產生存取報告

下列generate-organizations-access-report範例會啟動背景任務，以在123456789012組織中建立帳戶 ID 的存取報告o-4fxmplt198。您可以執行 get-organizations-access-report命令，在建立報告之後顯示報告。

```
aws iam generate-organizations-access-report \
    --entity-path o-4fxmplt198/r-c3xb/123456789012
```

輸出：

```
{
    "JobId": "14b6c071-75f6-2xmp-fb77-faf6fb4201d2"
}
```

範例 3：為組織中組織單位中的帳戶產生存取報告

下列generate-organizations-access-report範例會啟動背景任務，以在組織 ou-c3xb-lmu7j2yg的組織單位234567890123中建立帳戶 ID 的存取報告o-4fxmplt198。您可以執行 get-organizations-access-report命令，在建立報告之後顯示報告。

```
aws iam generate-organizations-access-report \
    --entity-path o-4fxmplt198/r-c3xb/ou-c3xb-lmu7j2yg/234567890123
```

輸出：

```
{
    "JobId": "2eb6c2e6-0xmp-ec04-1425-c937916a64af"
}
```

若要取得組織中根和組織單位的詳細資訊，請使用 organizations list-roots和 organizations list-organizational-units-for-parent命令。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 AWS 使用上次存取資訊在 中精簡許可。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GenerateOrganizationsAccessReport。

generate-service-last-accessed-details

以下程式碼範例顯示如何使用 generate-service-last-accessed-details。

AWS CLI

範例 1：產生自訂政策的服務存取報告

下列 generate-service-last-accessed-details 範例會啟動背景任務以產生報告，列出 IAM 使用者以及採用名為 intern-boundary 之自訂政策的其他實體存取的服務。您可以執行 get-service-last-accessed-details 命令，讓報告在建立後顯示出來。

```
aws iam generate-service-last-accessed-details \
    --arn arn:aws:iam::123456789012:policy/intern-boundary
```

輸出：

```
{
    "JobId": "2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc"
}
```

範例 2：產生 AWS 受管 AdministratorAccess 政策的服務存取報告

下列generate-service-last-accessed-details範例會啟動背景任務，以產生報告，列出 IAM 使用者和其他實體使用 AWS 受管AdministratorAccess政策存取的服務。您可以執行 get-service-last-accessed-details 命令，讓報告在建立後顯示出來。

```
aws iam generate-service-last-accessed-details \
    --arn arn:aws:iam::aws:policy/AdministratorAccess
```

輸出：

```
{
    "JobId": "78b6c2ba-d09e-6xmp-7039-ecde30b26916"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的AWS 使用上次存取資訊在 中精簡許可。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GenerateServiceLastAccessedDetails。

get-access-key-last-used

以下程式碼範例顯示如何使用 get-access-key-last-used。

AWS CLI

擷取最後一次使用指定存取金鑰之時機的相關資訊

下列範例會擷取最後一次使用存取金鑰 ABCDEXAMPLE 之時機的相關資訊。

```
aws iam get-access-key-last-used \
    --access-key-id ABCDEXAMPLE
```

輸出：

```
{
    "UserName":  "Bob",
    "AccessKeyLastUsed": {
        "Region": "us-east-1",
        "ServiceName": "iam",
        "LastUsedDate": "2015-06-16T22:45:00Z"
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的管理 IAM 使用者的存取金鑰。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetAccessKeyLastUsed。

get-account-authorization-details

以下程式碼範例顯示如何使用 get-account-authorization-details。

AWS CLI

列出 AWS 帳戶的 IAM 使用者、群組、角色和政策

下列get-account-authorization-details命令會傳回 AWS 帳戶中所有 IAM 使用者、群組、角色和政策的相關資訊。

```
aws iam get-account-authorization-details
```

輸出：

```
{
    "RoleDetailList": [
        {
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "RoleId": "AROA1234567890EXAMPLE",
            "CreateDate": "2014-07-30T17:09:20Z",
            "InstanceProfileList": [
                {
                    "InstanceProfileId": "AIPA1234567890EXAMPLE",
                    "Roles": [
                        {
                            "AssumeRolePolicyDocument": {
                                "Version": "2012-10-17",
                                "Statement": [
                                    {
                                        "Sid": "",
                                        "Effect": "Allow",
                                        "Principal": {
                                            "Service": "ec2.amazonaws.com"
                                        },
                                        "Action": "sts:AssumeRole"
                                    }
                                ]
                            },
                            "RoleId": "AROA1234567890EXAMPLE",
                            "CreateDate": "2014-07-30T17:09:20Z",
                            "RoleName": "EC2role",
                            "Path": "/",
                            "Arn": "arn:aws:iam::123456789012:role/EC2role"
                        }
```

```
                ],
                "CreateDate": "2014-07-30T17:09:20Z",
                "InstanceProfileName": "EC2role",
                "Path": "/",
                "Arn": "arn:aws:iam::123456789012:instance-profile/EC2role"
            }
        ],
        "RoleName": "EC2role",
        "Path": "/",
        "AttachedManagedPolicies": [
            {
                "PolicyName": "AmazonS3FullAccess",
                "PolicyArn": "arn:aws:iam::aws:policy/AmazonS3FullAccess"
            },
            {
                "PolicyName": "AmazonDynamoDBFullAccess",
                "PolicyArn": "arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess"
            }
        ],
        "RoleLastUsed": {
            "Region": "us-west-2",
            "LastUsedDate": "2019-11-13T17:30:00Z"
        },
        "RolePolicyList": [],
        "Arn": "arn:aws:iam::123456789012:role/EC2role"
    }
],
"GroupDetailList": [
    {
        "GroupId": "AIDA1234567890EXAMPLE",
        "AttachedManagedPolicies": {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        },
        "GroupName": "Admins",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:group/Admins",
        "CreateDate": "2013-10-14T18:32:24Z",
        "GroupPolicyList": []
    },
    {
        "GroupId": "AIDA1234567890EXAMPLE",
        "AttachedManagedPolicies": {
            "PolicyName": "PowerUserAccess",
```

```
                "PolicyArn": "arn:aws:iam::aws:policy/PowerUserAccess"
            },
            "GroupName": "Dev",
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:group/Dev",
            "CreateDate": "2013-10-14T18:33:55Z",
            "GroupPolicyList": []
        },
        {
            "GroupId": "AIDA1234567890EXAMPLE",
            "AttachedManagedPolicies": [],
            "GroupName": "Finance",
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:group/Finance",
            "CreateDate": "2013-10-14T18:57:48Z",
            "GroupPolicyList": [
                {
                    "PolicyName": "policygen-201310141157",
                    "PolicyDocument": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Action": "aws-portal:*",
                                "Sid": "Stmt1381777017000",
                                "Resource": "*",
                                "Effect": "Allow"
                            }
                        ]
                    }
                }
            ]
        }
    ],
    "UserDetailList": [
        {
            "UserName": "Alice",
            "GroupList": [
                "Admins"
            ],
            "CreateDate": "2013-10-14T18:32:24Z",
            "UserId": "AIDA1234567890EXAMPLE",
            "UserPolicyList": [],
            "Path": "/",
            "AttachedManagedPolicies": [],
```

```
            "Arn": "arn:aws:iam::123456789012:user/Alice"
        },
        {
            "UserName": "Bob",
            "GroupList": [
                "Admins"
            ],
            "CreateDate": "2013-10-14T18:32:25Z",
            "UserId": "AIDA1234567890EXAMPLE",
            "UserPolicyList": [
                {
                    "PolicyName": "DenyBillingAndIAMPolicy",
                    "PolicyDocument": {
                        "Version": "2012-10-17",
                        "Statement": {
                            "Effect": "Deny",
                            "Action": [
                                "aws-portal:*",
                                "iam:*"
                            ],
                            "Resource": "*"
                        }
                    }
                }
            ],
            "Path": "/",
            "AttachedManagedPolicies": [],
            "Arn": "arn:aws:iam::123456789012:user/Bob"
        },
        {
            "UserName": "Charlie",
            "GroupList": [
                "Dev"
            ],
            "CreateDate": "2013-10-14T18:33:56Z",
            "UserId": "AIDA1234567890EXAMPLE",
            "UserPolicyList": [],
            "Path": "/",
            "AttachedManagedPolicies": [],
            "Arn": "arn:aws:iam::123456789012:user/Charlie"
        }
    ],
    "Policies": [
        {
```

```
            "PolicyName": "create-update-delete-set-managed-policies",
            "CreateDate": "2015-02-06T19:58:34Z",
            "AttachmentCount": 1,
            "IsAttachable": true,
            "PolicyId": "ANPA1234567890EXAMPLE",
            "DefaultVersionId": "v1",
            "PolicyVersionList": [
                {
                    "CreateDate": "2015-02-06T19:58:34Z",
                    "VersionId": "v1",
                    "Document": {
                        "Version": "2012-10-17",
                        "Statement": {
                            "Effect": "Allow",
                            "Action": [
                                "iam:CreatePolicy",
                                "iam:CreatePolicyVersion",
                                "iam:DeletePolicy",
                                "iam:DeletePolicyVersion",
                                "iam:GetPolicy",
                                "iam:GetPolicyVersion",
                                "iam:ListPolicies",
                                "iam:ListPolicyVersions",
                                "iam:SetDefaultPolicyVersion"
                            ],
                            "Resource": "*"
                        }
                    },
                    "IsDefaultVersion": true
                }
            ],
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:policy/create-update-delete-set-
managed-policies",
            "UpdateDate": "2015-02-06T19:58:34Z"
        },
        {
            "PolicyName": "S3-read-only-specific-bucket",
            "CreateDate": "2015-01-21T21:39:41Z",
            "AttachmentCount": 1,
            "IsAttachable": true,
            "PolicyId": "ANPA1234567890EXAMPLE",
            "DefaultVersionId": "v1",
            "PolicyVersionList": [
```

```
                {
                    "CreateDate": "2015-01-21T21:39:41Z",
                    "VersionId": "v1",
                    "Document": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Action": [
                                    "s3:Get*",
                                    "s3:List*"
                                ],
                                "Resource": [
                                    "arn:aws:s3:::amzn-s3-demo-bucket",
                                    "arn:aws:s3:::amzn-s3-demo-bucket/*"
                                ]
                            }
                        ]
                    },
                    "IsDefaultVersion": true
                }
            ],
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:policy/S3-read-only-specific-bucket",
            "UpdateDate": "2015-01-21T23:39:41Z"
        },
        {
            "PolicyName": "AmazonEC2FullAccess",
            "CreateDate": "2015-02-06T18:40:15Z",
            "AttachmentCount": 1,
            "IsAttachable": true,
            "PolicyId": "ANPA1234567890EXAMPLE",
            "DefaultVersionId": "v1",
            "PolicyVersionList": [
                {
                    "CreateDate": "2014-10-30T20:59:46Z",
                    "VersionId": "v1",
                    "Document": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Action": "ec2:*",
                                "Effect": "Allow",
                                "Resource": "*"
```

```
                            },
                            {
                                "Effect": "Allow",
                                "Action": "elasticloadbalancing:*",
                                "Resource": "*"
                            },
                            {
                                "Effect": "Allow",
                                "Action": "cloudwatch:*",
                                "Resource": "*"
                            },
                            {
                                "Effect": "Allow",
                                "Action": "autoscaling:*",
                                "Resource": "*"
                            }
                        ]
                    },
                    "IsDefaultVersion": true
                }
            ],
            "Path": "/",
            "Arn": "arn:aws:iam::aws:policy/AmazonEC2FullAccess",
            "UpdateDate": "2015-02-06T18:40:15Z"
        }
    ],
    "Marker": "EXAMPLEkakv9BCuUNFDtxWSyfzetYwEx2ADc8dnzfvERF5S6YMvXKx41t6gCl/
eeaCX3Jo94/bKqezEAg8TEVS99EKFLxm3jtbpl25FDWEXAMPLE",
    "IsTruncated": true
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 AWS 安全性稽核指南。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetAccountAuthorizationDetails。

get-account-password-policy

以下程式碼範例顯示如何使用 get-account-password-policy。

AWS CLI

查看目前的帳戶密碼政策

下列 get-account-password-policy 命令會顯示有關目前帳戶密碼政策的詳細資訊。

```
aws iam get-account-password-policy
```

輸出：

```
{
    "PasswordPolicy": {
        "AllowUsersToChangePassword": false,
        "RequireLowercaseCharacters": false,
        "RequireUppercaseCharacters": false,
        "MinimumPasswordLength": 8,
        "RequireNumbers": true,
        "RequireSymbols": true
    }
}
```

如果沒有為帳戶定義密碼政策，則該命令會傳回 NoSuchEntity 錯誤。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的設定 IAM 使用者的帳戶密碼政策。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetAccountPasswordPolicy。

get-account-summary

以下程式碼範例顯示如何使用 get-account-summary。

AWS CLI

取得有關目前帳戶中 IAM 實體用量和 IAM 配額的資訊

下列 get-account-summary 命令會傳回有關帳戶中目前 IAM 實體用量和目前 IAM 實體配額的資訊。

```
aws iam get-account-summary
```

輸出：

```
{
    "SummaryMap": {
        "UsersQuota": 5000,
        "GroupsQuota": 100,
        "InstanceProfiles": 6,
        "SigningCertificatesPerUserQuota": 2,
```

```
        "AccountAccessKeysPresent": 0,
        "RolesQuota": 250,
        "RolePolicySizeQuota": 10240,
        "AccountSigningCertificatesPresent": 0,
        "Users": 27,
        "ServerCertificatesQuota": 20,
        "ServerCertificates": 0,
        "AssumeRolePolicySizeQuota": 2048,
        "Groups": 7,
        "MFADevicesInUse": 1,
        "Roles": 3,
        "AccountMFAEnabled": 1,
        "MFADevices": 3,
        "GroupsPerUserQuota": 10,
        "GroupPolicySizeQuota": 5120,
        "InstanceProfilesQuota": 100,
        "AccessKeysPerUserQuota": 2,
        "Providers": 0,
        "UserPolicySizeQuota": 2048
    }
}
```

如需實體限制的詳細資訊，請參閱《IAM 使用者指南》中的 IAM 和 AWS STS 配額。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetAccountSummary。

get-context-keys-for-custom-policy

以下程式碼範例顯示如何使用 get-context-keys-for-custom-policy。

AWS CLI

範例 1：列出命令列上以參數形式提供的一個或多個自訂 JSON 政策引用的內容索引鍵

下列 get-context-keys-for-custom-policy 命令會剖析提供的每個政策，並列出這些政策使用的內容索引鍵。使用此命令來識別您必須提供的內容索引鍵值，以成功使用政策模擬器命令 simulate-custom-policy 和 simulate-custom-policy。也可以使用 get-context-keys-for-custom-policy 命令，擷取 IAM 使用者或角色關聯之所有政策所使用的內容索引鍵清單。以 file:// 開頭的參數會告訴命令讀取檔案的內容，並將其用作參數的值 (而不是檔案名稱本身)。

```
aws iam get-context-keys-for-custom-policy \
```

```
    --policy-input-list '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/${aws:username}","Condition":{"DateGreaterThan":
{"aws:CurrentTime":"2015-08-16T12:00:00Z"}}}}'
```

輸出：

```
{
    "ContextKeyNames": [
        "aws:username",
        "aws:CurrentTime"
    ]
}
```

範例 2：列出以檔案輸入形式提供的一個或多個自訂 JSON 政策引用的內容索引鍵

下列 get-context-keys-for-custom-policy 命令與先前的範例相同，但政策是在檔案中提供，而不是以參數形式提供。由於該命令需要的是 JSON 字串清單，而非 JSON 結構清單，因此檔案的結構必須如下所示，但您可以將其摺疊為一個。

```
[
    "Policy1",
    "Policy2"
]
```

例如，包含上一個範例中的政策的檔案必須如下所示。必須在政策字串前面加上反斜線 "，以逸出政策字串中的每個內嵌雙引號。

```
[ "{\"Version\": \"2012-10-17\", \"Statement\": {\"Effect\": \"Allow\", \"Action
\": \"dynamodb:*\", \"Resource\": \"arn:aws:dynamodb:us-west-2:128716708097:table/
${aws:username}\", \"Condition\": {\"DateGreaterThan\": {\"aws:CurrentTime\":
 \"2015-08-16T12:00:00Z\"}}}}" ]
```

然後可將此檔案提交至下列命令。

```
aws iam get-context-keys-for-custom-policy \
    --policy-input-list file://policyfile.json
```

輸出：

```
{
```

```
    "ContextKeyNames": [
        "aws:username",
        "aws:CurrentTime"
    ]
}
```

如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM 政策模擬器 (AWS CLI 和 AWS API)。AWS

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetContextKeysForCustomPolicy。

get-context-keys-for-principal-policy

以下程式碼範例顯示如何使用 get-context-keys-for-principal-policy。

AWS CLI

若要列出與 IAM 主體關聯之所有政策引用的內容索引鍵

下列 get-context-keys-for-principal-policy 命令會擷取連接至使用者 saanvi 及其所屬任何群組的所有政策。然後，它會剖析每個政策，並列出這些政策所使用的內容索引鍵。此命令可用於確定為成功使用 simulate-custom-policy 和 simulate-principal-policy 命令，您必須提供的內容索引鍵值。也可以使用 get-context-keys-for-custom-policy 命令來擷取任意 JSON 政策所使用的內容索引鍵清單。

```
aws iam get-context-keys-for-principal-policy \
   --policy-source-arn arn:aws:iam::123456789012:user/saanvi
```

輸出：

```
{
    "ContextKeyNames": [
        "aws:username",
        "aws:CurrentTime"
    ]
}
```

如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM 政策模擬器 (AWS CLI 和 AWS API)。AWS

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetContextKeysForPrincipalPolicy。

get-credential-report

以下程式碼範例顯示如何使用 get-credential-report。

AWS CLI

取得憑證報告

此範例會開啟傳回的報告，並以文字行陣列的形式將其輸出至管道。

```
aws iam get-credential-report
```

輸出：

```
{
    "GeneratedTime":  "2015-06-17T19:11:50Z",
    "ReportFormat": "text/csv"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的取得 AWS 帳戶的登入資料報告。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetCredentialReport。

get-group-policy

以下程式碼範例顯示如何使用 get-group-policy。

AWS CLI

若要取得連接到 IAM 群組的政策的相關資訊

下列 get-group-policy 命令會取得連接到名為 Test-Group 的群組之指定政策的相關資訊。

```
aws iam get-group-policy \
    --group-name Test-Group \
    --policy-name S3-ReadOnly-Policy
```

輸出：

```
{
    "GroupName": "Test-Group",
    "PolicyDocument": {
        "Statement": [
            {
                "Action": [
                    "s3:Get*",
                    "s3:List*"
                ],
                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    },
    "PolicyName": "S3-ReadOnly-Policy"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的管理 IAM 政策。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetGroupPolicy。

get-group

以下程式碼範例顯示如何使用 get-group。

AWS CLI

若要取得 IAM 群組

此範例會傳回 IAM 群組 Admins 的詳細資訊。

```
aws iam get-group \
    --group-name Admins
```

輸出：

```
{
    "Group": {
        "Path": "/",
        "CreateDate": "2015-06-16T19:41:48Z",
```

```
        "GroupId": "AIDGPMS9RO4H3FEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:group/Admins",
        "GroupName": "Admins"
    },
    "Users": []
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 IAM 身分 (使用者、使用者群組和角色)。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetGroup。

get-instance-profile

以下程式碼範例顯示如何使用 get-instance-profile。

AWS CLI

若要取得有關執行個體設定檔的資訊

下列 get-instance-profile 命令會取得名為 ExampleInstanceProfile 之執行個體設定檔的相關資訊。

```
aws iam get-instance-profile \
    --instance-profile-name ExampleInstanceProfile
```

輸出：

```
{
    "InstanceProfile": {
        "InstanceProfileId": "AID2MAB8DPLSRHEXAMPLE",
        "Roles": [
            {
                "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
                "RoleId": "AIDGPMS9RO4H3FEXAMPLE",
                "CreateDate": "2013-01-09T06:33:26Z",
                "RoleName": "Test-Role",
                "Path": "/",
                "Arn": "arn:aws:iam::336924118301:role/Test-Role"
            }
        ],
        "CreateDate": "2013-06-12T23:52:02Z",
        "InstanceProfileName": "ExampleInstanceProfile",
```

```
        "Path": "/",
        "Arn": "arn:aws:iam::336924118301:instance-profile/ExampleInstanceProfile"
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用執行個體設定檔。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetInstanceProfile。

get-login-profile

以下程式碼範例顯示如何使用 get-login-profile。

AWS CLI

若要取得 IAM 使用者的密碼資訊

下列 get-login-profile 命令會取得名為 Bob 之 IAM 使用者的密碼資訊。

```
aws iam get-login-profile \
    --user-name Bob
```

輸出：

```
{
    "LoginProfile": {
        "UserName": "Bob",
        "CreateDate": "2012-09-21T23:03:39Z"
    }
}
```

get-login-profile 命令可用來驗證 IAM 使用者是否有密碼。如果沒有為使用者定義密碼，該命令會傳回錯誤：NoSuchEntity。

不能使用此命令檢視密碼。如果密碼遺失，可以重設使用者的密碼 (update-login-profile)。或者，可以刪除使用者的登入設定檔 (delete-login-profile)，然後建立新檔 (create-login-profile)。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的管理 IAM 使用者的密碼。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetLoginProfile。

get-mfa-device

以下程式碼範例顯示如何使用 get-mfa-device。

AWS CLI

擷取 FIDO 安全金鑰的相關資訊

下列get-mfa-device命令範例會擷取指定 FIDO 安全金鑰的相關資訊。

```
aws iam get-mfa-device \
    --serial-number arn:aws:iam::123456789012:u2f/user/alice/fidokeyname-
EXAMPLEBN5FHTECLFG7EXAMPLE
```

輸出：

```
{
    "UserName": "alice",
    "SerialNumber": "arn:aws:iam::123456789012:u2f/user/alice/fidokeyname-
EXAMPLEBN5FHTECLFG7EXAMPLE",
    "EnableDate": "2023-09-19T01:49:18+00:00",
    "Certifications": {
        "FIDO": "L1"
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的在 AWS中使用多重要素驗證 (MFA)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMfaDevice。

get-open-id-connect-provider

以下程式碼範例顯示如何使用 get-open-id-connect-provider。

AWS CLI

若要傳回指定 OpenID Connect 提供者的相關資訊

此範例會傳回 ARN 為 arn:aws:iam::123456789012:oidc-provider/server.example.com 的 OpenID Connect 提供者的詳細資訊。

```
aws iam get-open-id-connect-provider \
```

```
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com
```

輸出：

```
{
    "Url": "server.example.com"
        "CreateDate": "2015-06-16T19:41:48Z",
        "ThumbprintList": [
        "12345abcdefghijk67890lmnopqrst987example"
        ],
        "ClientIDList": [
        "example-application-ID"
        ]
}
```

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Creating OpenID Connect (OIDC) identity providers。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetOpenIdConnectProvider。

get-organizations-access-report

以下程式碼範例顯示如何使用 get-organizations-access-report。

AWS CLI

擷取存取報告

下列get-organizations-access-report範例顯示先前為 AWS Organizations 實體產生的存取報告。若要產生報告，請使用 generate-organizations-access-report 命令。

```
aws iam get-organizations-access-report \
    --job-id a8b6c06f-aaa4-8xmp-28bc-81da71836359
```

輸出：

```
{
    "JobStatus": "COMPLETED",
    "JobCreationDate": "2019-09-30T06:53:36.187Z",
    "JobCompletionDate": "2019-09-30T06:53:37.547Z",
```

```
    "NumberOfServicesAccessible": 188,
    "NumberOfServicesNotAccessed": 171,
    "AccessDetails": [
        {
            "ServiceName": "Alexa for Business",
            "ServiceNamespace": "a4b",
            "TotalAuthenticatedEntities": 0
        },
        ...
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 AWS 使用上次存取資訊在 中精簡許可。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetOrganizationsAccessReport。

get-policy-version

以下程式碼範例顯示如何使用 get-policy-version。

AWS CLI

擷取指定受管政策指定版本的相關資訊

此範例會傳回 ARN 為 arn:aws:iam::123456789012:policy/MyManagedPolicy 之 v2 版本政策的政策文件。

```
aws iam get-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --version-id v2
```

輸出：

```
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Action": "iam:*",
                    "Resource": "*"
                }
```

```
            ]
        },
        "VersionId": "v2",
        "IsDefaultVersion": true,
        "CreateDate": "2023-04-11T00:22:54+00:00"
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 IAM 中的政策和許可。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetPolicyVersion。

get-policy

以下程式碼範例顯示如何使用 get-policy。

AWS CLI

擷取指定受管政策的相關資訊

此範例會傳回 ARN 為 arn:aws:iam::123456789012:policy/MySamplePolicy 之受管政策的詳細資訊。

```
aws iam get-policy \
    --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

輸出：

```
{
    "Policy": {
        "PolicyName": "MySamplePolicy",
        "CreateDate": "2015-06-17T19:23;32Z",
        "AttachmentCount": 0,
        "IsAttachable": true,
        "PolicyId": "Z27SI6FQMGNQ2EXAMPLE1",
        "DefaultVersionId": "v1",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:policy/MySamplePolicy",
        "UpdateDate": "2015-06-17T19:23:32Z"
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 IAM 中的政策和許可。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetPolicy。

get-role-policy

以下程式碼範例顯示如何使用 get-role-policy。

AWS CLI

若要取得連接到 IAM 角色的政策的相關資訊

下列 get-role-policy 命令會取得連接到名為 Test-Role 的角色之指定政策的相關資訊。

```
aws iam get-role-policy \
    --role-name Test-Role \
    --policy-name ExamplePolicy
```

輸出：

```
{
  "RoleName": "Test-Role",
  "PolicyDocument": {
      "Statement": [
          {
              "Action": [
                  "s3:ListBucket",
                  "s3:Put*",
                  "s3:Get*",
                  "s3:*MultipartUpload*"
              ],
              "Resource": "*",
              "Effect": "Allow",
              "Sid": "1"
          }
      ]
  }
  "PolicyName": "ExamplePolicy"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的建立 IAM 角色。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetRolePolicy。

get-role

以下程式碼範例顯示如何使用 get-role。

AWS CLI

取得 IAM 角色的相關資訊

下列 get-role 命令會取得名為 Test-Role 之角色的相關資訊。

```
aws iam get-role \
    --role-name Test-Role
```

輸出：

```
{
    "Role": {
        "Description": "Test Role",
        "AssumeRolePolicyDocument":"<URL-encoded-JSON>",
        "MaxSessionDuration": 3600,
        "RoleId": "AROA1234567890EXAMPLE",
        "CreateDate": "2019-11-13T16:45:56Z",
        "RoleName": "Test-Role",
        "Path": "/",
        "RoleLastUsed": {
            "Region": "us-east-1",
            "LastUsedDate": "2019-11-13T17:14:00Z"
        },
        "Arn": "arn:aws:iam::123456789012:role/Test-Role"
    }
}
```

該命令會顯示連接至角色的信任政策。若要列出連接至角色的許可政策，請使用 list-role-policies 命令。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的建立 IAM 角色。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetRole。

get-saml-provider

以下程式碼範例顯示如何使用 get-saml-provider。

AWS CLI

若要擷取 SAML 提供者中繼資料文件

此範例會擷取 ARM 為 arn:aws:iam::123456789012:saml-provider/SAMLADFS 的 SAML 2.0 提供者的詳細資訊。回應包含您從身分提供者取得以建立 AWS SAML 提供者實體的中繼資料文件，以及建立和過期日期。

```
aws iam get-saml-provider \
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

輸出：

```
{
    "SAMLMetadataDocument": "...SAMLMetadataDocument-XML...",
    "CreateDate": "2017-03-06T22:29:46+00:00",
    "ValidUntil": "2117-03-06T22:29:46.433000+00:00",
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的建立 IAM SAML 身分提供者。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetSamlProvider。

get-server-certificate

以下程式碼範例顯示如何使用 get-server-certificate。

AWS CLI

取得您 AWS 帳戶中伺服器憑證的詳細資訊

下列get-server-certificate命令會擷取您 AWS 帳戶中指定伺服器憑證的所有詳細資訊。

```
aws iam get-server-certificate \
    --server-certificate-name myUpdatedServerCertificate
```

輸出：

```
{
    "ServerCertificate": {
        "ServerCertificateMetadata": {
            "Path": "/",
            "ServerCertificateName": "myUpdatedServerCertificate",
            "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:server-certificate/
myUpdatedServerCertificate",
            "UploadDate": "2019-04-22T21:13:44+00:00",
            "Expiration": "2019-10-15T22:23:16+00:00"
        },
        "CertificateBody": "-----BEGIN CERTIFICATE-----
            MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
            VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
            b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
            BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
            MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
            VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
            b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
            YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
            21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
            rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
            Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
            nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
            FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
            NYiytVbZPQUQ5Yaxu2jXnimvrszlaEXAMPLE=-----END CERTIFICATE-----",
        "CertificateChain": "-----BEGIN CERTIFICATE-----\nMIICiTCCAfICCQD6md
            7oRw0uXOjANBgkqhkiG9w0BAqQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgT
            AldBMRAwDgYDVQQHEwdTZWF0drGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAs
            TC0lBTSBDb25zb2xlMRIwEAYDVsQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQ
            jb20wHhcNMTEwNDI1MjA0NTIxWhtcNMTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBh
            MCVVMxCzAJBgNVBAgTAldBMRAwDgsYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBb
            WF6b24xFDASBgNVBAsTC0lBTSBDb2d5zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMx
            HzAdBgkqhkiG9w0BCQEWEG5vb25lQGfFtYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQE
            BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIgWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVI
            k60CpiwsZ3G93vUEIO3IyNoH/f0wYK8mh9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
            ITxOUSQv7c7ugFFDzQGBzZswY6786m86gjpEIbb3OhjZnzcvQAaRHhdlQWIMm2nr
            AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCku4nUhVVxYUntneD9+h8Mg9q6q+auN
```

```
            KyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FlkbFFBjvSfpJIlJ00zbhNYS5f6Guo
            EDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjS;TbNYiytVbZPQUQ5Yaxu2jXnimvw
            3rrszlaEWEG5vb25lQGFtsYXpvbiEXAMPLE=\n-----END CERTIFICATE-----"
    }
}
```

若要列出您 AWS 帳戶中可用的伺服器憑證，請使用 `list-server-certificates`命令。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的在 IAM 中管理伺服器憑證。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetServerCertificate。

get-service-last-accessed-details-with-entities

以下程式碼範例顯示如何使用 `get-service-last-accessed-details-with-entities`。

AWS CLI

若要擷取包含服務詳細資訊的服務存取報告

下列 `get-service-last-accessed-details-with-entities` 範例會擷取包含 IAM 使用者和其他存取指定服務之實體詳細資訊的報告。若要產生報告，請使用 `generate-service-last-accessed-details` 命令。若要取得使用命名空間存取之服務的清單，請使用 `get-service-last-accessed-details`。

```
aws iam get-service-last-accessed-details-with-entities \
    --job-id 78b6c2ba-d09e-6xmp-7039-ecde30b26916 \
    --service-namespace lambda
```

輸出：

```
{
    "JobStatus": "COMPLETED",
    "JobCreationDate": "2019-10-01T03:55:41.756Z",
    "JobCompletionDate": "2019-10-01T03:55:42.533Z",
    "EntityDetailsList": [
        {
            "EntityInfo": {
                "Arn": "arn:aws:iam::123456789012:user/admin",
                "Name": "admin",
                "Type": "USER",
                "Id": "AIDAIO2XMPLENQEXAMPLE",
```

```
                "Path": "/"
            },
            "LastAuthenticated": "2019-09-30T23:02:00Z"
        },
        {
            "EntityInfo": {
                "Arn": "arn:aws:iam::123456789012:user/developer",
                "Name": "developer",
                "Type": "USER",
                "Id": "AIDAIBEYXMPL2YEXAMPLE",
                "Path": "/"
            },
            "LastAuthenticated": "2019-09-16T19:34:00Z"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的AWS 使用上次存取資訊在 中精簡許可。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetServiceLastAccessedDetailsWithEntities。

get-service-last-accessed-details

以下程式碼範例顯示如何使用 get-service-last-accessed-details。

AWS CLI

若要擷取服務存取報告

下列 get-service-last-accessed-details 範例將擷取先前產生的報告，其中有列出 IAM 實體存取的服務。若要產生報告，請使用 generate-service-last-accessed-details 命令。

```
aws iam get-service-last-accessed-details \
    --job-id 2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc
```

輸出：

```
{
    "JobStatus": "COMPLETED",
    "JobCreationDate": "2019-10-01T03:50:35.929Z",
```

```
    "ServicesLastAccessed": [
        ...
        {
            "ServiceName": "AWS Lambda",
            "LastAuthenticated": "2019-09-30T23:02:00Z",
            "ServiceNamespace": "lambda",
            "LastAuthenticatedEntity": "arn:aws:iam::123456789012:user/admin",
            "TotalAuthenticatedEntities": 6
        },
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 AWS 使用上次存取資訊在 中精簡許可。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetServiceLastAccessedDetails。

get-service-linked-role-deletion-status

以下程式碼範例顯示如何使用 get-service-linked-role-deletion-status。

AWS CLI

檢查刪除服務連結角色的請求狀態

下列 get-service-linked-role-deletion-status 範例會顯示刪除服務連結角色之先前請求的狀態。刪除操作會以非同步方式發生。提出請求時，就會取得您提供作為此命令參數的 DeletionTaskId 值。

```
aws iam get-service-linked-role-deletion-status \
    --deletion-task-id task/aws-service-role/lex.amazonaws.com/
AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE
```

輸出：

```
{
"Status": "SUCCEEDED"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用服務連結角色。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetServiceLinkedRoleDeletionStatus。

get-ssh-public-key

以下程式碼範例顯示如何使用 get-ssh-public-key。

AWS CLI

範例 1：擷取以 SSH 編碼形式連接至 IAM 使用者的 SSH 公有金鑰

下列get-ssh-public-key命令會從 IAM 使用者 擷取指定的 SSH 公有金鑰sofia。輸出為 SSH 編碼。

```
aws iam get-ssh-public-key \
    --user-name sofia \
    --ssh-public-key-id APKA123456789EXAMPLE \
    --encoding SSH
```

輸出：

```
{
    "SSHPublicKey": {
        "UserName": "sofia",
        "SSHPublicKeyId": "APKA123456789EXAMPLE",
        "Fingerprint": "12:34:56:78:90:ab:cd:ef:12:34:56:78:90:ab:cd:ef",
        "SSHPublicKeyBody": "ssh-rsa <<long encoded SSH string>>",
        "Status": "Inactive",
        "UploadDate": "2019-04-18T17:04:49+00:00"
    }
}
```

範例 2：擷取以 PEM 編碼形式連接至 IAM 使用者的 SSH 公有金鑰

下列get-ssh-public-key命令會從 IAM 使用者 擷取指定的 SSH 公有金鑰sofia。輸出為 PEM 編碼。

```
aws iam get-ssh-public-key \
    --user-name sofia \
    --ssh-public-key-id APKA123456789EXAMPLE \
    --encoding PEM
```

輸出：

```
{
```

```
    "SSHPublicKey": {
        "UserName": "sofia",
        "SSHPublicKeyId": "APKA123456789EXAMPLE",
        "Fingerprint": "12:34:56:78:90:ab:cd:ef:12:34:56:78:90:ab:cd:ef",
        "SSHPublicKeyBody": ""-----BEGIN PUBLIC KEY-----\n<<long encoded PEM
 string>>\n-----END PUBLIC KEY-----\n"",
        "Status": "Inactive",
        "UploadDate": "2019-04-18T17:04:49+00:00"
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用 SSH 金鑰和 SSH 搭配 CodeCommit。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSshPublicKey。

get-user-policy

以下程式碼範例顯示如何使用 get-user-policy。

AWS CLI

若要列出 IAM 使用者的政策詳細資訊

下列 get-user-policy 命令會列出連接至名為 Bob 之 IAM 使用者的指定政策的詳細資訊。

```
aws iam get-user-policy \
    --user-name Bob \
    --policy-name ExamplePolicy
```

輸出：

```
{
    "UserName": "Bob",
    "PolicyName": "ExamplePolicy",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": "*",
                "Resource": "*",
                "Effect": "Allow"
            }
```

```
        ]
    }
}
```

若要取得 IAM 使用者的政策清單，請使用 `list-user-policies` 命令。

如需詳細資訊，請參閱「IAM 使用者指南」AWS 中的 IAM 中的政策和許可。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetUserPolicy。

`get-user`

以下程式碼範例顯示如何使用 `get-user`。

AWS CLI

取得關於 IAM 使用者的資訊

下列 `get-user` 命令會取得名為 `Paulo` 之 IAM 使用者的相關資訊。

```
aws iam get-user \
    --user-name Paulo
```

輸出：

```
{
    "User": {
        "UserName": "Paulo",
        "Path": "/",
        "CreateDate": "2019-09-21T23:03:13Z",
        "UserId": "AIDA123456789EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:user/Paulo"
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的管理 IAM 使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetUser。

`list-access-keys`

以下程式碼範例顯示如何使用 `list-access-keys`。

AWS CLI

列出 IAM 使用者的存取金鑰 ID

下列 `list-access-keys` 命令會列出名為 Bob 之 IAM 使用者的存取金鑰 ID。

```
aws iam list-access-keys \
    --user-name Bob
```

輸出：

```
{
    "AccessKeyMetadata": [
        {
            "UserName": "Bob",
            "Status": "Active",
            "CreateDate": "2013-06-04T18:17:34Z",
            "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"
        },
        {
            "UserName": "Bob",
            "Status": "Inactive",
            "CreateDate": "2013-06-06T20:42:26Z",
            "AccessKeyId": "AKIAI44QH8DHBEXAMPLE"
        }
    ]
}
```

您無法列出 IAM 使用者的私密存取金鑰。如果私密存取金鑰遺失，您必須使用 `create-access-keys` 命令建立新的存取金鑰。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的管理 IAM 使用者的存取金鑰。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListAccessKeys。

`list-account-aliases`

以下程式碼範例顯示如何使用 `list-account-aliases`。

AWS CLI

列出帳戶別名

下列 list-account-aliases 命令會列出目前帳戶的別名。

```
aws iam list-account-aliases
```

輸出：

```
{
    "AccountAliases": [
    "mycompany"
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的AWS 您的帳戶 ID 及其別名。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListAccountAliases。

list-attached-group-policies

以下程式碼範例顯示如何使用 list-attached-group-policies。

AWS CLI

若要列出連接至指定群組所有受管政策

此範例會傳回連接到Admins AWS 帳戶中名為 之 IAM 群組的受管政策的名稱和 ARNs。

```
aws iam list-attached-group-policies \
    --group-name Admins
```

輸出：

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        },
        {
            "PolicyName": "SecurityAudit",
            "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
        }
```

```
    ],
    "IsTruncated": false
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 IAM 中的政策和許可。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 ListAttachedGroupPolicies。

list-attached-role-policies

以下程式碼範例顯示如何使用 list-attached-role-policies。

AWS CLI

列出連接至指定角色的所有受管政策

此命令會傳回連接到SecurityAuditRole AWS 帳戶中名為 之 IAM 角色的受管政策的名稱和 ARNs。

```
aws iam list-attached-role-policies \
    --role-name SecurityAuditRole
```

輸出：

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "SecurityAudit",
            "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
        }
    ],
    "IsTruncated": false
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 IAM 中的政策和許可。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListAttachedRolePolicies。

list-attached-user-policies

以下程式碼範例顯示如何使用 list-attached-user-policies。

AWS CLI

若要列出連接至指定使用者的所有受管政策

此命令會傳回 AWS 帳戶中名為 之 IAM 使用者Bob之受管政策的名稱和 ARNs。

```
aws iam list-attached-user-policies \
    --user-name Bob
```

輸出：

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        },
        {
            "PolicyName": "SecurityAudit",
            "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
        }
    ],
    "IsTruncated": false
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 IAM 中的政策和許可。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 ListAttachedUserPolicies。

list-entities-for-policy

以下程式碼範例顯示如何使用 list-entities-for-policy。

AWS CLI

若要列出指定受管政策所連接的所有使用者、群組和角色

此範例會傳回已連接政策 arn:aws:iam::123456789012:policy/TestPolicy 的 IAM 群組、角色和使用者的清單。

```
aws iam list-entities-for-policy \
```

```
    --policy-arn arn:aws:iam::123456789012:policy/TestPolicy
```

輸出：

```
{
    "PolicyGroups": [
        {
            "GroupName": "Admins",
            "GroupId": "AGPACKCEVSQ6C2EXAMPLE"
        }
    ],
    "PolicyUsers": [
        {
            "UserName": "Alice",
            "UserId": "AIDACKCEVSQ6C2EXAMPLE"
        }
    ],
    "PolicyRoles": [
        {
            "RoleName": "DevRole",
            "RoleId": "AROADBQP57FF2AEXAMPLE"
        }
    ],
    "IsTruncated": false
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 IAM 中的政策和許可。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 ListEntitiesForPolicy。

list-group-policies

以下程式碼範例顯示如何使用 `list-group-policies`。

AWS CLI

若要列出連接至指定群組的所有內嵌政策

下列 `list-group-policies` 命令會列出連接至目前帳戶中名為 `Admins` 之 IAM 群組的內嵌政策名稱。

```
aws iam list-group-policies \
```

```
    --group-name Admins
```

輸出：

```
{
    "PolicyNames": [
        "AdminRoot",
        "ExamplePolicy"
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的管理 IAM 政策。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 ListGroupPolicies。

list-groups-for-user

以下程式碼範例顯示如何使用 list-groups-for-user。

AWS CLI

若要列出 IAM 使用者所屬的群組

下列 list-groups-for-user 命令顯示名為 Bob 之 IAM 使用者所屬的群組。

```
aws iam list-groups-for-user \
    --user-name Bob
```

輸出：

```
{
    "Groups": [
        {
            "Path": "/",
            "CreateDate": "2013-05-06T01:18:08Z",
            "GroupId": "AKIAIOSFODNN7EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/Admin",
            "GroupName": "Admin"
        },
        {
            "Path": "/",
            "CreateDate": "2013-05-06T01:37:28Z",
```

```
            "GroupId": "AKIAI44QH8DHBEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/s3-Users",
            "GroupName": "s3-Users"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的管理 IAM 使用者群組。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 ListGroupsForUser。

`list-groups`

以下程式碼範例顯示如何使用 `list-groups`。

AWS CLI

列出目前帳戶的 IAM 群組

下列 `list-groups` 命令會列出目前帳戶中的 IAM 群組。

```
aws iam list-groups
```

輸出：

```
{
    "Groups": [
        {
            "Path": "/",
            "CreateDate": "2013-06-04T20:27:27.972Z",
            "GroupId": "AIDACKCEVSQ6C2EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/Admins",
            "GroupName": "Admins"
        },
        {
            "Path": "/",
            "CreateDate": "2013-04-16T20:30:42Z",
            "GroupId": "AIDGPMS9RO4H3FEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/S3-Admins",
            "GroupName": "S3-Admins"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的管理 IAM 使用者群組。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListGroups。

list-instance-profile-tags

以下程式碼範例顯示如何使用 list-instance-profile-tags。

AWS CLI

列出連接至執行個體描述檔的標籤

下列list-instance-profile-tags命令會擷取與指定執行個體描述檔相關聯的標籤清單。

```
aws iam list-instance-profile-tags \
    --instance-profile-name deployment-role
```

輸出：

```
{
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListInstanceProfileTags。

list-instance-profiles-for-role

以下程式碼範例顯示如何使用 list-instance-profiles-for-role。

AWS CLI

若要列出 IAM 角色的執行個體設定檔

下列 `list-instance-profiles-for-role` 命令會列出與角色 `Test-Role` 關聯的執行個體設定檔。

```
aws iam list-instance-profiles-for-role \
    --role-name Test-Role
```

輸出：

```
{
    "InstanceProfiles": [
        {
            "InstanceProfileId": "AIDGPMS9RO4H3FEXAMPLE",
            "Roles": [
                {
                    "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
                    "RoleId": "AIDACKCEVSQ6C2EXAMPLE",
                    "CreateDate": "2013-06-07T20:42:15Z",
                    "RoleName": "Test-Role",
                    "Path": "/",
                    "Arn": "arn:aws:iam::123456789012:role/Test-Role"
                }
            ],
            "CreateDate": "2013-06-07T21:05:24Z",
            "InstanceProfileName": "ExampleInstanceProfile",
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:instance-profile/
ExampleInstanceProfile"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用執行個體設定檔。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 ListInstanceProfilesForRole。

`list-instance-profiles`

以下程式碼範例顯示如何使用 `list-instance-profiles`。

AWS CLI

若要列出帳戶的執行個體設定檔

下列 list-instance-profiles 命令列出與目前帳戶關聯的執行個體設定檔。

```
aws iam list-instance-profiles
```

輸出：

```
{
    "InstanceProfiles": [
        {
            "Path": "/",
            "InstanceProfileName": "example-dev-role",
            "InstanceProfileId": "AIPAIXEU4NUHUPEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:instance-profile/example-dev-role",
            "CreateDate": "2023-09-21T18:17:41+00:00",
            "Roles": [
                {
                    "Path": "/",
                    "RoleName": "example-dev-role",
                    "RoleId": "AROAJ52OTH4H7LEXAMPLE",
                    "Arn": "arn:aws:iam::123456789012:role/example-dev-role",
                    "CreateDate": "2023-09-21T18:17:40+00:00",
                    "AssumeRolePolicyDocument": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Principal": {
                                    "Service": "ec2.amazonaws.com"
                                },
                                "Action": "sts:AssumeRole"
                            }
                        ]
                    }
                }
            ]
        },
        {
            "Path": "/",
            "InstanceProfileName": "example-s3-role",
```

```
            "InstanceProfileId": "AIPAJVJVNRIQFREXAMPLE",
            "Arn": "arn:aws:iam::123456789012:instance-profile/example-s3-role",
            "CreateDate": "2023-09-21T18:18:50+00:00",
            "Roles": [
                {
                    "Path": "/",
                    "RoleName": "example-s3-role",
                    "RoleId": "AROAINUBC5O7XLEXAMPLE",
                    "Arn": "arn:aws:iam::123456789012:role/example-s3-role",
                    "CreateDate": "2023-09-21T18:18:49+00:00",
                    "AssumeRolePolicyDocument": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Principal": {
                                    "Service": "ec2.amazonaws.com"
                                },
                                "Action": "sts:AssumeRole"
                            }
                        ]
                    }
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用執行個體設定檔。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 ListInstanceProfiles。

`list-mfa-device-tags`

以下程式碼範例顯示如何使用 `list-mfa-device-tags`。

AWS CLI

列出連接至 MFA 裝置的標籤

下列`list-mfa-device-tags`命令會擷取與指定 MFA 裝置相關聯的標籤清單。

```
aws iam list-mfa-device-tags \
```

```
    --serial-number arn:aws:iam::123456789012:mfa/alice
```

輸出：

```
{
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMfaDeviceTags。

`list-mfa-devices`

以下程式碼範例顯示如何使用 `list-mfa-devices`。

AWS CLI

若要列出指定使用者的所有 MFA 裝置

此範例會傳回指派給 IAM 使用者 Bob 之 MFA 裝置的詳細資訊。

```
aws iam list-mfa-devices \
    --user-name Bob
```

輸出：

```
{
    "MFADevices": [
        {
            "UserName": "Bob",
            "SerialNumber": "arn:aws:iam::123456789012:mfa/Bob",
```

```
            "EnableDate": "2019-10-28T20:37:09+00:00"
        },
        {
            "UserName": "Bob",
            "SerialNumber": "GAKT12345678",
            "EnableDate": "2023-02-18T21:44:42+00:00"
        },
        {
            "UserName": "Bob",
            "SerialNumber": "arn:aws:iam::123456789012:u2f/user/Bob/
fidosecuritykey1-7XNL7NFNLZ123456789EXAMPLE",
            "EnableDate": "2023-09-19T02:25:35+00:00"
        },
        {
            "UserName": "Bob",
            "SerialNumber": "arn:aws:iam::123456789012:u2f/user/Bob/
fidosecuritykey2-VDRQTDBBN5123456789EXAMPLE",
            "EnableDate": "2023-09-19T01:49:18+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的在 AWS中使用多重要素驗證 (MFA)。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 ListMfaDevices。

`list-open-id-connect-provider-tags`

以下程式碼範例顯示如何使用 `list-open-id-connect-provider-tags`。

AWS CLI

列出連接至 OpenID Connect (OIDC) 相容身分提供者的標籤

下列`list-open-id-connect-provider-tags`命令會擷取與指定 OIDC 身分提供者相關聯的標籤清單。

```
aws iam list-open-id-connect-provider-tags \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com
```

輸出：

```
{
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListOpenIdConnectProviderTags。

`list-open-id-connect-providers`

以下程式碼範例顯示如何使用 `list-open-id-connect-providers`。

AWS CLI

列出 AWS 帳戶中 OpenID Connect 供應商的相關資訊

此範例會傳回目前 AWS 帳戶中定義之所有 OpenID Connect 供應商的 ARNS 清單。

```
aws iam list-open-id-connect-providers
```

輸出：

```
{
    "OpenIDConnectProviderList": [
        {
            "Arn": "arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Creating OpenID Connect (OIDC) identity providers。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 ListOpenIdConnectProviders。

`list-organizations-features`

以下程式碼範例顯示如何使用 `list-organizations-features`。

AWS CLI

列出為您的組織啟用的集中式根存取功能

下列`list-organizations-features`命令列出為您的組織啟用的集中式根存取功能。

```
aws iam list-organizations-features
```

輸出：

```
{
    "EnabledFeatures": [
        "RootCredentialsManagement",
        "RootSessions"
    ]
    "OrganizationId": "o-aa111bb222"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的集中管理成員帳戶的根存取權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListOrganizationsFeatures。

`list-policies-granting-service-access`

以下程式碼範例顯示如何使用 `list-policies-granting-service-access`。

AWS CLI

列出授予委託人存取指定服務的政策

下列`list-policies-granting-service-access`範例會擷取授予 IAM 使用者 AWS CodeCommit 服務`sofia`存取權的政策清單。

```
aws iam list-policies-granting-service-access \
    --arn arn:aws:iam::123456789012:user/sofia \
```

```
    --service-namespaces codecommit
```

輸出：

```
{
    "PoliciesGrantingServiceAccess": [
        {
            "ServiceNamespace": "codecommit",
            "Policies": [
                {
                    "PolicyName": "Grant-Sofia-Access-To-CodeCommit",
                    "PolicyType": "INLINE",
                    "EntityType": "USER",
                    "EntityName": "sofia"
                }
            ]
        }
    ],
    "IsTruncated": false
}
```

如需詳細資訊，請參閱《IAM 使用者指南》中的搭配使用 IAM 與 CodeCommit：Git 登入資料、SSH 金鑰和 AWS 存取金鑰。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPoliciesGrantingServiceAccess。

list-policies

以下程式碼範例顯示如何使用 `list-policies`。

AWS CLI

列出可供您 AWS 帳戶使用的受管政策

此範例會傳回目前 AWS 帳戶中可用的前兩個受管政策的集合。

```
aws iam list-policies \
    --max-items 3
```

輸出：

```
{
```

```
    "Policies": [
        {
            "PolicyName": "AWSCloudTrailAccessPolicy",
            "PolicyId": "ANPAXQE2B5PJ7YEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:policy/AWSCloudTrailAccessPolicy",
            "Path": "/",
            "DefaultVersionId": "v1",
            "AttachmentCount": 0,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2019-09-04T17:43:42+00:00",
            "UpdateDate": "2019-09-04T17:43:42+00:00"
        },
        {
            "PolicyName": "AdministratorAccess",
            "PolicyId": "ANPAIWMBCKSKIEE64ZLYK",
            "Arn": "arn:aws:iam::aws:policy/AdministratorAccess",
            "Path": "/",
            "DefaultVersionId": "v1",
            "AttachmentCount": 6,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2015-02-06T18:39:46+00:00",
            "UpdateDate": "2015-02-06T18:39:46+00:00"
        },
        {
            "PolicyName": "PowerUserAccess",
            "PolicyId": "ANPAJYRXTHIB4FOVS3ZXS",
            "Arn": "arn:aws:iam::aws:policy/PowerUserAccess",
            "Path": "/",
            "DefaultVersionId": "v5",
            "AttachmentCount": 1,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2015-02-06T18:39:47+00:00",
            "UpdateDate": "2023-07-06T22:04:00+00:00"
        }
    ],
    "NextToken": "EXAMPLErZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiA4fQ=="
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 IAM 中的政策和許可。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListPolicies。

list-policy-tags

以下程式碼範例顯示如何使用 list-policy-tags。

AWS CLI

列出連接至受管政策的標籤

下列list-policy-tags命令會擷取與指定受管政策相關聯的標籤清單。

```
aws iam list-policy-tags \
    --policy-arn arn:aws:iam::123456789012:policy/billing-access
```

輸出：

```
{
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPolicyTags。

list-policy-versions

以下程式碼範例顯示如何使用 list-policy-versions。

AWS CLI

若要列出指定受管政策的版本相關資訊

此範例會傳回 ARN 為 arn:aws:iam::123456789012:policy/MySamplePolicy 之政策的可用版本清單。

```
aws iam list-policy-versions \
    --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

輸出：

```
{
    "IsTruncated": false,
    "Versions": [
        {
        "VersionId": "v2",
        "IsDefaultVersion": true,
        "CreateDate": "2015-06-02T23:19:44Z"
        },
        {
        "VersionId": "v1",
        "IsDefaultVersion": false,
        "CreateDate": "2015-06-02T22:30:47Z"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 IAM 中的政策和許可。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 ListPolicyVersions。

list-role-policies

以下程式碼範例顯示如何使用 list-role-policies。

AWS CLI

列出連接至 IAM 角色的政策

下列 list-role-policies 命令會列出指定 IAM 角色的許可政策名稱。

```
aws iam list-role-policies \
    --role-name Test-Role
```

輸出：

```
{
```

```
    "PolicyNames": [
        "ExamplePolicy"
    ]
}
```

若要查看連接至角色的信任政策，請使用 `get-role` 命令。若要查看許可政策的詳細資訊，請使用 `get-role-policy` 命令。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的建立 IAM 角色。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListRolePolicies。

`list-role-tags`

以下程式碼範例顯示如何使用 `list-role-tags`。

AWS CLI

若要列出連接至角色的標籤

下列 `list-role-tags` 命令會擷取與指定角色關聯的標籤之清單。

```
aws iam list-role-tags \
    --role-name production-role
```

輸出：

```
{
    "Tags": [
        {
            "Key": "Department",
            "Value": "Accounting"
        },
        {
            "Key": "DeptID",
            "Value": "12345"
        }
    ],
    "IsTruncated": false
}
```

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 ListRoleTags。

list-roles

以下程式碼範例顯示如何使用 list-roles。

AWS CLI

列出目前帳戶的 IAM 角色

下列 list-roles 命令會列出目前帳戶的 IAM 角色。

```
aws iam list-roles
```

輸出：

```
{
    "Roles": [
        {
            "Path": "/",
            "RoleName": "ExampleRole",
            "RoleId": "AROAJ52OTH4H7LEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:role/ExampleRole",
            "CreateDate": "2017-09-12T19:23:36+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/example_path/",
            "RoleName": "ExampleRoleWithPath",
```

```
            "RoleId": "AROAI4QRP7UFT7EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:role/example_path/
ExampleRoleWithPath",
            "CreateDate": "2023-09-21T20:29:38+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的建立 IAM 角色。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListRoles。

list-saml-provider-tags

以下程式碼範例顯示如何使用 list-saml-provider-tags。

AWS CLI

列出連接到 SAML 供應商的標籤

下列list-saml-provider-tags命令會擷取與指定 SAML 供應商相關聯的標籤清單。

```
aws iam list-saml-provider-tags \
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS
```

輸出：

```
{
```

```
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSamlProviderTags。

`list-saml-providers`

以下程式碼範例顯示如何使用 `list-saml-providers`。

AWS CLI

列出 AWS 帳戶中的 SAML 供應商

此範例會擷取目前 AWS 帳戶中建立的 SAML 2.0 供應商清單。

```
aws iam list-saml-providers
```

輸出：

```
{
    "SAMLProviderList": [
        {
            "Arn": "arn:aws:iam::123456789012:saml-provider/SAML-ADFS",
            "ValidUntil": "2015-06-05T22:45:14Z",
            "CreateDate": "2015-06-05T22:45:14Z"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的建立 IAM SAML 身分提供者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListSAMLProviders。

list-server-certificate-tags

以下程式碼範例顯示如何使用 list-server-certificate-tags。

AWS CLI

列出連接到伺服器憑證的標籤

下列list-server-certificate-tags命令會擷取與指定伺服器憑證相關聯的標籤清單。

```
aws iam list-server-certificate-tags \
    --server-certificate-name ExampleCertificate
```

輸出：

```
{
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListServerCertificateTags。

list-server-certificates

以下程式碼範例顯示如何使用 list-server-certificates。

AWS CLI

列出您 AWS 帳戶中的伺服器憑證

下列list-server-certificates命令會列出您 AWS 帳戶中存放和可使用的所有伺服器憑證。

```
aws iam list-server-certificates
```

輸出：

```
{
    "ServerCertificateMetadataList": [
        {
            "Path": "/",
            "ServerCertificateName": "myUpdatedServerCertificate",
            "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:server-certificate/
myUpdatedServerCertificate",
            "UploadDate": "2019-04-22T21:13:44+00:00",
            "Expiration": "2019-10-15T22:23:16+00:00"
        },
        {
            "Path": "/cloudfront/",
            "ServerCertificateName": "MyTestCert",
            "ServerCertificateId": "ASCAEXAMPLE456EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:server-certificate/Org1/Org2/
MyTestCert",
            "UploadDate": "2015-04-21T18:14:16+00:00",
            "Expiration": "2018-01-14T17:52:36+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的在 IAM 中管理伺服器憑證。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListServerCertificates。

list-service-specific-credential

以下程式碼範例顯示如何使用 list-service-specific-credential。

AWS CLI

範例 1：列出使用者的服務特定登入資料

下列list-service-specific-credentials範例顯示指派給指定使用者的所有服務特定登入資料。密碼不包含在回應中。

```
aws iam list-service-specific-credentials \
    --user-name sofia
```

輸出：

```
{
    "ServiceSpecificCredential": {
        "CreateDate": "2019-04-18T20:45:36+00:00",
        "ServiceName": "codecommit.amazonaws.com",
        "ServiceUserName": "sofia-at-123456789012",
        "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
        "UserName": "sofia",
        "Status": "Active"
    }
}
```

範例 2：列出篩選至指定服務之使用者的服務特定登入資料

下列`list-service-specific-credentials`範例顯示指派給提出請求之使用者的服務特定登入資料。清單會經過篩選，只包含指定服務的登入資料。密碼不包含在回應中。

```
aws iam list-service-specific-credentials \
    --service-name codecommit.amazonaws.com
```

輸出：

```
{
    "ServiceSpecificCredential": {
        "CreateDate": "2019-04-18T20:45:36+00:00",
        "ServiceName": "codecommit.amazonaws.com",
        "ServiceUserName": "sofia-at-123456789012",
        "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
        "UserName": "sofia",
        "Status": "Active"
    }
}
```

如需詳細資訊，請參閱CodeCommit》中的為 HTTPS 連線至 CodeCommit 建立 Git AWS CodeCommit 登入資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListServiceSpecificCredential。

list-service-specific-credentials

以下程式碼範例顯示如何使用 list-service-specific-credentials。

AWS CLI

擷取登入資料清單

下列list-service-specific-credentials範例列出針對名為 之使用者對 AWS CodeCommit 儲存庫的 HTTPS 存取所產生的登入資料developer。

```
aws iam list-service-specific-credentials \
    --user-name developer \
    --service-name codecommit.amazonaws.com
```

輸出：

```
{
    "ServiceSpecificCredentials": [
        {
            "UserName": "developer",
            "Status": "Inactive",
            "ServiceUserName": "developer-at-123456789012",
            "CreateDate": "2019-10-01T04:31:41Z",
            "ServiceSpecificCredentialId": "ACCAQFODXMPL4YFHP7DZE",
            "ServiceName": "codecommit.amazonaws.com"
        },
        {
            "UserName": "developer",
            "Status": "Active",
            "ServiceUserName": "developer+1-at-123456789012",
            "CreateDate": "2019-10-01T04:31:45Z",
            "ServiceSpecificCredentialId": "ACCAQFOXMPL6VW57M7AJP",
            "ServiceName": "codecommit.amazonaws.com"
        }
    ]
}
```

如需詳細資訊，請參閱CodeCommit》中的為 HTTPS 連線至 CodeCommit 建立 Git AWS CodeCommit 登入資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListServiceSpecificCredentials。

list-signing-certificates

以下程式碼範例顯示如何使用 list-signing-certificates。

AWS CLI

若要列出 IAM 使用者的簽署憑證

下列 list-signing-certificates 命令列出名為 Bob 之 IAM 使用者的簽署憑證。

```
aws iam list-signing-certificates \
    --user-name Bob
```

輸出：

```
{
    "Certificates": [
        {
            "UserName": "Bob",
            "Status": "Inactive",
            "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-body>-----
END CERTIFICATE-----",
            "CertificateId": "TA7SMP42TDN5Z26OBPJE7EXAMPLE",
            "UploadDate": "2013-06-06T21:40:08Z"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的管理簽署憑證。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 ListSigningCertificates。

list-ssh-public-keys

以下程式碼範例顯示如何使用 list-ssh-public-keys。

AWS CLI

列出連接到 IAM 使用者的 SSH 公有金鑰

下列list-ssh-public-keys範例列出連接至 IAM 使用者 的 SSH 公有金鑰sofia。

```
aws iam list-ssh-public-keys \
```

```
    --user-name sofia
```

輸出：

```
{
    "SSHPublicKeys": [
        {
            "UserName": "sofia",
            "SSHPublicKeyId": "APKA1234567890EXAMPLE",
            "Status": "Inactive",
            "UploadDate": "2019-04-18T17:04:49+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用 SSH 金鑰和 SSH 搭配 CodeCommit

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSshPublicKeys。

list-user-policies

以下程式碼範例顯示如何使用 list-user-policies。

AWS CLI

列出 IAM 使用者的政策

下列 list-user-policies 命令會列出連接至名為 Bob 之 IAM 使用者的政策。

```
aws iam list-user-policies \
    --user-name Bob
```

輸出：

```
{
    "PolicyNames": [
        "ExamplePolicy",
        "TestPolicy"
    ]
}
```

如需詳細資訊，請參閱《IAM 使用者指南》中的在 AWS 帳戶中建立 AWS IAM 使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListUserPolicies。

list-user-tags

以下程式碼範例顯示如何使用 list-user-tags。

AWS CLI

若要列出連接至使用者的標籤

下列 list-user-tags 命令會擷取與指定 IAM 使用者關聯的標籤之清單。

```
aws iam list-user-tags \
    --user-name alice
```

輸出：

```
{
    "Tags": [
        {
            "Key": "Department",
            "Value": "Accounting"
        },
        {
            "Key": "DeptID",
            "Value": "12345"
        }
    ],
    "IsTruncated": false
}
```

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 ListUserTags。

list-users

以下程式碼範例顯示如何使用 list-users。

AWS CLI

列出 IAM 使用者

下列 list-users 命令會列出目前帳戶中的 IAM 使用者。

```
aws iam list-users
```

輸出：

```
{
    "Users": [
        {
            "UserName": "Adele",
            "Path": "/",
            "CreateDate": "2013-03-07T05:14:48Z",
            "UserId": "AKIAI44QH8DHBEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/Adele"
        },
        {
            "UserName": "Bob",
            "Path": "/",
            "CreateDate": "2012-09-21T23:03:13Z",
            "UserId": "AKIAIOSFODNN7EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/Bob"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的列出 IAM 使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListUsers。

list-virtual-mfa-devices

以下程式碼範例顯示如何使用 list-virtual-mfa-devices。

AWS CLI

若要列出虛擬 MFA 裝置

下列 list-virtual-mfa-devices 命令會列出已為目前帳戶設定的虛擬 MFA 裝置。

```
aws iam list-virtual-mfa-devices
```

輸出：

```
{
    "VirtualMFADevices": [
        {
            "SerialNumber": "arn:aws:iam::123456789012:mfa/ExampleMFADevice"
        },
        {
            "SerialNumber": "arn:aws:iam::123456789012:mfa/Fred"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的啟用虛擬多重要素驗證 (MFA) 裝置。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListVirtualMfaDevices。

put-group-policy

以下程式碼範例顯示如何使用 put-group-policy。

AWS CLI

將政策新增至群組

下列 put-group-policy 命令會將政策新增至名為 Admins 的 IAM 群組。

```
aws iam put-group-policy \
    --group-name Admins \
    --policy-document file://AdminPolicy.json \
    --policy-name AdminRoot
```

此命令不會產生輸出。

會在 AdminPolicy.json 檔案中將此政策定義為 JSON 文件。(檔案名稱和副檔名沒有意義。)

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的管理 IAM 政策。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutGroupPolicy。

put-role-permissions-boundary

以下程式碼範例顯示如何使用 put-role-permissions-boundary。

AWS CLI

範例 1：根據自訂政策將許可界限套用至 IAM 角色

下列 put-role-permissions-boundary 範例會將名為 intern-boundary 的自訂政策套用為指定 IAM 角色的許可界限。

```
aws iam put-role-permissions-boundary \
    --permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \
    --role-name lambda-application-role
```

此命令不會產生輸出。

範例 2：根據 AWS 受管政策將許可界限套用至 IAM 角色

下列put-role-permissions-boundary範例會將 AWS 受管PowerUserAccess政策套用為指定 IAM 角色的許可界限。

```
aws iam put-role-permissions-boundary \
    --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \
    --role-name x-account-admin
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的修改角色。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 PutRolePermissionsBoundary。

put-role-policy

以下程式碼範例顯示如何使用 put-role-policy。

AWS CLI

將許可政策連接至 IAM 角色

下列 put-role-policy 命令會將許可政策連接到名為 Test-Role 的角色。

```
aws iam put-role-policy \
    --role-name Test-Role \
```

```
    --policy-name ExamplePolicy \
    --policy-document file://AdminPolicy.json
```

此命令不會產生輸出。

會在 AdminPolicy.json 檔案中將此政策定義為 JSON 文件。(檔案名稱和副檔名沒有意義。)

若要將信任政策連接至角色，請使用 update-assume-role-policy 命令。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的修改角色。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutRolePolicy。

put-user-permissions-boundary

以下程式碼範例顯示如何使用 put-user-permissions-boundary。

AWS CLI

範例 1：根據自訂政策將許可界限套用至 IAM 使用者

下列 put-user-permissions-boundary 範例會將名為 intern-boundary 的自訂政策套用為指定 IAM 使用者的許可界限。

```
aws iam put-user-permissions-boundary \
    --permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \
    --user-name intern
```

此命令不會產生輸出。

範例 2：根據 AWS 受管政策將許可界限套用至 IAM 使用者

下列put-user-permissions-boundary範例會套用名為 的 AWS 受管污染，PowerUserAccess做為指定 IAM 使用者的許可界限。

```
aws iam put-user-permissions-boundary \
    --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \
    --user-name developer
```

此命令不會產生輸出。

如需詳細資訊，請參閱《IAM 使用者指南》AWS 中的新增和移除 IAM 身分許可。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 PutUserPermissionsBoundary。

put-user-policy

以下程式碼範例顯示如何使用 put-user-policy。

AWS CLI

將政策連接至 IAM 使用者

下列 put-user-policy 命令會將政策連接至名為 Bob 的 IAM 使用者。

```
aws iam put-user-policy \
    --user-name Bob \
    --policy-name ExamplePolicy \
    --policy-document file://AdminPolicy.json
```

此命令不會產生輸出。

會在 AdminPolicy.json 檔案中將此政策定義為 JSON 文件。(檔案名稱和副檔名沒有意義。)

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的新增和移除 IAM 身分許可。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutUserPolicy。

remove-client-id-from-open-id-connect-provider

以下程式碼範例顯示如何使用 remove-client-id-from-open-id-connect-provider。

AWS CLI

若要從已向指定 IAM OpenID Connect 提供者註冊的用戶端 ID 清單中移除指定的用戶端 ID

此範例會從與 ARN 為 arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com 的 IAM OIDC 提供者關聯的用戶端 ID 清單中移除用戶端 ID My-TestApp-3。

```
aws iam remove-client-id-from-open-id-connect-provider
    --client-id My-TestApp-3 \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Creating OpenID Connect (OIDC) identity providers。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 RemoveClientIdFromOpenIdConnectProvider。

remove-role-from-instance-profile

以下程式碼範例顯示如何使用 remove-role-from-instance-profile。

AWS CLI

若要從執行個體設定檔中移除角色

下列 remove-role-from-instance-profile 命令會將名為 Test-Role 的角色從名為 ExampleInstanceProfile 的執行個體設定檔中移除。

```
aws iam remove-role-from-instance-profile \
    --instance-profile-name ExampleInstanceProfile \
    --role-name Test-Role
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用執行個體設定檔。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 RemoveRoleFromInstanceProfile。

remove-user-from-group

以下程式碼範例顯示如何使用 remove-user-from-group。

AWS CLI

從 IAM 群組移除使用者

下列 remove-user-from-group 命令會將名為 Bob 的使用者從名為 Admins 的 IAM 群組中移除。

```
aws iam remove-user-from-group \
    --user-name Bob \
```

```
    --group-name Admins
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的在 IAM 使用者群組中新增和移除使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 RemoveUserFromGroup。

reset-service-specific-credential

以下程式碼範例顯示如何使用 reset-service-specific-credential。

AWS CLI

範例 1：重設連接至提出請求之使用者的服務特定登入資料的密碼

下列reset-service-specific-credential範例會為連接至發出請求的使用者的指定服務特定登入資料產生新的密碼編譯強式密碼。

```
aws iam reset-service-specific-credential \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

輸出：

```
{
    "ServiceSpecificCredential": {
        "CreateDate": "2019-04-18T20:45:36+00:00",
        "ServiceName": "codecommit.amazonaws.com",
        "ServiceUserName": "sofia-at-123456789012",
        "ServicePassword": "+oaFsNk7tLco+C/obP9GhhcOzGcKOayTmE3LnAmAmH4=",
        "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
        "UserName": "sofia",
        "Status": "Active"
    }
}
```

範例 2：為連接至指定使用者的服務特定登入資料重設密碼

下列reset-service-specific-credential範例會為連接至指定使用者的服務特定登入資料產生新的密碼編譯強式密碼。

```
aws iam reset-service-specific-credential \
```

```
    --user-name sofia \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

輸出：

```
{
    "ServiceSpecificCredential": {
        "CreateDate": "2019-04-18T20:45:36+00:00",
        "ServiceName": "codecommit.amazonaws.com",
        "ServiceUserName": "sofia-at-123456789012",
        "ServicePassword": "+oaFsNk7tLco+C/obP9GhhcOzGcKOayTmE3LnAmAmH4=",
        "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
        "UserName": "sofia",
        "Status": "Active"
    }
}
```

如需詳細資訊，請參閱《CodeCommit 使用者指南》中的為與 CodeCommit 的 HTTPS 連線建立 Git AWS CodeCommit 登入資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetServiceSpecificCredential。

resync-mfa-device

以下程式碼範例顯示如何使用 resync-mfa-device。

AWS CLI

若要同步 MFA 裝置

下列 resync-mfa-device 範例會將與 IAM 使用者 Bob 關聯且 ARN 為 arn:aws:iam::123456789012:mfa/BobsMFADevice 的 MFA 裝置，與提供兩個驗證碼的身分驗證器程式同步。

```
aws iam resync-mfa-device \
    --user-name Bob \
    --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \
    --authentication-code1 123456 \
    --authentication-code2 987654
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的在 AWS中使用多重要素驗證 (MFA)。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 ResyncMfaDevice。

set-default-policy-version

以下程式碼範例顯示如何使用 set-default-policy-version。

AWS CLI

若要將指定政策的指定版本設定為政策的預設版本。

此範例會將 ARN 為 arn:aws:iam::123456789012:policy/MyPolicy 的政策的 v2 版本設定為預設的作用中版本。

```
aws iam set-default-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --version-id v2
```

如需詳細資訊，請參閱「 IAM 使用者指南」AWS 中的 IAM 中的政策和許可。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 SetDefaultPolicyVersion。

set-security-token-service-preferences

以下程式碼範例顯示如何使用 set-security-token-service-preferences。

AWS CLI

設定全域端點字符版本

下列set-security-token-service-preferences範例會將 Amazon STS 設定為在您向全域端點進行身分驗證時使用第 2 版字符。

```
aws iam set-security-token-service-preferences \
    --global-endpoint-token-version v2Token
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南AWS 》中的在 區域中管理 AWS STS。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetSecurityTokenServicePreferences。

simulate-custom-policy

以下程式碼範例顯示如何使用 simulate-custom-policy。

AWS CLI

範例 1：模擬與 IAM 使用者或角色相關聯的所有 IAM 政策的效果

以下simulate-custom-policy說明如何提供政策並定義變數值，以及模擬 API 呼叫，以查看是否允許或拒絕。下列範例顯示僅在指定日期和時間之後才啟用資料庫存取的政策。模擬成功，因為模擬的動作和指定的aws:CurrentTime變數都符合政策的要求。

```
aws iam simulate-custom-policy \
    --policy-input-list '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"*","Condition":
{"DateGreaterThan":{"aws:CurrentTime":"2018-08-16T12:00:00Z"}}}}' \
    --action-names dynamodb:CreateBackup \
    --context-
entries "ContextKeyName='aws:CurrentTime',ContextKeyValues='2019-04-25T11:00:00Z',ContextKey
```

輸出：

```
{
    "EvaluationResults": [
        {
            "EvalActionName": "dynamodb:CreateBackup",
            "EvalResourceName": "*",
            "EvalDecision": "allowed",
            "MatchedStatements": [
                {
                    "SourcePolicyId": "PolicyInputList.1",
                    "StartPosition": {
                        "Line": 1,
                        "Column": 38
                    },
                    "EndPosition": {
                        "Line": 1,
                        "Column": 167
                    }
                }
            ],
            "MissingContextValues": []
        }
```

```
    ]
}
```

範例 2：模擬政策禁止的命令

下列simulate-custom-policy範例顯示模擬政策所禁止之命令的結果。在此範例中，提供的日期早於政策條件所需的日期。

```
aws iam simulate-custom-policy \
    --policy-input-list '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"*","Condition":
{"DateGreaterThan":{"aws:CurrentTime":"2018-08-16T12:00:00Z"}}}}' \
    --action-names dynamodb:CreateBackup \
    --context-
entries "ContextKeyName='aws:CurrentTime',ContextKeyValues='2014-04-25T11:00:00Z',ContextKey
```

輸出：

```
{
    "EvaluationResults": [
        {
            "EvalActionName": "dynamodb:CreateBackup",
            "EvalResourceName": "*",
            "EvalDecision": "implicitDeny",
            "MatchedStatements": [],
            "MissingContextValues": []
        }
    ]
}
```

如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM 政策模擬器測試 IAM 政策。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SimulateCustomPolicy。

simulate-principal-policy

以下程式碼範例顯示如何使用 simulate-principal-policy。

AWS CLI

範例 1：模擬任意 IAM 政策的效果

以下simulate-principal-policy說明如何模擬呼叫 API 動作的使用者，並判斷與該使用者相關聯的政策是否允許或拒絕該動作。在下列範例中，使用者具有僅允許codecommit:ListRepositories動作的政策。

```
aws iam simulate-principal-policy \
    --policy-source-arn arn:aws:iam::123456789012:user/alejandro \
    --action-names codecommit:ListRepositories
```

輸出：

```
{
    "EvaluationResults": [
        {
            "EvalActionName": "codecommit:ListRepositories",
            "EvalResourceName": "*",
            "EvalDecision": "allowed",
            "MatchedStatements": [
                {
                    "SourcePolicyId": "Grant-Access-To-CodeCommit-ListRepo",
                    "StartPosition": {
                        "Line": 3,
                        "Column": 19
                    },
                    "EndPosition": {
                        "Line": 9,
                        "Column": 10
                    }
                }
            ],
            "MissingContextValues": []
        }
    ]
}
```

範例 2：模擬禁止命令的效果

下列simulate-custom-policy範例顯示模擬其中一個使用者政策所禁止的命令的結果。在下列範例中，使用者有一個政策，僅允許在特定日期和時間之後存取 DynamoDB 資料庫。模擬會讓使用者嘗試使用早於政策條件允許aws:CurrentTime的值來存取資料庫。

```
aws iam simulate-principal-policy \
```

```
    --policy-source-arn arn:aws:iam::123456789012:user/alejandro \
    --action-names dynamodb:CreateBackup \
    --context-
entries "ContextKeyName='aws:CurrentTime',ContextKeyValues='2018-04-25T11:00:00Z',ContextKey
```

輸出：

```
{
    "EvaluationResults": [
        {
            "EvalActionName": "dynamodb:CreateBackup",
            "EvalResourceName": "*",
            "EvalDecision": "implicitDeny",
            "MatchedStatements": [],
            "MissingContextValues": []
        }
    ]
}
```

如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM 政策模擬器測試 IAM 政策。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SimulatePrincipalPolicy。

tag-instance-profile

以下程式碼範例顯示如何使用 tag-instance-profile。

AWS CLI

將標籤新增至執行個體描述檔

下列tag-instance-profile命令會將具有部門名稱的標籤新增至指定的執行個體描述檔。

```
aws iam tag-instance-profile \
    --instance-profile-name deployment-role \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TagInstanceProfile。

tag-mfa-device

以下程式碼範例顯示如何使用 tag-mfa-device。

AWS CLI

將標籤新增至 MFA 裝置

下列tag-mfa-device命令會將具有部門名稱的標籤新增至指定的 MFA 裝置。

```
aws iam tag-mfa-device \
    --serial-number arn:aws:iam::123456789012:mfa/alice \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TagMfaDevice。

tag-open-id-connect-provider

以下程式碼範例顯示如何使用 tag-open-id-connect-provider。

AWS CLI

將標籤新增至與 OpenID Connect (OIDC) 相容的身分提供者

下列tag-open-id-connect-provider命令會將具有部門名稱的標籤新增至指定的 OIDC 身分提供者。

```
aws iam tag-open-id-connect-provider \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TagOpenIdConnectProvider。

tag-policy

以下程式碼範例顯示如何使用 tag-policy。

AWS CLI

將標籤新增至客戶受管政策

下列tag-policy命令會將具有部門名稱的標籤新增至指定的客戶受管政策。

```
aws iam tag-policy \
    --policy-arn arn:aws:iam::123456789012:policy/billing-access \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TagPolicy。

tag-role

以下程式碼範例顯示如何使用 tag-role。

AWS CLI

若要將標籤新增至角色

下列 tag-role 命令會將含部門名稱的標籤新增至指定角色。

```
aws iam tag-role --role-name my-role \
    --tags '{"Key": "Department", "Value": "Accounting"}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 TagRole。

tag-saml-provider

以下程式碼範例顯示如何使用 tag-saml-provider。

AWS CLI

將標籤新增至 SAML 供應商

下列tag-saml-provider命令會將具有部門名稱的標籤新增至指定的 SAML 供應商。

```
aws iam tag-saml-provider \
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TagSamlProvider。

tag-server-certificate

以下程式碼範例顯示如何使用 tag-server-certificate。

AWS CLI

將標籤新增至伺服器憑證

下列tag-saml-provider命令會將具有部門名稱的標籤新增至指定的伺服器憑證。

```
aws iam tag-server-certificate \
    --server-certificate-name ExampleCertificate \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TagServerCertificate。

tag-user

以下程式碼範例顯示如何使用 tag-user。

AWS CLI

若要將標籤新增至使用者

下列 tag-user 命令將含關聯部門的標籤新增至指定使用者。

```
aws iam tag-user \
    --user-name alice \
    --tags '{"Key": "Department", "Value": "Accounting"}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 TagUser。

untag-instance-profile

以下程式碼範例顯示如何使用 untag-instance-profile。

AWS CLI

從執行個體描述檔移除標籤

下列untag-instance-profile命令會從指定的執行個體描述檔中移除金鑰名稱為 'Department' 的任何標籤。

```
aws iam untag-instance-profile \
    --instance-profile-name deployment-role \
    --tag-keys Department
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UntagInstanceProfile。

untag-mfa-device

以下程式碼範例顯示如何使用 untag-mfa-device。

AWS CLI

從 MFA 裝置移除標籤

下列untag-mfa-device命令會從指定的 MFA 裝置移除金鑰名稱為 'Department' 的任何標籤。

```
aws iam untag-mfa-device \
    --serial-number arn:aws:iam::123456789012:mfa/alice \
    --tag-keys Department
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UntagMfaDevice。

untag-open-id-connect-provider

以下程式碼範例顯示如何使用 untag-open-id-connect-provider。

AWS CLI

從 OIDC 身分提供者移除標籤

下列untag-open-id-connect-provider命令會從指定的 OIDC 身分提供者移除金鑰名稱為 'Department' 的任何標籤。

```
aws iam untag-open-id-connect-provider \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com \
    --tag-keys Department
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UntagOpenIdConnectProvider。

untag-policy

以下程式碼範例顯示如何使用 untag-policy。

AWS CLI

從客戶受管政策移除標籤

下列untag-policy命令會從指定的客戶受管政策中移除金鑰名稱為 'Department' 的任何標籤。

```
aws iam untag-policy \
    --policy-arn arn:aws:iam::452925170507:policy/billing-access \
    --tag-keys Department
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UntagPolicy。

untag-role

以下程式碼範例顯示如何使用 untag-role。

AWS CLI

若要從角色中移除標籤

下列 untag-role 命令會從指定的角色移除任何金鑰名稱為「Department」的標籤。

```
aws iam untag-role \
    --role-name my-role \
    --tag-keys Department
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UntagRole。

untag-saml-provider

以下程式碼範例顯示如何使用 untag-saml-provider。

AWS CLI

從 SAML 供應商移除標籤

下列untag-saml-provider命令會從指定的執行個體描述檔中移除金鑰名稱為 'Department' 的任何標籤。

```
aws iam untag-saml-provider \
```

```
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS \
    --tag-keys Department
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UntagSamlProvider。

untag-server-certificate

以下程式碼範例顯示如何使用 untag-server-certificate。

AWS CLI

從伺服器憑證移除標籤

下列untag-server-certificate命令會從指定的伺服器憑證中移除金鑰名稱為 'Department' 的任何標籤。

```
aws iam untag-server-certificate \
    --server-certificate-name ExampleCertificate \
    --tag-keys Department
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UntagServerCertificate。

untag-user

以下程式碼範例顯示如何使用 untag-user。

AWS CLI

若要從使用者中移除標籤

下列 untag-user 命令會從指定的使用者移除任何金鑰名稱為「Department」的標籤。

```
aws iam untag-user \
    --user-name alice \
```

```
    --tag-keys Department
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Tagging IAM resources。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UntagUser。

update-access-key

以下程式碼範例顯示如何使用 update-access-key。

AWS CLI

啟用或停用 IAM 使用者的存取金鑰

下列 update-access-key 命令會為名為 Bob 的 IAM 使用者停用指定的存取金鑰 (存取金鑰 ID 與私密存取金鑰)。

```
aws iam update-access-key \
    --access-key-id AKIAIOSFODNN7EXAMPLE \
    --status Inactive \
    --user-name Bob
```

此命令不會產生輸出。

停用金鑰表示它無法用於程式設計存取 AWS。但是，金鑰仍然可用，並且可以重新啟用。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的管理 IAM 使用者的存取金鑰。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateAccessKey。

update-account-password-policy

以下程式碼範例顯示如何使用 update-account-password-policy。

AWS CLI

若要設定或變更目前的帳戶密碼政策

下列 update-account-password-policy 命令會設定密碼政策，要求密碼最小長度為 8 個字元，且必須包含一個或多個數字。

```
aws iam update-account-password-policy \
    --minimum-password-length 8 \
    --require-numbers
```

此命令不會產生輸出。

變更帳戶的密碼政策會影響為帳戶中的 IAM 使用者建立的任何新密碼。密碼政策變更不影響現有密碼。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的設定 IAM 使用者的帳戶密碼政策。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UpdateAccountPasswordPolicy。

update-assume-role-policy

以下程式碼範例顯示如何使用 update-assume-role-policy。

AWS CLI

若要更新 IAM 角色的信任政策

下列 update-assume-role-policy 命令會更新名為 Test-Role 之角色的信任政策。

```
aws iam update-assume-role-policy \
    --role-name Test-Role \
    --policy-document file://Test-Role-Trust-Policy.json
```

此命令不會產生輸出。

在 Test-Role-Trust-Policy.json 檔案中，將信任政策定義為 JSON 文件。(檔案名稱和副檔名沒有意義。) 信任政策必須指定主體。

若要更新角色的許可政策，請使用 put-role-policy 命令。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的建立 IAM 角色。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UpdateAssumeRolePolicy。

update-group

以下程式碼範例顯示如何使用 update-group。

AWS CLI

若要重新命名 IAM 群組

下列 update-group 命令會將 IAM 群組 Test 的名稱變更為 Test-1。

```
aws iam update-group \
    --group-name Test \
    --new-group-name Test-1
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的重新命名 IAM 使用者群組。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UpdateGroup。

update-login-profile

以下程式碼範例顯示如何使用 update-login-profile。

AWS CLI

若要更新 IAM 使用者的密碼

下列 update-login-profile 命令會為名為 Bob 的 IAM 使用者建立新的密碼。

```
aws iam update-login-profile \
    --user-name Bob \
    --password <password>
```

此命令不會產生輸出。

若要設定帳戶的密碼政策，請使用 update-account-password-policy 命令。如果新密碼違反帳戶密碼政策，則命令會傳回 PasswordPolicyViolation 錯誤。

如果帳戶密碼政策允許，IAM 使用者可以使用 change-password 命令變更自己的密碼。

將密碼存放於安全處。密碼一旦遺失，便無法復原，您必須使用 create-login-profile 命令建立新密碼。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的管理 IAM 使用者的密碼。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UpdateLoginProfile。

update-open-id-connect-provider-thumbprint

以下程式碼範例顯示如何使用 update-open-id-connect-provider-thumbprint。

AWS CLI

若要以新清單取代現有的伺服器憑證指紋清單

此範例會更新 ARN 為 arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com 之 OIDC 提供者的憑證指紋清單，以使用新指紋。

```
aws iam update-open-id-connect-provider-thumbprint \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com \
    --thumbprint-list 7359755EXAMPLEabc3060bce3EXAMPLEec4542a3
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Creating OpenID Connect (OIDC) identity providers。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UpdateOpenIdConnectProviderThumbprint。

update-role-description

以下程式碼範例顯示如何使用 update-role-description。

AWS CLI

變更 IAM 角色的描述

下列 update-role 命令會將 IAM 角色的描述從 production-role 變更為 Main production role。

```
aws iam update-role-description \
    --role-name production-role \
    --description 'Main production role'
```

輸出：

```
{
    "Role": {
        "Path": "/",
        "RoleName": "production-role",
        "RoleId": "AROA1234567890EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:role/production-role",
        "CreateDate": "2017-12-06T17:16:37+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "AWS": "arn:aws:iam::123456789012:root"
                    },
                    "Action": "sts:AssumeRole",
                    "Condition": {}
                }
            ]
        },
        "Description": "Main production role"
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的修改角色。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UpdateRoleDescription。

update-role

以下程式碼範例顯示如何使用 update-role。

AWS CLI

若要變更 IAM 角色的描述或工作階段持續時間

下列 update-role 命令會將 IAM 角色 production-role 的描述變更為 Main production role，並將工作階段持續時間上限設定為 12 小時。

```
aws iam update-role \
    --role-name production-role \
```

```
    --description 'Main production role' \
    --max-session-duration 43200
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的修改角色。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UpdateRole。

update-saml-provider

以下程式碼範例顯示如何使用 update-saml-provider。

AWS CLI

若要更新現有 SAML 提供者的中繼資料文件

此範例會更新 IAM 中 ARN 為 arn:aws:iam::123456789012:saml-provider/SAMLADFS、使用檔案 SAMLMetaData.xml 中的新 SAML 中繼資料文件的 SAML 提供者。

```
aws iam update-saml-provider \
    --saml-metadata-document file://SAMLMetaData.xml \
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

輸出：

```
{
    "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/SAMLADFS"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的建立 IAM SAML 身分提供者。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UpdateSamlProvider。

update-server-certificate

以下程式碼範例顯示如何使用 update-server-certificate。

AWS CLI

變更 AWS 帳戶中伺服器憑證的路徑或名稱

下列 update-server-certificate 命令會將憑證名稱從 myServerCertificate 變更為 myUpdatedServerCertificate。該命令也會將路徑變更為 /cloudfront/，這樣 Amazon CloudFront 服務就可以存取該路徑。此命令不會產生輸出。您可以透過執行 list-server-certificates 命令來查看更新的結果。

```
aws-iam update-server-certificate \
    --server-certificate-name myServerCertificate \
    --new-server-certificate-name myUpdatedServerCertificate \
    --new-path /cloudfront/
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的在 IAM 中管理伺服器憑證。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateServerCertificate。

update-service-specific-credential

以下程式碼範例顯示如何使用 update-service-specific-credential。

AWS CLI

範例 1：更新請求使用者服務特定登入資料的狀態

下列update-service-specific-credential範例會變更向 提出請求之使用者所指定登入資料的狀態Inactive。

```
aws iam update-service-specific-credential \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE \
    --status Inactive
```

此命令不會產生輸出。

範例 2：更新指定使用者服務特定登入資料的狀態

下列update-service-specific-credential範例會將指定使用者的登入資料狀態變更為非作用中。

```
aws iam update-service-specific-credential \
    --user-name sofia \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE \
```

```
    --status Inactive
```

此命令不會產生輸出。

如需詳細資訊，請參閱 CodeCommit 使用者指南中的為 HTTPS 連線至 CodeCommit 建立 Git 登入資料 AWS CodeCommit

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateServiceSpecificCredential。

update-signing-certificate

以下程式碼範例顯示如何使用 update-signing-certificate。

AWS CLI

若要啟用或停用 IAM 使用者的簽署憑證

下列 update-signing-certificate 命令將停用名為 Bob 之 IAM 使用者的指定簽署憑證。

```
aws iam update-signing-certificate \
    --certificate-id TA7SMP42TDN5Z26OBPJE7EXAMPLE \
    --status Inactive \
    --user-name Bob
```

若要取得簽署憑證的 ID，請使用 list-signing-certificates 命令。

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的管理簽署憑證。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UpdateSigningCertificate。

update-ssh-public-key

以下程式碼範例顯示如何使用 update-ssh-public-key。

AWS CLI

變更 SSH 公有金鑰的狀態

下列update-ssh-public-key命令會將指定公有金鑰的狀態變更為 Inactive。

```
aws iam update-ssh-public-key \
    --user-name sofia \
```

```
    --ssh-public-key-id APKA1234567890EXAMPLE \
    --status Inactive
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用 SSH 金鑰和 SSH 搭配 CodeCommit。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSshPublicKey。

update-user

以下程式碼範例顯示如何使用 update-user。

AWS CLI

變更 IAM 使用者的名稱

下列 update-user 命令會將 IAM 使用者名稱從 Bob 變更為 Robert。

```
aws iam update-user \
    --user-name Bob \
    --new-user-name Robert
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的重新命名 IAM 使用者群組。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateUser。

upload-server-certificate

以下程式碼範例顯示如何使用 upload-server-certificate。

AWS CLI

將伺服器憑證上傳至 AWS 您的帳戶

下列 upload-server-certificate 命令會將伺服器憑證上傳至 AWS 您的帳戶。在此範例中，憑證位於檔案 public_key_cert_file.pem 中，相關聯的私密金鑰位於檔案 my_private_key.pem 中，而憑證授權機構 (CA) 提供的憑證鏈結位於 my_certificate_chain_file.pem 檔案中。檔案上傳完成後，其就會以 myServerCertificate 名稱提供使用。以 file:// 開頭的參數會告訴命令讀取檔案的內容，並將其用作參數值 (而不是檔案名稱本身)。

```
aws iam upload-server-certificate \
    --server-certificate-name myServerCertificate \
    --certificate-body file://public_key_cert_file.pem \
    --private-key file://my_private_key.pem \
    --certificate-chain file://my_certificate_chain_file.pem
```

輸出：

```
{
    "ServerCertificateMetadata": {
        "Path": "/",
        "ServerCertificateName": "myServerCertificate",
        "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",
        "Arn": "arn:aws:iam::1234567989012:server-certificate/myServerCertificate",
        "UploadDate": "2019-04-22T21:13:44+00:00",
        "Expiration": "2019-10-15T22:23:16+00:00"
    }
}
```

如需詳細資訊，請參閱《使用 IAM 指南》中的「建立、上傳和刪除伺服器憑證」。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UploadServrCertificate。

upload-signing-certificate

以下程式碼範例顯示如何使用 upload-signing-certificate。

AWS CLI

若要上傳 IAM 使用者的簽署憑證

下列 upload-signing-certificate 命令會上傳名為 Bob 之 IAM 使用者的簽署憑證。

```
aws iam upload-signing-certificate \
    --user-name Bob \
    --certificate-body file://certificate.pem
```

輸出：

```
{
    "Certificate": {
        "UserName": "Bob",
```

```
        "Status": "Active",
        "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-body>-----END
 CERTIFICATE-----",
        "CertificateId": "TA7SMP42TDN5Z26OBPJE7EXAMPLE",
        "UploadDate": "2013-06-06T21:40:08.121Z"
    }
}
```

憑證位於名為 certificate.pem 的 PEM 格式的檔案中。

如需詳細資訊，請參閱《IAM 使用指南》中的「建立和上傳使用者簽署憑證」。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UploadSigningCertificate。

upload-ssh-public-key

以下程式碼範例顯示如何使用 upload-ssh-public-key。

AWS CLI

上傳 SSH 公有金鑰並將其與使用者建立關聯

下列upload-ssh-public-key命令會上傳 檔案中找到的公有金鑰，sshkey.pub並將其連接到使用者 sofia。

```
aws iam upload-ssh-public-key \
    --user-name sofia \
    --ssh-public-key-body file://sshkey.pub
```

輸出：

```
{
    "SSHPublicKey": {
        "UserName": "sofia",
        "SSHPublicKeyId": "APKA1234567890EXAMPLE",
        "Fingerprint": "12:34:56:78:90:ab:cd:ef:12:34:56:78:90:ab:cd:ef",
        "SSHPublicKeyBody": "ssh-rsa <<long string generated by ssh-keygen
 command>>",
        "Status": "Active",
        "UploadDate": "2019-04-18T17:04:49+00:00"
    }
}
```

如需如何以適合此命令的格式產生金鑰的詳細資訊，請參閱《CodeCommit 使用者指南》中的 SSH 和 Linux、macOS 或 Unix：為 Git 和 CodeCommit 或 SSH 和 Windows 設定公有和私有金鑰：為 Git 和 CodeCommit 設定公有和私有金鑰。 AWS CodeCommit

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UploadSshPublicKey。

使用 的 IAM Access Analyzer 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 IAM Access Analyzer 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`apply-archive-rule`

以下程式碼範例顯示如何使用 `apply-archive-rule`。

AWS CLI

將封存規則套用至符合封存規則條件的現有問題清單

下列`apply-archive-rule`範例會將封存規則套用至符合封存規則條件的現有問題清單。

```
aws accessanalyzer apply-archive-rule \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
UnusedAccess-ConsoleAnalyzer-organization \
    --rule-name MyArchiveRule
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的封存規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ApplyArchiveRule。

cancel-policy-generation

以下程式碼範例顯示如何使用 cancel-policy-generation。

AWS CLI

取消請求的政策產生

下列cancel-policy-generation範例會取消請求的政策產生任務 ID。

```
aws accessanalyzer cancel-policy-generation \
    --job-id 923a56b0-ebb8-4e80-8a3c-a11ccfbcd6f2
```

此命令不會產生輸出。

如需更多資訊，請參閱「IAM 使用者指南」AWS 中的 IAM Access Analyzer 政策驗證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelPolicyGeneration。

check-access-not-granted

以下程式碼範例顯示如何使用 check-access-not-granted。

AWS CLI

檢查政策是否不允許指定的存取

下列check-access-not-granted範例會檢查政策是否不允許指定的存取。

```
aws accessanalyzer check-access-not-granted \
    --policy-document file://myfile.json \
    --access actions="s3:DeleteBucket","s3:GetBucketLocation" \
    --policy-type IDENTITY_POLICY
```

myfile.json 的內容：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

輸出：

```
{
    "result": "PASS",
    "message": "The policy document does not grant access to perform one or more of
 the listed actions."
}
```

如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer APIs 預覽存取權。AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CheckAccessNotGranted。

check-no-new-access

以下程式碼範例顯示如何使用 check-no-new-access。

AWS CLI

檢查與現有政策相比，更新的政策是否允許新存取

下列check-no-new-access範例會檢查與現有政策相比，更新的政策是否允許新存取。

```
aws accessanalyzer check-no-new-access \
    --existing-policy-document file://existing-policy.json \
    --new-policy-document file://new-policy.json \
    --policy-type IDENTITY_POLICY
```

existing-policy.json 的內容：

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

new-policy.json 的內容：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

輸出：

```
{
    "result": "FAIL",
    "message": "The modified permissions grant new access compared to your existing
 policy.",
    "reasons": [
```

```
        {
            "description": "New access in the statement with index: 0.",
            "statementIndex": 0
        }
    ]
}
```

如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer APIs 預覽存取權。AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CheckNoNewAccess。

check-no-public-access

以下程式碼範例顯示如何使用 check-no-public-access。

AWS CLI

檢查資源政策是否可以授予對指定資源類型的公開存取權

下列check-no-public-access範例會檢查資源政策是否可以授予對指定資源類型的公開存取權。

```
aws accessanalyzer check-no-public-access \
    --policy-document file://check-no-public-access-myfile.json \
    --resource-type AWS::S3::Bucket
```

myfile.json 的內容：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CheckNoPublicAccess",
            "Effect": "Allow",
            "Principal": { "AWS": "arn:aws:iam::111122223333:user/JohnDoe" },
            "Action": [
                "s3:GetObject"
            ]
        }
    ]
}
```

輸出：

```
{
    "result": "PASS",
    "message": "The resource policy does not grant public access for the given
 resource type."
}
```

如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer APIs 預覽存取權。AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CheckNoPublicAccess。

create-access-preview

以下程式碼範例顯示如何使用 create-access-preview。

AWS CLI

若要建立存取預覽，可讓您在部署資源許可之前預覽資源的 IAM Access Analyzer 問題清單

下列create-access-preview範例會建立存取預覽，可讓您在 AWS 帳戶中部署資源許可之前，先預覽資源的 IAM Access Analyzer 調查結果。

```
aws accessanalyzer create-access-preview \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --configurations file://myfile.json
```

myfile.json 的內容：

```
{
    "arn:aws:s3:::amzn-s3-demo-bucket": {
        "s3Bucket": {
            "bucketPolicy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Principal\":{\"AWS\":[\"arn:aws:iam::111122223333:root\"]},\"Action
\":[\"s3:PutObject\",\"s3:PutObjectAcl\"],\"Resource\":\"arn:aws:s3:::amzn-s3-demo-
bucket/*\"}]}",
            "bucketPublicAccessBlock": {
                "ignorePublicAcls": true,
                "restrictPublicBuckets": true
            },
```

```
            "bucketAclGrants": [
                {
                    "grantee": {
                        "id":
 "79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be"
                    },
                    "permission": "READ"
                }
            ]
        }
    }
}
```

輸出：

```
{
    "id": "3c65eb13-6ef9-4629-8919-a32043619e6b"
}
```

如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer APIs 預覽存取權。AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAccessPreview。

create-analyzer

以下程式碼範例顯示如何使用 create-analyzer。

AWS CLI

建立分析器

下列create-analyzer範例會在您的帳戶中建立分析器 AWS 。

```
aws accessanalyzer create-analyzer \
    --analyzer-name example \
    --type ACCOUNT
```

輸出：

```
{
```

```
    "arn": "arn:aws:access-analyzer:us-east-2:111122223333:analyzer/example"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的 AWS Identity and Access Management Access Analyzer 調查結果入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAnalyzer。

create-archive-rule

以下程式碼範例顯示如何使用 create-archive-rule。

AWS CLI

為指定的分析器建立封存規則

下列create-archive-rule範例會為您 AWS 帳戶中指定的分析器建立封存規則。

```
aws accessanalyzer create-archive-rule \
    --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \
    --rule-name MyRule \
    --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":
 ["AWS::IAM::Role"]}}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的封存規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateArchiveRule。

delete-analyzer

以下程式碼範例顯示如何使用 delete-analyzer。

AWS CLI

刪除指定的分析器

下列delete-analyzer範例會刪除您 AWS 帳戶中指定的分析器。

```
aws accessanalyzer delete-analyzer \
```

```
    --analyzer-name example
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的封存規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAnalyzer。

delete-archive-rule

以下程式碼範例顯示如何使用 delete-archive-rule。

AWS CLI

刪除指定的封存規則

下列delete-archive-rule範例會刪除您 AWS 帳戶中指定的封存規則。

```
aws accessanalyzer delete-archive-rule \
    --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \
    --rule-name MyRule
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的封存規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteArchiveRule。

get-access-preview

以下程式碼範例顯示如何使用 get-access-preview。

AWS CLI

擷取指定分析器存取預覽的相關資訊

下列get-access-preview範例會擷取您 AWS 帳戶中指定分析器之存取預覽的相關資訊。

```
aws accessanalyzer get-access-preview \
    --access-preview-id 3c65eb13-6ef9-4629-8919-a32043619e6b \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account
```

輸出：

```
{
    "accessPreview": {
        "id": "3c65eb13-6ef9-4629-8919-a32043619e6b",
        "analyzerArn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account",
        "configurations": {
            "arn:aws:s3:::amzn-s3-demo-bucket": {
                "s3Bucket": {
                    "bucketPolicy": "{\"Version\":\"2012-10-17\",\"Statement\":
[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":[\"arn:aws:iam::111122223333:root\"]},
\"Action\":[\"s3:PutObject\",\"s3:PutObjectAcl\"],\"Resource\":\"arn:aws:s3:::amzn-
s3-demo-bucket/*\"}]}",
                    "bucketAclGrants": [
                        {
                            "permission": "READ",
                            "grantee": {
                                "id":
 "79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be"
                            }
                        }
                    ],
                    "bucketPublicAccessBlock": {
                        "ignorePublicAcls": true,
                        "restrictPublicBuckets": true
                    }
                }
            }
        },
        "createdAt": "2024-02-17T00:18:44+00:00",
        "status": "COMPLETED"
    }
}
```

如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer APIs 預覽存取權。AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAccessPreview。

get-analyzed-resource

以下程式碼範例顯示如何使用 get-analyzed-resource。

AWS CLI

擷取已分析資源的相關資訊

下列get-analyzed-resource範例會擷取您 AWS 帳戶中已分析之資源的相關資訊。

```
aws accessanalyzer get-analyzed-resource \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --resource-arn arn:aws:s3:::amzn-s3-demo-bucket
```

輸出：

```
{
    "resource": {
        "analyzedAt": "2024-02-15T18:01:53.002000+00:00",
        "isPublic": false,
        "resourceArn": "arn:aws:s3:::amzn-s3-demo-bucket",
        "resourceOwnerAccount": "111122223333",
        "resourceType": "AWS::S3::Bucket"
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用 AWS Identity and Access Management Access Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAnalyzedResource。

get-analyzer

以下程式碼範例顯示如何使用 get-analyzer。

AWS CLI

擷取指定分析器的相關資訊

下列get-analyzer範例會擷取您 AWS 帳戶中指定分析器的相關資訊。

```
aws accessanalyzer get-analyzer \
    --analyzer-name ConsoleAnalyzer-account
```

輸出：

```
{
    "analyzer": {
        "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account",
        "createdAt": "2019-12-03T07:28:17+00:00",
        "lastResourceAnalyzed": "arn:aws:sns:us-west-2:111122223333:config-topic",
        "lastResourceAnalyzedAt": "2024-02-15T18:01:53.003000+00:00",
        "name": "ConsoleAnalyzer-account",
        "status": "ACTIVE",
        "tags": {
            "auto-delete": "no"
        },
        "type": "ACCOUNT"
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用 AWS Identity and Access Management Access Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAnalyzer。

get-archive-rule

以下程式碼範例顯示如何使用 get-archive-rule。

AWS CLI

擷取封存規則的相關資訊

下列get-archive-rule範例會擷取您 AWS 帳戶中封存規則的相關資訊。

```
aws accessanalyzer get-archive-rule \
    --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \
    --rule-name MyArchiveRule
```

輸出：

```
{
    "archiveRule": {
        "createdAt": "2024-02-15T00:49:27+00:00",
```

```
        "filter": {
            "resource": {
                "contains": [
                    "Cognito"
                ]
            },
            "resourceType": {
                "eq": [
                    "AWS::IAM::Role"
                ]
            }
        },
        "ruleName": "MyArchiveRule",
        "updatedAt": "2024-02-15T00:49:27+00:00"
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的封存規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetArchiveRule。

get-finding-v2

以下程式碼範例顯示如何使用 get-finding-v2。

AWS CLI

擷取指定問題清單的相關資訊

下列get-finding-v2範例會擷取您 AWS 帳戶中指定問題清單的相關資訊。

```
aws accessanalyzer get-finding-v2 \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-organization \
    --id 0910eedb-381e-4e95-adda-0d25c19e6e90
```

輸出：

```
{
    "findingDetails": [
        {
            "externalAccessDetails": {
```

```
                "action": [
                    "sts:AssumeRoleWithWebIdentity"
                ],
                "condition": {
                    "cognito-identity.amazonaws.com:aud": "us-
west-2:EXAMPLE0-0000-0000-0000-000000000000"
                },
                "isPublic": false,
                "principal": {
                    "Federated": "cognito-identity.amazonaws.com"
                }
            }
        }
    ],
    "resource": "arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role",
    "status": "ACTIVE",
    "error": null,
    "createdAt": "2021-02-26T21:17:50.905000+00:00",
    "resourceType": "AWS::IAM::Role",
    "findingType": "ExternalAccess",
    "resourceOwnerAccount": "111122223333",
    "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
    "id": "0910eedb-381e-4e95-adda-0d25c19e6e90",
    "updatedAt": "2021-02-26T21:17:50.905000+00:00"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的檢閱問題清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFindingV2。

get-finding

以下程式碼範例顯示如何使用 get-finding。

AWS CLI

擷取指定問題清單的相關資訊

下列get-finding範例會擷取您 AWS 帳戶中指定問題清單的相關資訊。

```
aws accessanalyzer get-finding \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-organization \
```

```
    --id 0910eedb-381e-4e95-adda-0d25c19e6e90
```

輸出：

```
{
    "finding": {
        "id": "0910eedb-381e-4e95-adda-0d25c19e6e90",
        "principal": {
            "Federated": "cognito-identity.amazonaws.com"
        },
        "action": [
            "sts:AssumeRoleWithWebIdentity"
        ],
        "resource": "arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role",
        "isPublic": false,
        "resourceType": "AWS::IAM::Role",
        "condition": {
            "cognito-identity.amazonaws.com:aud": "us-
west-2:EXAMPLE0-0000-0000-0000-000000000000"
        },
        "createdAt": "2021-02-26T21:17:50.905000+00:00",
        "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
        "updatedAt": "2021-02-26T21:17:50.905000+00:00",
        "status": "ACTIVE",
        "resourceOwnerAccount": "111122223333"
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的檢閱問題清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFinding。

get-generated-policy

以下程式碼範例顯示如何使用 get-generated-policy。

AWS CLI

擷取使用 `StartPolicyGeneration` API 產生的政策

下列get-generated-policy範例會擷取使用您 AWS 帳戶中的 StartPolicyGeneration API 產生的政策。

```
aws accessanalyzer get-generated-policy \
    --job-id c557dc4a-0338-4489-95dd-739014860ff9
```

輸出：

```
{
    "generatedPolicyResult": {
        "generatedPolicies": [
            {
                "policy": "{\"Version\":\"2012-10-17\",\"Statement\":
[{\"Sid\":\"SupportedServiceSid0\",\"Effect\":\"Allow\",\"Action\":
[\"access-analyzer:GetAnalyzer\",\"access-analyzer:ListAnalyzers\",
\"access-analyzer:ListArchiveRules\",\"access-analyzer:ListFindings
\",\"cloudtrail:DescribeTrails\",\"cloudtrail:GetEventDataStore\",
\"cloudtrail:GetEventSelectors\",\"cloudtrail:GetInsightSelectors
\",\"cloudtrail:GetTrailStatus\",\"cloudtrail:ListChannels\",
\"cloudtrail:ListEventDataStores\",\"cloudtrail:ListQueries\",\"cloudtrail:ListTags
\",\"cloudtrail:LookupEvents\",\"ec2:DescribeRegions\",\"iam:GetAccountSummary
\",\"iam:GetOpenIDConnectProvider\",\"iam:GetRole\",\"iam:ListAccessKeys\",
\"iam:ListAccountAliases\",\"iam:ListOpenIDConnectProviders\",\"iam:ListRoles
\",\"iam:ListSAMLProviders\",\"kms:ListAliases\",\"s3:GetBucketLocation\",
\"s3:ListAllMyBuckets\"],\"Resource\":\"*\"}]}"
            }
        ],
        "properties": {
            "cloudTrailProperties": {
                "endTime": "2024-02-14T22:44:40+00:00",
                "startTime": "2024-02-13T00:30:00+00:00",
                "trailProperties": [
                    {
                        "allRegions": true,
                        "cloudTrailArn": "arn:aws:cloudtrail:us-
west-2:111122223333:trail/my-trail",
                        "regions": []
                    }
                ]
            },
            "isComplete": false,
            "principalArn": "arn:aws:iam::111122223333:role/Admin"
        }
    },
    "jobDetails": {
        "completedOn": "2024-02-14T22:47:01+00:00",
```

```
        "jobId": "c557dc4a-0338-4489-95dd-739014860ff9",
        "startedOn": "2024-02-14T22:44:41+00:00",
        "status": "SUCCEEDED"
    }
}
```

如需更多資訊，請參閱「IAM 使用者指南」AWS 中的 IAM Access Analyzer 政策驗證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetGeneratedPolicy。

`list-access-preview-findings`

以下程式碼範例顯示如何使用 `list-access-preview-findings`。

AWS CLI

擷取指定存取預覽所產生的存取預覽調查結果清單

下列`list-access-preview-findings`範例會擷取您 AWS 帳戶中指定存取預覽所產生的存取預覽調查結果清單。

```
aws accessanalyzer list-access-preview-findings \
    --access-preview-id 3c65eb13-6ef9-4629-8919-a32043619e6b \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account
```

輸出：

```
{
    "findings": [
        {
            "id": "e22fc158-1c87-4c32-9464-e7f405ce8d74",
            "principal": {
                "AWS": "111122223333"
            },
            "action": [
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "condition": {},
            "resource": "arn:aws:s3:::amzn-s3-demo-bucket",
            "isPublic": false,
```

```
            "resourceType": "AWS::S3::Bucket",
            "createdAt": "2024-02-17T00:18:46+00:00",
            "changeType": "NEW",
            "status": "ACTIVE",
            "resourceOwnerAccount": "111122223333",
            "sources": [
                {
                    "type": "POLICY"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer APIs 預覽存取權。AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAccessPreviewFindings。

list-access-previews

以下程式碼範例顯示如何使用 list-access-previews。

AWS CLI

擷取指定分析器的存取預覽清單

下列list-access-previews範例會擷取您 AWS 帳戶中指定分析器的存取預覽清單。

```
aws accessanalyzer list-access-previews \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account
```

輸出：

```
{
    "accessPreviews": [
        {
            "id": "3c65eb13-6ef9-4629-8919-a32043619e6b",
            "analyzerArn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account",
            "createdAt": "2024-02-17T00:18:44+00:00",
```

```
            "status": "COMPLETED"
        }
    ]
}
```

如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer APIs 預覽存取權。AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAccessPreviews。

`list-analyzed-resources`

以下程式碼範例顯示如何使用 `list-analyzed-resources`。

AWS CLI

列出可用的小工具

下列`list-analyzed-resources`範例列出您 AWS 帳戶中可用的小工具。

```
aws accessanalyzer list-analyzed-resources \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --resource-type AWS::IAM::Role
```

輸出：

```
{
    "analyzedResources": [
        {
            "resourceArn": "arn:aws:sns:us-west-2:111122223333:Validation-Email",
            "resourceOwnerAccount": "111122223333",
            "resourceType": "AWS::SNS::Topic"
        },
        {
            "resourceArn": "arn:aws:sns:us-west-2:111122223333:admin-alerts",
            "resourceOwnerAccount": "111122223333",
            "resourceType": "AWS::SNS::Topic"
        },
        {
            "resourceArn": "arn:aws:sns:us-west-2:111122223333:config-topic",
            "resourceOwnerAccount": "111122223333",
```

```
            "resourceType": "AWS::SNS::Topic"
        },
        {
            "resourceArn": "arn:aws:sns:us-west-2:111122223333:inspector-topic",
            "resourceOwnerAccount": "111122223333",
            "resourceType": "AWS::SNS::Topic"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用 AWS Identity and Access Management Access Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAnalyzedResources。

`list-analyzers`

以下程式碼範例顯示如何使用 `list-analyzers`。

AWS CLI

擷取分析器清單

下列`list-analyzers`範例會擷取您 AWS 帳戶中的分析器清單。

```
aws accessanalyzer list-analyzers
```

輸出：

```
{
    "analyzers": [
        {
            "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
UnusedAccess-ConsoleAnalyzer-organization",
            "createdAt": "2024-02-15T00:46:40+00:00",
            "name": "UnusedAccess-ConsoleAnalyzer-organization",
            "status": "ACTIVE",
            "tags": {
                "auto-delete": "no"
            },
            "type": "ORGANIZATION_UNUSED_ACCESS"
        },
        {
```

```
            "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-organization",
            "createdAt": "2020-04-25T07:43:28+00:00",
            "lastResourceAnalyzed": "arn:aws:s3:::amzn-s3-demo-bucket",
            "lastResourceAnalyzedAt": "2024-02-15T21:51:56.517000+00:00",
            "name": "ConsoleAnalyzer-organization",
            "status": "ACTIVE",
            "tags": {
                "auto-delete": "no"
            },
            "type": "ORGANIZATION"
        },
        {
            "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account",
            "createdAt": "2019-12-03T07:28:17+00:00",
            "lastResourceAnalyzed": "arn:aws:sns:us-west-2:111122223333:config-
topic",
            "lastResourceAnalyzedAt": "2024-02-15T18:01:53.003000+00:00",
            "name": "ConsoleAnalyzer-account",
            "status": "ACTIVE",
            "tags": {
                "auto-delete": "no"
            },
            "type": "ACCOUNT"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用 AWS Identity and Access Management Access Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAnalyzers。

`list-archive-rules`

以下程式碼範例顯示如何使用 `list-archive-rules`。

AWS CLI

擷取為指定分析器建立的封存規則清單

下列`list-archive-rules`範例會擷取您 AWS 帳戶中指定分析器所建立的封存規則清單。

```
aws accessanalyzer list-archive-rules \
    --analyzer-name UnusedAccess-ConsoleAnalyzer-organization
```

輸出：

```
{
    "archiveRules": [
        {
            "createdAt": "2024-02-15T00:49:27+00:00",
            "filter": {
                "resource": {
                    "contains": [
                        "Cognito"
                    ]
                },
                "resourceType": {
                    "eq": [
                        "AWS::IAM::Role"
                    ]
                }
            },
            "ruleName": "MyArchiveRule",
            "updatedAt": "2024-02-15T00:49:27+00:00"
        },
        {
            "createdAt": "2024-02-15T23:27:45+00:00",
            "filter": {
                "findingType": {
                    "eq": [
                        "UnusedIAMUserAccessKey"
                    ]
                }
            },
            "ruleName": "ArchiveRule-56125a39-e517-4ff8-afb1-ef06f58db612",
            "updatedAt": "2024-02-15T23:27:45+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用 AWS Identity and Access Management Access Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListArchiveRules。

list-findings-v2

以下程式碼範例顯示如何使用 list-findings-v2。

AWS CLI

擷取指定分析器產生的調查結果清單

下列list-findings-v2範例會擷取您 AWS 帳戶中指定分析器所產生的調查結果清單。此範例會篩選結果，只包含名稱包含 的 IAM 角色Cognito。

```
aws accessanalyzer list-findings-v2 \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":
 ["AWS::IAM::Role"]}}'
```

輸出：

```
{
    "findings": [
        {
            "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
            "createdAt": "2021-02-26T21:17:24.710000+00:00",
            "id": "597f3bc2-3adc-4c18-9879-5c4b23485e46",
            "resource": "arn:aws:iam::111122223333:role/
Cognito_testpoolUnauth_Role",
            "resourceType": "AWS::IAM::Role",
            "resourceOwnerAccount": "111122223333",
            "status": "ACTIVE",
            "updatedAt": "2021-02-26T21:17:24.710000+00:00",
            "findingType": "ExternalAccess"
        },
        {
            "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
            "createdAt": "2021-02-26T21:17:50.905000+00:00",
            "id": "ce0e221a-85b9-4d52-91ff-d7678075442f",
            "resource": "arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role",
            "resourceType": "AWS::IAM::Role",
            "resourceOwnerAccount": "111122223333",
            "status": "ACTIVE",
            "updatedAt": "2021-02-26T21:17:50.905000+00:00",
            "findingType": "ExternalAccess"
```

```
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用 AWS Identity and Access Management Access Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFindingsV2。

`list-findings`

以下程式碼範例顯示如何使用 `list-findings`。

AWS CLI

擷取指定分析器產生的調查結果清單

下列`list-findings`範例會擷取您 AWS 帳戶中指定分析器所產生的調查結果清單。此範例會篩選結果，只包含名稱包含 的 IAM 角色`Cognito`。

```
aws accessanalyzer list-findings \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":
 ["AWS::IAM::Role"]}}'
```

輸出：

```
{
    "findings": [
        {
            "id": "597f3bc2-3adc-4c18-9879-5c4b23485e46",
            "principal": {
                "Federated": "cognito-identity.amazonaws.com"
            },
            "action": [
                "sts:AssumeRoleWithWebIdentity"
            ],
            "resource": "arn:aws:iam::111122223333:role/
Cognito_testpoolUnauth_Role",
            "isPublic": false,
```

```
            "resourceType": "AWS::IAM::Role",
            "condition": {
                "cognito-identity.amazonaws.com:aud": "us-
west-2:EXAMPLE0-0000-0000-0000-000000000000"
            },
            "createdAt": "2021-02-26T21:17:24.710000+00:00",
            "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
            "updatedAt": "2021-02-26T21:17:24.710000+00:00",
            "status": "ACTIVE",
            "resourceOwnerAccount": "111122223333"
        },
        {
            "id": "ce0e221a-85b9-4d52-91ff-d7678075442f",
            "principal": {
                "Federated": "cognito-identity.amazonaws.com"
            },
            "action": [
                "sts:AssumeRoleWithWebIdentity"
            ],
            "resource": "arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role",
            "isPublic": false,
            "resourceType": "AWS::IAM::Role",
            "condition": {
                "cognito-identity.amazonaws.com:aud": "us-
west-2:EXAMPLE0-0000-0000-0000-000000000000"
            },
            "createdAt": "2021-02-26T21:17:50.905000+00:00",
            "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
            "updatedAt": "2021-02-26T21:17:50.905000+00:00",
            "status": "ACTIVE",
            "resourceOwnerAccount": "111122223333"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用 AWS Identity and Access Management Access Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFindings。

`list-policy-generations`

以下程式碼範例顯示如何使用 `list-policy-generations`。

AWS CLI

列出過去七天內請求的所有政策產生

下列list-policy-generations範例列出您 AWS 帳戶中過去七天內請求的所有政策產生。

```
aws accessanalyzer list-policy-generations
```

輸出：

```
{
    "policyGenerations": [
        {
            "completedOn": "2024-02-14T23:43:38+00:00",
            "jobId": "923a56b0-ebb8-4e80-8a3c-a11ccfbcd6f2",
            "principalArn": "arn:aws:iam::111122223333:role/Admin",
            "startedOn": "2024-02-14T23:43:02+00:00",
            "status": "CANCELED"
        },
        {
            "completedOn": "2024-02-14T22:47:01+00:00",
            "jobId": "c557dc4a-0338-4489-95dd-739014860ff9",
            "principalArn": "arn:aws:iam::111122223333:role/Admin",
            "startedOn": "2024-02-14T22:44:41+00:00",
            "status": "SUCCEEDED"
        }
    ]
}
```

如需更多資訊，請參閱「IAM 使用者指南」AWS 中的 IAM Access Analyzer 政策驗證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPolicyGenerations。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

擷取套用至指定資源的標籤清單

下列list-tags-for-resource範例會擷取套用至 AWS 帳戶中指定資源的標籤清單。

```
aws accessanalyzer list-tags-for-resource \
    --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account
```

輸出：

```
{
    "tags": {
        "Zone-of-trust": "Account",
        "Name": "ConsoleAnalyzer"
    }
}
```

如需更多資訊，請參閱「IAM 使用者指南」AWS 中的 IAM Access Analyzer 政策驗證。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

start-policy-generation

以下程式碼範例顯示如何使用 start-policy-generation。

AWS CLI

開始政策產生請求

下列start-policy-generation範例會在您的帳戶中啟動政策產生請求 AWS 。

```
aws accessanalyzer start-policy-generation \
    --policy-generation-details '{"principalArn":"arn:aws:iam::111122223333:role/
Admin"}' \
    --cloud-trail-details file://myfile.json
```

myfile.json 的內容：

```
{
    "accessRole": "arn:aws:iam::111122223333:role/service-role/
AccessAnalyzerMonitorServiceRole",
    "startTime": "2024-02-13T00:30:00Z",
    "trails": [
        {
```

```
            "allRegions": true,
            "cloudTrailArn": "arn:aws:cloudtrail:us-west-2:111122223333:trail/my-
trail"
        }
    ]
}
```

輸出：

```
{
    "jobId": "c557dc4a-0338-4489-95dd-739014860ff9"
}
```

如需更多資訊，請參閱「IAM 使用者指南」AWS 中的 IAM Access Analyzer 政策驗證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartPolicyGeneration。

start-resource-scan

以下程式碼範例顯示如何使用 start-resource-scan。

AWS CLI

立即開始掃描套用至指定資源的政策

下列start-resource-scan範例會立即開始掃描套用至 AWS 帳戶中指定資源的政策。

```
aws accessanalyzer start-resource-scan \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --resource-arn arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role
```

此命令不會產生輸出。

如需更多資訊，請參閱「IAM 使用者指南」AWS 中的 IAM Access Analyzer 政策驗證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartResourceScan。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤新增至指定的資源

下列tag-resource範例會將標籤新增至您 AWS 帳戶中的指定資源。

```
aws accessanalyzer tag-resource \
    --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --tags Environment=dev,Purpose=testing
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用 AWS Identity and Access Management Access Analyzer。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從指定的資源移除標籤

下列untag-resource範例會從您 AWS 帳戶中的指定資源移除標籤。

```
aws accessanalyzer untag-resource \
    --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --tag-keys Environment Purpose
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用 AWS Identity and Access Management Access Analyzer。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-archive-rule

以下程式碼範例顯示如何使用 update-archive-rule。

AWS CLI

更新指定封存規則的條件和值

下列update-archive-rule範例會更新您 AWS 帳戶中指定封存規則的條件和值。

```
aws accessanalyzer update-archive-rule \
    --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \
    --rule-name MyArchiveRule \
    --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":
 ["AWS::IAM::Role"]}}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的封存規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateArchiveRule。

update-findings

以下程式碼範例顯示如何使用 update-findings。

AWS CLI

更新指定問題清單的狀態

下列update-findings範例會更新您 AWS 帳戶中指定問題清單的狀態。

```
aws accessanalyzer update-findings \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
UnusedAccess-ConsoleAnalyzer-organization \
    --ids 4f319ac3-2e0c-4dc4-bf51-7013a086b6ae 780d586a-2cce-4f72-aff6-359d450e7500
 \
    --status ARCHIVED
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的使用 AWS Identity and Access Management Access Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateFindings。

validate-policy

以下程式碼範例顯示如何使用 validate-policy。

AWS CLI

請求驗證政策並傳回問題清單

下列validate-policy範例會請求政策的驗證，並傳回問題清單。範例中的政策是用於 Web 聯合身分的 Amazon Cognito 角色的角色信任政策。由於使用不正確的擔任角色動作，從信任政策產生的調查結果與空Sid元素值和不相符的政策主體相關sts:AssumeRole。與 Cognito 搭配使用的正確擔任角色動作為 sts:AssumeRoleWithWebIdentity。

```
aws accessanalyzer validate-policy \
    --policy-document file://myfile.json \
    --policy-type RESOURCE_POLICY
```

myfile.json 的內容：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Federated": "cognito-identity.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:TagSession"
            ],
            "Condition": {
                "StringEquals": {
                    "cognito-identity.amazonaws.com:aud": "us-west-2_EXAMPLE"
                }
            }
        }
    ]
}
```

輸出：

```
{
    "findings": [
        {
            "findingDetails": "Add a value to the empty string in the Sid element.",
            "findingType": "SUGGESTION",
            "issueCode": "EMPTY_SID_VALUE",
            "learnMoreLink": "https://docs.aws.amazon.com/IAM/latest/UserGuide/
access-analyzer-reference-policy-checks.html#access-analyzer-reference-policy-
checks-suggestion-empty-sid-value",
            "locations": [
                {
                    "path": [
                        {
                            "value": "Statement"
                        },
                        {
                            "index": 0
                        },
                        {
                            "value": "Sid"
                        }
                    ],
                    "span": {
                        "end": {
                            "column": 21,
                            "line": 5,
                            "offset": 81
                        },
                        "start": {
                            "column": 19,
                            "line": 5,
                            "offset": 79
                        }
                    }
                }
            ]
        },
        {
            "findingDetails": "The sts:AssumeRole action is invalid with the
 following principal(s): cognito-identity.amazonaws.com. Use a SAML provider
 principal with the sts:AssumeRoleWithSAML action or use an OIDC provider principal
 with the sts:AssumeRoleWithWebIdentity action. Ensure the provider is Federated if
 you use either of the two options.",
```

```
            "findingType": "ERROR",
            "issueCode": "MISMATCHED_ACTION_FOR_PRINCIPAL",
            "learnMoreLink": "https://docs.aws.amazon.com/IAM/latest/UserGuide/
access-analyzer-reference-policy-checks.html#access-analyzer-reference-policy-
checks-error-mismatched-action-for-principal",
            "locations": [
                {
                    "path": [
                        {
                            "value": "Statement"
                        },
                        {
                            "index": 0
                        },
                        {
                            "value": "Action"
                        },
                        {
                            "index": 0
                        }
                    ],
                    "span": {
                        "end": {
                            "column": 32,
                            "line": 11,
                            "offset": 274
                        },
                        "start": {
                            "column": 16,
                            "line": 11,
                            "offset": 258
                        }
                    }
                },
                {
                    "path": [
                        {
                            "value": "Statement"
                        },
                        {
                            "index": 0
                        },
                        {
                            "value": "Principal"
```

```
                        },
                        {
                            "value": "Federated"
                        }
                    ],
                    "span": {
                        "end": {
                            "column": 61,
                            "line": 8,
                            "offset": 202
                        },
                        "start": {
                            "column": 29,
                            "line": 8,
                            "offset": 170
                        }
                    }
                }
            ]
        },
        {
            "findingDetails": "The following actions: sts:TagSession are not
 supported by the condition key cognito-identity.amazonaws.com:aud. The condition
 will not be evaluated for these actions. We recommend that you move these actions
 to a different statement without this condition key.",
            "findingType": "ERROR",
            "issueCode": "UNSUPPORTED_ACTION_FOR_CONDITION_KEY",
            "learnMoreLink": "https://docs.aws.amazon.com/IAM/latest/UserGuide/
access-analyzer-reference-policy-checks.html#access-analyzer-reference-policy-
checks-error-unsupported-action-for-condition-key",
            "locations": [
                {
                    "path": [
                        {
                            "value": "Statement"
                        },
                        {
                            "index": 0
                        },
                        {
                            "value": "Action"
                        },
                        {
                            "index": 1
```

```
                        }
                    ],
                    "span": {
                        "end": {
                            "column": 32,
                            "line": 12,
                            "offset": 308
                        },
                        "start": {
                            "column": 16,
                            "line": 12,
                            "offset": 292
                        }
                    }
                },
                {
                    "path": [
                        {
                            "value": "Statement"
                        },
                        {
                            "index": 0
                        },
                        {
                            "value": "Condition"
                        },
                        {
                            "value": "StringEquals"
                        },
                        {
                            "value": "cognito-identity.amazonaws.com:aud"
                        }
                    ],
                    "span": {
                        "end": {
                            "column": 79,
                            "line": 16,
                            "offset": 464
                        },
                        "start": {
                            "column": 58,
                            "line": 16,
                            "offset": 443
                        }
```

```
                    }
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的檢查以驗證政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ValidatePolicy。

使用 的映像建置器範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Image Builder 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-component`

以下程式碼範例顯示如何使用 `create-component`。

AWS CLI

建立元件

下列`create-component`範例會建立使用 JSON 文件檔案的元件，並以 YAML 格式參考上傳至 Amazon S3 儲存貯體的元件文件。

```
aws imagebuilder create-component \
    --cli-input-json file://create-component.json
```

`create-component.json` 的內容：

```
{
    "name": "MyExampleComponent",
    "semanticVersion": "2019.12.02",
    "description": "An example component that builds, validates and tests an image",
    "changeDescription": "Initial version.",
    "platform": "Windows",
    "uri": "s3://s3-bucket-name/s3-bucket-path/component.yaml"
}
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "componentBuildVersionArn": "arn:aws:imagebuilder:us-
west-2:123456789012:component/examplecomponent/2019.12.02/1"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateComponent。

create-distribution-configuration

以下程式碼範例顯示如何使用 create-distribution-configuration。

AWS CLI

建立分佈組態

下列create-distribution-configuration範例會使用 JSON 檔案建立分佈組態。

```
aws imagebuilder create-distribution-configuration \
    --cli-input-json file:/create-distribution-configuration.json
```

create-distribution-configuration.json 的內容：

```
{
    "name": "MyExampleDistribution",
```

```
    "description": "Copies AMI to eu-west-1",
    "distributions": [
        {
            "region": "us-west-2",
            "amiDistributionConfiguration": {
                "name": "Name {{imagebuilder:buildDate}}",
                "description": "An example image name with parameter references",
                "amiTags": {
                    "KeyName": "{{ssm:parameter_name}}"
                },
                "launchPermission": {
                    "userIds": [
                        "123456789012"
                    ]
                }
            }
        },
        {
            "region": "eu-west-1",
            "amiDistributionConfiguration": {
                "name": "My {{imagebuilder:buildVersion}} image
 {{imagebuilder:buildDate}}",
                "amiTags": {
                    "KeyName": "Value"
                },
                "launchPermission": {
                    "userIds": [
                        "123456789012"
                    ]
                }
            }
        }
    ]
}
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDistributionConfiguration。

create-image-pipeline

以下程式碼範例顯示如何使用 create-image-pipeline。

AWS CLI

建立映像管道

下列create-image-pipeline範例會使用 JSON 檔案建立映像管道。

```
aws imagebuilder create-image-pipeline \
    --cli-input-json file://create-image-pipeline.json
```

create-image-pipeline.json 的內容：

```
{
    "name": "MyWindows2016Pipeline",
    "description": "Builds Windows 2016 Images",
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03",
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",
    "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution",
    "imageTestsConfiguration": {
        "imageTestsEnabled": true,
        "timeoutMinutes": 60
    },
    "schedule": {
        "scheduleExpression": "cron(0 0 * * SUN)",
        "pipelineExecutionStartCondition":
 "EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"
    },
    "status": "ENABLED"
}
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateImagePipeline。

create-image-recipe

以下程式碼範例顯示如何使用 create-image-recipe。

AWS CLI

建立配方

下列create-image-recipe範例會使用 JSON 檔案建立映像配方。元件會依指定順序進行安裝。

```
aws imagebuilder create-image-recipe \
    --cli-input-json file://create-image-recipe.json
```

create-image-recipe.json 的內容：

```
{
    "name": "MyBasicRecipe",
    "description": "This example image recipe creates a Windows 2016 image.",
    "semanticVersion": "2019.12.03",
    "components":
    [
        {
            "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
myexamplecomponent/2019.12.02/1"
        },
        {
            "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
myimportedcomponent/1.0.0/1"
        }
```

```
    ],
    "parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/windows-server-2016-
english-full-base-x86/xxxx.x.x"
}
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateImageRecipe。

create-image

以下程式碼範例顯示如何使用 create-image。

AWS CLI

建立映像

下列create-image範例會建立 映像。

```
aws imagebuilder create-image \
    --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03 \
    --infrastructure-configuration-arn arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "imageBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/1"
```

```
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateImage。

create-infrastructure-configuration

以下程式碼範例顯示如何使用 create-infrastructure-configuration。

AWS CLI

建立基礎設施組態

下列create-infrastructure-configuration範例使用 JSON 檔案建立基礎設施組態。

```
aws imagebuilder create-infrastructure-configuration \
    --cli-input-json file://create-infrastructure-configuration.json
```

create-infrastructure-configuration.json 的內容：

```
{
    "name": "MyExampleInfrastructure",
    "description": "An example that will retain instances of failed builds",
    "instanceTypes": [
        "m5.large", "m5.xlarge"
    ],
    "instanceProfileName": "EC2InstanceProfileForImageBuilder",
    "securityGroupIds": [
        "sg-a1b2c3d4"
    ],
    "subnetId": "subnet-a1b2c3d4",
    "logging": {
        "s3Logs": {
            "s3BucketName": "bucket-name",
            "s3KeyPrefix": "bucket-path"
        }
    },
    "keyPair": "key-pair-name",
    "terminateInstanceOnFailure": false,
    "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:sns-topic-name"
```

```
}
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateInfrastructureConfiguration。

`delete-component`

以下程式碼範例顯示如何使用 `delete-component`。

AWS CLI

刪除元件

下列`delete-component`範例會透過指定元件建置版本 ARN 來刪除元件建置版本。

```
aws imagebuilder delete-component \
    --component-build-version-arn arn:aws:imagebuilder:us-
west-2:123456789012:component/myexamplecomponent/2019.12.02/1
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "componentBuildVersionArn": "arn:aws:imagebuilder:us-
west-2:123456789012:component/myexamplecomponent/2019.12.02/1"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteComponent。

delete-image-pipeline

以下程式碼範例顯示如何使用 delete-image-pipeline。

AWS CLI

刪除映像管道

下列delete-image-pipeline範例透過指定映像管道的 ARN 來刪除映像管道。

```
aws imagebuilder delete-image-pipeline \
    --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/my-example-pipeline
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/mywindows2016pipeline"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteImagePipeline。

delete-image-recipe

以下程式碼範例顯示如何使用 delete-image-recipe。

AWS CLI

刪除映像配方

下列delete-image-recipe範例會透過指定映像配方的 ARN 來刪除映像配方。

```
aws imagebuilder delete-image-recipe \
```

```
    --image-recipe-arn arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/
mybasicrecipe/2019.12.03
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteImageRecipe。

delete-image

以下程式碼範例顯示如何使用 delete-image。

AWS CLI

刪除映像

下列delete-image範例會透過指定映像建置版本 ARN 來刪除映像建置版本。

```
aws imagebuilder delete-image \
    --image-build-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/my-
example-image/2019.12.02/1
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/1"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteImage。

delete-infrastructure-configuration

以下程式碼範例顯示如何使用 delete-infrastructure-configuration。

AWS CLI

刪除基礎設施組態

下列delete-infrastructure-configuration範例會透過指定映像管道的 ARN 來刪除映像管道。

```
aws imagebuilder delete-infrastructure-configuration \
    --infrastructure-configuration-arn arn:aws:imagebuilder:us-
east-1:123456789012:infrastructure-configuration/myexampleinfrastructure
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteInfrastructureConfiguration。

get-component-policy

以下程式碼範例顯示如何使用 get-component-policy。

AWS CLI

取得元件政策詳細資訊

下列get-component-policy範例透過指定元件政策的 ARN 來列出元件政策的詳細資訊。

```
aws imagebuilder get-component-policy \
```

```
    --component-arn arn:aws:imagebuilder:us-west-2:123456789012:component/my-
example-component/2019.12.03/1
```

輸出：

```
{
    "Policy": "{ "Version": "2012-10-17", "Statement": [ { "Effect":
 "Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action":
 [ "imagebuilder:GetComponent", "imagebuilder:ListComponents" ], "Resource":
 [ "arn:aws:imagebuilder:us-west-2:123456789012:component/my-example-
component/2019.12.03/1" ] } ] }"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI <https://docs.aws.amazon.com/imagebuilder/latest/userguide/managing-image-builder-cli.html>`__ 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetComponentPolicy。

get-component

以下程式碼範例顯示如何使用 get-component。

AWS CLI

取得元件詳細資訊

下列get-component範例透過指定元件的 ARN 來列出元件的詳細資訊。

```
aws imagebuilder get-component \
    --component-build-version-arn arn:aws:imagebuilder:us-
west-2:123456789012:component/component-name/1.0.0/1
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "component": {
        "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/component-
name/1.0.0/1",
```

```
        "name": "component-name",
        "version": "1.0.0",
        "type": "TEST",
        "platform": "Linux",
        "owner": "123456789012",
        "data": "name: HelloWorldTestingDocument\ndescription: This is hello world
 testing document.\nschemaVersion: 1.0\n\nphases:\n  - name: test\n    steps:\n
   - name: HelloWorldStep\n        action: ExecuteBash\n        inputs:\n
 commands:\n            - echo \"Hello World! Test.\"\n",
        "encrypted": true,
        "dateCreated": "2020-01-27T20:43:30.306Z",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetComponent。

get-distribution-configuration

以下程式碼範例顯示如何使用 get-distribution-configuration。

AWS CLI

取得分佈組態的詳細資訊

下列get-distribution-configuration範例透過指定分佈組態的 ARN 來顯示分佈組態的詳細資訊。

```
aws imagebuilder get-distribution-configuration \
    --distribution-configuration-arn arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "distributionConfiguration": {
        "arn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-
configuration/myexampledistribution",
```

```
        "name": "MyExampleDistribution",
        "description": "Copies AMI to eu-west-1 and exports to S3",
        "distributions": [
            {
                "region": "us-west-2",
                "amiDistributionConfiguration": {
                    "name": "Name {{imagebuilder:buildDate}}",
                    "description": "An example image name with parameter
 references",
                    "amiTags": {
                        "KeyName": "{{ssm:parameter_name}}"
                    },
                    "launchPermission": {
                        "userIds": [
                            "123456789012"
                        ]
                    }
                }
            },
            {
                "region": "eu-west-1",
                "amiDistributionConfiguration": {
                    "name": "My {{imagebuilder:buildVersion}} image
 {{imagebuilder:buildDate}}",
                    "amiTags": {
                        "KeyName": "Value"
                    },
                    "launchPermission": {
                        "userIds": [
                            "123456789012"
                        ]
                    }
                }
            }
        ],
        "dateCreated": "2020-02-19T18:40:10.529Z",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDistributionConfiguration。

get-image-pipeline

以下程式碼範例顯示如何使用 get-image-pipeline。

AWS CLI

若要取得映像管道詳細資訊

下列get-image-pipeline範例透過指定映像管道的 ARN 來列出其詳細資訊。

```
aws imagebuilder get-image-pipeline \
    --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imagePipeline": {
        "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
        "name": "MyWindows2016Pipeline",
        "description": "Builds Windows 2016 Images",
        "platform": "Windows",
        "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03",
        "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",
        "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution",
        "imageTestsConfiguration": {
            "imageTestsEnabled": true,
            "timeoutMinutes": 60
        },
        "schedule": {
            "scheduleExpression": "cron(0 0 * * SUN)",
            "pipelineExecutionStartCondition":
 "EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"
        },
        "status": "ENABLED",
        "dateCreated": "2020-02-19T19:04:01.253Z",
        "dateUpdated": "2020-02-19T19:04:01.253Z",
        "tags": {}
    }
```

```
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetImagePipeline。

get-image-policy

以下程式碼範例顯示如何使用 get-image-policy。

AWS CLI

若要取得映像政策詳細資訊

下列get-image-policy範例會指定影像政策的 ARN，以列出影像政策的詳細資訊。

```
aws imagebuilder get-image-policy \
    --image-arn arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-image/2019.12.03/1
```

輸出：

```
{
    "Policy": "{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow",
 "Principal": { "AWS": [ "123456789012" ] }, "Action": [ "imagebuilder:GetImage",
 "imagebuilder:ListImages" ], "Resource": [ "arn:aws:imagebuilder:us-
west-2:123456789012:image/my-example-image/2019.12.03/1" ] } ] }"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetImagePolicy。

get-image-recipe-policy

以下程式碼範例顯示如何使用 get-image-recipe-policy。

AWS CLI

若要取得映像配方政策詳細資訊

下列get-image-recipe-policy範例會指定映像配方政策的 ARN，以列出其詳細資訊。

```
aws imagebuilder get-image-recipe-policy \
    --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-image-recipe/2019.12.03/1
```

輸出：

```
{
    "Policy": "{ "Version": "2012-10-17", "Statement": [ { "Effect":
 "Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action":
 [ "imagebuilder:GetImageRecipe", "imagebuilder:ListImageRecipes" ], "Resource":
 [ "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-image-recipe/2019.12.03/1" ] } ] }"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetImageRecipePolicy。

get-image

以下程式碼範例顯示如何使用 get-image。

AWS CLI

若要取得影像詳細資訊

下列get-image範例會指定影像的 ARN，以列出影像的詳細資訊。

```
aws imagebuilder get-image \
    --image-build-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/mybasicrecipe/2019.12.03/1
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "image": {
        "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/mybasicrecipe/2019.12.03/1",
```

```
        "name": "MyBasicRecipe",
        "version": "2019.12.03/1",
        "platform": "Windows",
        "state": {
            "status": "BUILDING"
        },
        "imageRecipe": {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03",
            "name": "MyBasicRecipe",
            "description": "This example image recipe creates a Windows 2016
 image.",
            "platform": "Windows",
            "version": "2019.12.03",
            "components": [
                {
                    "componentArn": "arn:aws:imagebuilder:us-
west-2:123456789012:component/myexamplecomponent/2019.12.02/1"
                },
                {
                    "componentArn": "arn:aws:imagebuilder:us-
west-2:123456789012:component/myimportedcomponent/1.0.0/1"
                }
            ],
            "parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/windows-
server-2016-english-full-base-x86/2019.12.17/1",
            "dateCreated": "2020-02-14T19:46:16.904Z",
            "tags": {}
        },
        "infrastructureConfiguration": {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-
configuration/myexampleinfrastructure",
            "name": "MyExampleInfrastructure",
            "description": "An example that will retain instances of failed builds",
            "instanceTypes": [
                "m5.large",
                "m5.xlarge"
            ],
            "instanceProfileName": "EC2InstanceProfileForImageFactory",
            "securityGroupIds": [
                "sg-a1b2c3d4"
            ],
            "subnetId": "subnet-a1b2c3d4",
            "logging": {
```

```
                "s3Logs": {
                    "s3BucketName": "bucket-name",
                    "s3KeyPrefix": "bucket-path"
                }
            },
            "keyPair": "Sam",
            "terminateInstanceOnFailure": false,
            "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:sns-name",
            "dateCreated": "2020-02-14T21:21:05.098Z",
            "tags": {}
        },
        "imageTestsConfiguration": {
            "imageTestsEnabled": true,
            "timeoutMinutes": 720
        },
        "dateCreated": "2020-02-14T23:14:13.597Z",
        "outputResources": {
            "amis": []
        },
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetImage。

get-infrastructure-configuration

以下程式碼範例顯示如何使用 get-infrastructure-configuration。

AWS CLI

取得基礎設施組態詳細資訊

下列get-infrastructure-configuration範例會指定基礎設施組態的 ARN，以列出其詳細資訊。

```
aws imagebuilder get-infrastructure-configuration \
    --infrastructure-configuration-arn arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "infrastructureConfiguration": {
        "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-
configuration/myexampleinfrastructure",
        "name": "MyExampleInfrastructure",
        "description": "An example that will retain instances of failed builds",
        "instanceTypes": [
            "m5.large",
            "m5.xlarge"
        ],
        "instanceProfileName": "EC2InstanceProfileForImageBuilder",
        "securityGroupIds": [
            "sg-a48c95ef"
        ],
        "subnetId": "subnet-a48c95ef",
        "logging": {
            "s3Logs": {
                "s3BucketName": "bucket-name",
                "s3KeyPrefix": "bucket-path"
            }
        },
        "keyPair": "Name",
        "terminateInstanceOnFailure": false,
        "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:sns-name",
        "dateCreated": "2020-02-19T19:11:51.858Z",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInfrastructureConfiguration。

`import-component`

以下程式碼範例顯示如何使用 `import-component`。

AWS CLI

匯入元件

下列import-component範例會使用 JSON 檔案匯入預先存在的指令碼。

```
aws imagebuilder import-component \
    --cli-input-json file://import-component.json
```

import-component.json 的內容：

```
{
    "name": "MyImportedComponent",
    "semanticVersion": "1.0.0",
    "description": "An example of how to import a component",
    "changeDescription": "First commit message.",
    "format": "SHELL",
    "platform": "Windows",
    "type": "BUILD",
    "uri": "s3://s3-bucket-name/s3-bucket-path/component.yaml"
}
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "componentBuildVersionArn": "arn:aws:imagebuilder:us-
west-2:123456789012:component/myimportedcomponent/1.0.0/1"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ImportComponent。

list-component-build-versions

以下程式碼範例顯示如何使用 list-component-build-versions。

AWS CLI

列出元件建置版本

下列list-component-build-versions範例列出具有特定語意版本的元件建置版本。

```
aws imagebuilder list-component-build-versions --component-
version-arn arn:aws:imagebuilder:us-west-2:123456789012:component/
myexamplecomponent/2019.12.02
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "componentSummaryList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
myexamplecomponent/2019.12.02/1",
            "name": "MyExampleComponent",
            "version": "2019.12.02",
            "platform": "Windows",
            "type": "BUILD",
            "owner": "123456789012",
            "description": "An example component that builds, validates and tests an
 image",
            "changeDescription": "Initial version.",
            "dateCreated": "2020-02-19T18:53:45.940Z",
            "tags": {
                "KeyName": "KeyValue"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListComponentBuildVersions。

list-components

以下程式碼範例顯示如何使用 list-components。

AWS CLI

列出所有元件語意版本

下列list-components範例列出您有權存取的所有元件語意版本。您可以選擇性地篩選是否列出您擁有的元件、Amazon 擁有的元件，還是其他帳戶已與您共用的元件。

```
aws imagebuilder list-components
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "componentVersionList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/component-
name/1.0.0",
            "name": "component-name",
            "version": "1.0.0",
            "platform": "Linux",
            "type": "TEST",
            "owner": "123456789012",
            "dateCreated": "2020-01-27T20:43:30.306Z"
        }
    ]
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListComponents。

list-distribution-configurations

以下程式碼範例顯示如何使用 list-distribution-configurations。

AWS CLI

列出分佈

下列list-distribution-configurations範例列出您的所有分佈。

```
aws imagebuilder list-distribution-configurations
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "distributionConfigurationSummaryList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-
configuration/myexampledistribution",
            "name": "MyExampleDistribution",
            "description": "Copies AMI to eu-west-1 and exports to S3",
            "dateCreated": "2020-02-19T18:40:10.529Z",
            "tags": {
                "KeyName": "KeyValue"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDistributionConfigurations。

list-image-build-versions

以下程式碼範例顯示如何使用 list-image-build-versions。

AWS CLI

列出映像建置版本

下列list-image-build-versions範例列出具有語意版本的所有映像建置版本。

```
aws imagebuilder list-image-build-versions \
    --image-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03
```

輸出：

```
{
```

```
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageSummaryList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/7",
            "name": "MyBasicRecipe",
            "version": "2019.12.03/7",
            "platform": "Windows",
            "state": {
                "status": "FAILED",
                "reason": "Can't start SSM Automation for arn
 arn:aws:imagebuilder:us-west-2:123456789012:image/mybasicrecipe/2019.12.03/7 during
 building. Parameter \"iamInstanceProfileName\" has a null value."
            },
            "owner": "123456789012",
            "dateCreated": "2020-02-19T18:56:11.511Z",
            "outputResources": {
                "amis": []
            },
            "tags": {}
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/6",
            "name": "MyBasicRecipe",
            "version": "2019.12.03/6",
            "platform": "Windows",
            "state": {
                "status": "FAILED",
                "reason": "An internal error has occurred."
            },
            "owner": "123456789012",
            "dateCreated": "2020-02-18T22:49:08.142Z",
            "outputResources": {
                "amis": [
                    {
                        "region": "us-west-2",
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "MyBasicRecipe 2020-02-18T22-49-38.704Z",
                        "description": "This example image recipe creates a Windows
 2016 image."
                    },
                    {
                        "region": "us-west-2",
```

```
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "Name 2020-02-18T22-49-08.131Z",
                        "description": "Copies AMI to eu-west-2 and exports to S3"
                    },
                    {
                        "region": "eu-west-2",
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "My 6 image 2020-02-18T22-49-08.131Z",
                        "description": "Copies AMI to eu-west-2 and exports to S3"
                    }
                ]
            },
            "tags": {}
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/5",
            "name": "MyBasicRecipe",
            "version": "2019.12.03/5",
            "platform": "Windows",
            "state": {
                "status": "AVAILABLE"
            },
            "owner": "123456789012",
            "dateCreated": "2020-02-18T16:51:48.403Z",
            "outputResources": {
                "amis": [
                    {
                        "region": "us-west-2",
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "MyBasicRecipe 2020-02-18T16-52-18.965Z",
                        "description": "This example image recipe creates a Windows
 2016 image."
                    }
                ]
            },
            "tags": {}
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/4",
            "name": "MyBasicRecipe",
            "version": "2019.12.03/4",
            "platform": "Windows",
```

```
            "state": {
                "status": "AVAILABLE"
            },
            "owner": "123456789012",
            "dateCreated": "2020-02-18T16:50:01.827Z",
            "outputResources": {
                "amis": [
                    {
                        "region": "us-west-2",
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "MyBasicRecipe 2020-02-18T16-50-32.280Z",
                        "description": "This example image recipe creates a Windows
 2016 image."
                    }
                ]
            },
            "tags": {}
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/3",
            "name": "MyBasicRecipe",
            "version": "2019.12.03/3",
            "platform": "Windows",
            "state": {
                "status": "AVAILABLE"
            },
            "owner": "123456789012",
            "dateCreated": "2020-02-14T23:14:13.597Z",
            "outputResources": {
                "amis": [
                    {
                        "region": "us-west-2",
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "MyBasicRecipe 2020-02-14T23-14-44.243Z",
                        "description": "This example image recipe creates a Windows
 2016 image."
                    }
                ]
            },
            "tags": {}
        },
        {
```

```
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/2",
            "name": "MyBasicRecipe",
            "version": "2019.12.03/2",
            "platform": "Windows",
            "state": {
                "status": "FAILED",
                "reason": "SSM execution 'a1b2c3d4-5678-90ab-cdef-EXAMPLE11111'
 failed with status = 'Failed' and failure message = 'Step fails when it is
 verifying the command has completed. Command a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
 returns unexpected invocation result: \n{Status=[Failed], ResponseCode=[1],
 Output=[\n----------ERROR-------\nfailed to run commands: exit status 1],
 OutputPayload=[{\"Status\":\"Failed\",\"ResponseCode\":1,\"Output\":\"\
\n----------ERROR-------\\nfailed to run commands: exit status 1\",\"CommandId\":
\"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111\"}], CommandId=[a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111]}. Please refer to Automation Service Troubleshooting Guide for more
 diagnosis details.'"
            },
            "owner": "123456789012",
            "dateCreated": "2020-02-14T22:57:42.593Z",
            "outputResources": {
                "amis": []
            },
            "tags": {}
        }
    ]
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListImageBuildVersions。

`list-image-pipeline-images`

以下程式碼範例顯示如何使用 `list-image-pipeline-images`。

AWS CLI

列出映像管道映像

下列`list-image-pipeline-images`範例列出由特定映像管道建立的所有映像。

```
aws imagebuilder list-image-pipeline-images \
    --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imagePipelineList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
            "name": "MyWindows2016Pipeline",
            "description": "Builds Windows 2016 Images",
            "platform": "Windows",
            "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-
recipe/mybasicrecipe/2019.12.03",
            "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",
            "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution",
            "imageTestsConfiguration": {
                "imageTestsEnabled": true,
                "timeoutMinutes": 60
            },
            "schedule": {
                "scheduleExpression": "cron(0 0 * * SUN)",
                "pipelineExecutionStartCondition":
 "EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"
            },
            "status": "ENABLED",
            "dateCreated": "2020-02-19T19:04:01.253Z",
            "dateUpdated": "2020-02-19T19:04:01.253Z",
            "tags": {
                "KeyName": "KeyValue"
            }
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/sam",
            "name": "PipelineName",
            "platform": "Linux",
            "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-
recipe/recipe-name-a1b2c3d45678/1.0.0",
```

```
            "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/infrastructureconfiguration-name-
a1b2c3d45678",
            "imageTestsConfiguration": {
                "imageTestsEnabled": true,
                "timeoutMinutes": 720
            },
            "status": "ENABLED",
            "dateCreated": "2019-12-16T18:19:02.068Z",
            "dateUpdated": "2019-12-16T18:19:02.068Z",
            "tags": {
                "KeyName": "KeyValue"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListImagePipelineImages。

`list-image-recipes`

以下程式碼範例顯示如何使用 `list-image-recipes`。

AWS CLI

列出映像配方

下列`list-image-recipes`範例列出所有映像配方。

```
aws imagebuilder list-image-recipes
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageRecipeSummaryList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03",
```

```
            "name": "MyBasicRecipe",
            "platform": "Windows",
            "owner": "123456789012",
            "parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/windows-
server-2016-english-full-base-x86/2019.x.x",
            "dateCreated": "2020-02-19T18:54:25.975Z",
            "tags": {
                "KeyName": "KeyValue"
            }
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/recipe-
name-a1b2c3d45678/1.0.0",
            "name": "recipe-name-a1b2c3d45678",
            "platform": "Linux",
            "owner": "123456789012",
            "parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/amazon-linux-2-
x86/2019.11.21",
            "dateCreated": "2019-12-16T18:19:00.120Z",
            "tags": {
                "KeyName": "KeyValue"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListImageRecipes。

`list-images`

以下程式碼範例顯示如何使用 `list-images`。

AWS CLI

列出影像

下列`list-images`範例列出您有權存取的所有語意版本。

```
aws imagebuilder list-images
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageVersionList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03",
            "name": "MyBasicRecipe",
            "version": "2019.12.03",
            "platform": "Windows",
            "owner": "123456789012",
            "dateCreated": "2020-02-14T21:29:18.810Z"
        }
    ]
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListImages。

list-infrastructure-configurations

以下程式碼範例顯示如何使用 list-infrastructure-configurations。

AWS CLI

列出基礎設施組態

下列list-infrastructure-configurations範例列出所有基礎設施組態。

```
aws imagebuilder list-infrastructure-configurations
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "infrastructureConfigurationSummaryList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-
configuration/myexampleinfrastructure",
```

```
            "name": "MyExampleInfrastructure",
            "description": "An example that will retain instances of failed builds",
            "dateCreated": "2020-02-19T19:11:51.858Z",
            "tags": {}
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-
configuration/infrastructureconfiguration-name-a1b2c3d45678",
            "name": "infrastructureConfiguration-name-a1b2c3d45678",
            "dateCreated": "2019-12-16T18:19:01.038Z",
            "tags": {
                "KeyName": "KeyValue"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListInfrastructureConfigurations。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出特定資源的標籤

下列`list-tags-for-resource`範例列出特定資源的所有標籤。

```
aws imagebuilder list-tags-for-resource \
    --resource-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline
```

輸出：

```
{
    "tags": {
        "KeyName": "KeyValue"
    }
```

```
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

put-component-policy

以下程式碼範例顯示如何使用 put-component-policy。

AWS CLI

將資源政策套用至元件

下列put-component-policy命令會將資源政策套用至組建元件，以啟用組建元件的跨帳戶共用。我們建議您使用 RAM CLI 命令 create-resource-share。如果您使用 EC2 Image Builder CLI 命令 put-component-policy，您還必須使用 RAM CLI 命令，promote-resource-share-create-from-policy才能讓與資源共用的所有主體看見資源。

```
aws imagebuilder put-component-policy \
    --component-arn arn:aws:imagebuilder:us-west-2:123456789012:component/
examplecomponent/2019.12.02/1 \
    --policy '{ "Version": "2012-10-17", "Statement": [ { "Effect":
 "Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action":
 [ "imagebuilder:GetComponent", "imagebuilder:ListComponents" ],
 "Resource": [ "arn:aws:imagebuilder:us-west-2:123456789012:component/
examplecomponent/2019.12.02/1" ] } ] }'
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
examplecomponent/2019.12.02/1"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutComponentPolicy。

put-image-policy

以下程式碼範例顯示如何使用 put-image-policy。

AWS CLI

將資源政策套用至映像

下列put-image-policy命令會將資源政策套用至映像，以啟用映像的跨帳戶共用。我們建議您使用 RAM CLI 命令 create-resource-share。如果您使用 EC2 Image Builder CLI 命令 put-image-policy，您還必須使用 RAM CLI 命令 promote-resource-share-create-from-policy，才能讓與資源共用的所有主體看見資源。

```
aws imagebuilder put-image-policy \
    --image-arn arn:aws:imagebuilder:us-west-2:123456789012:image/example-image/2019.12.02/1 \
    --policy '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action": [ "imagebuilder:GetImage", "imagebuilder:ListImages" ], "Resource": [ "arn:aws:imagebuilder:us-west-2:123456789012:image/example-image/2019.12.02/1" ] } ] }'
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/example-image/2019.12.02/1"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutImagePolicy。

put-image-recipe-policy

以下程式碼範例顯示如何使用 put-image-recipe-policy。

AWS CLI

將資源政策套用至映像配方

下列put-image-recipe-policy命令會將資源政策套用至映像配方，以啟用映像配方的跨帳戶共用。我們建議您使用 RAM CLI 命令 create-resource-share。如果您使用 EC2 Image Builder CLI 命令 put-image-recipe-policy，您還必須使用 RAM CLI 命令，promote-resource-share-create-from-policy才能讓與資源共用的所有主體看見資源。

```
aws imagebuilder put-image-recipe-policy \
    --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
example-image-recipe/2019.12.02 \
    --policy '{ "Version": "2012-10-17", "Statement": [ { "Effect":
 "Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action":
 [ "imagebuilder:GetImageRecipe", "imagebuilder:ListImageRecipes" ], "Resource":
 [ "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/example-image-
recipe/2019.12.02" ] } ] }'
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
example-image-recipe/2019.12.02/1"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutImageRecipePolicy。

start-image-pipeline-execution

以下程式碼範例顯示如何使用 start-image-pipeline-execution。

AWS CLI

手動啟動映像管道

下列start-image-pipeline-execution範例會手動啟動映像管道。

```
aws imagebuilder start-image-pipeline-execution \
    --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "imageBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/1"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartImagePipelineExecution。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

標記資源

下列tag-resource範例會使用 JSON 檔案，將資源新增至 EC2 Image Builder 並加上標籤。

```
aws imagebuilder tag-resource \
    --cli-input-json file://tag-resource.json
```

tag-resource.json 的內容：

```
{
    "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
    "tags": {
        "KeyName: "KeyValue"
    }
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源移除標籤

下列untag-resource範例使用 JSON 檔案從資源移除標籤。

```
aws imagebuilder untag-resource \
    --cli-input-json file://tag-resource.json
```

untag-resource.json 的內容：

```
{
    "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
    "tagKeys": [
        "KeyName"
    ]
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-distribution-configuration

以下程式碼範例顯示如何使用 update-distribution-configuration。

AWS CLI

更新分佈組態

下列update-distribution-configuration範例會使用 JSON 檔案更新分佈組態。

```
aws imagebuilder update-distribution-configuration \
    --cli-input-json file://update-distribution-configuration.json
```

update-distribution-configuration.json 的內容：

```
{
    "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution",
    "description": "Copies AMI to eu-west-2 and exports to S3",
    "distributions": [
        {
            "region": "us-west-2",
            "amiDistributionConfiguration": {
                "name": "Name {{imagebuilder:buildDate}}",
                "description": "An example image name with parameter references"
            }
        },
        {
            "region": "eu-west-2",
            "amiDistributionConfiguration": {
                "name": "My {{imagebuilder:buildVersion}} image
 {{imagebuilder:buildDate}}"
            }
        }
    ]
}
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDistributionConfiguration。

update-image-pipeline

以下程式碼範例顯示如何使用 update-image-pipeline。

AWS CLI

更新映像管道

下列update-image-pipeline範例會使用 JSON 檔案更新映像管道。

```
aws imagebuilder update-image-pipeline \
    --cli-input-json file://update-image-pipeline.json
```

update-image-pipeline.json 的內容：

```
{
    "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03",
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",
    "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution",
    "imageTestsConfiguration": {
        "imageTestsEnabled": true,
        "timeoutMinutes": 120
    },
    "schedule": {
        "scheduleExpression": "cron(0 0 * * MON)",
        "pipelineExecutionStartCondition":
 "EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"
    },
    "status": "DISABLED"
}
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateImagePipeline。

update-infrastructure-configuration

以下程式碼範例顯示如何使用 update-infrastructure-configuration。

AWS CLI

更新基礎設施組態

下列update-infrastructure-configuration範例會使用 JSON 檔案更新基礎設施組態。

```
aws imagebuilder update-infrastructure-configuration \
    --cli-input-json file:/update-infrastructure-configuration.json
```

update-infrastructure-configuration.json 的內容：

```
{
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",
    "description": "An example that will terminate instances of failed builds",
    "instanceTypes": [
        "m5.large", "m5.2xlarge"
    ],
    "instanceProfileName": "EC2InstanceProfileForImageFactory",
    "securityGroupIds": [
        "sg-a48c95ef"
    ],
    "subnetId": "subnet-a48c95ef",
    "logging": {
        "s3Logs": {
            "s3BucketName": "bucket-name",
            "s3KeyPrefix": "bucket-path"
        }
    },
    "terminateInstanceOnFailure": true,
    "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:sns-name"
}
```

輸出：

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

如需詳細資訊，請參閱 EC2 Image Builder 使用者指南中的使用 AWS CLI 設定和管理 EC2 Image Builder Image Pipeline。 EC2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateInfrastructureConfiguration。

使用 的 Incident Manager 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Incident Manager 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-replication-set`

以下程式碼範例顯示如何使用 `create-replication-set`。

AWS CLI

建立複寫集

下列`create-replication-set`範例會建立 Incident Manager 用來複寫和加密 Amazon Web Services 帳戶中資料的複寫集。此範例在建立複寫集時使用 us-east-1 和 us-east-2 區域。

```
aws ssm-incidents create-replication-set \
    --regions '{"us-east-1": {"sseKmsKeyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"}, "us-east-2":
 {"sseKmsKeyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"}}'
```

輸出：

```
{
    "replicationSetArns": [
        "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-
bb3f-413c-08df53673b57"
    ]
```

```
}
```

如需詳細資訊，請參閱《Event Manager 使用者指南》中的使用 Incident Manager 複寫集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateReplicationSet。

create-response-plan

以下程式碼範例顯示如何使用 create-response-plan。

AWS CLI

建立回應計畫

下列create-response-plan範例會建立具有指定詳細資訊的回應計劃。

```
aws ssm-incidents create-response-plan \
    --chat-channel '{"chatbotSns": ["arn:aws:sns:us-
east-1:111122223333:Standard_User"]}' \
    --display-name "Example response plan" \
    --incident-template '{"impact": 5, "title": "example-incident"}' \
    --name "example-response" \
    --actions '[{"ssmAutomation": {"documentName": "AWSIncidents-
CriticalIncidentRunbookTemplate", "documentVersion": "$DEFAULT",
 "roleArn": "arn:aws:iam::111122223333:role/aws-service-role/ssm-
incidents.amazonaws.com/AWSServiceRoleForIncidentManager", "targetAccount":
 "RESPONSE_PLAN_OWNER_ACCOUNT"}}]' \
    --engagements '["arn:aws:ssm-contacts:us-east-1:111122223333:contact/example"]'
```

輸出：

```
{
    "arn": "arn:aws:ssm-incidents::111122223333:response-plan/example-response"
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的事件準備。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateResponsePlan。

create-timeline-event

以下程式碼範例顯示如何使用 create-timeline-event。

AWS CLI

範例 1：建立自訂時間軸事件

下列create-timeline-event範例會在指定事件的指定時間建立自訂時間軸事件。

```
aws ssm-incidents create-timeline-event \
    --event-data "\"example timeline event\"" \
    --event-time 2022-10-01T20:30:00.000 \
    --event-type "Custom Event" \
    --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"
```

輸出：

```
{
    "eventId": "c0bcc885-a41d-eb01-b4ab-9d2deEXAMPLE",
    "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"
}
```

範例 2：使用事件備註建立時間軸事件

下列create-timeline-event範例會建立 'Incident notes' 面板中列出的時間軸事件。

```
aws ssm-incidents create-timeline-event \
     --event-data "\"New Note\"" \
     --event-type "Note" \
     --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Test/6cc46130-ca6c-3b38-68f1-f6abeEXAMPLE" \
     --event-time 2023-06-20T12:06:00.000 \
     --event-references '[{"resource":"arn:aws:ssm-incidents::111122223333:incident-
record/Test/6cc46130-ca6c-3b38-68f1-f6abeEXAMPLE"}]'
```

輸出：

```
{
    "eventId": "a41dc885-c0bc-b4ab-eb01-de9d2EXAMPLE",
    "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的事件詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTimelineEvent。

delete-incident-record

以下程式碼範例顯示如何使用 delete-incident-record。

AWS CLI

刪除事件記錄

下列delete-incident-record範例會刪除指定的事件記錄。

```
aws ssm-incidents delete-incident-record \
    --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的事件追蹤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteIncidentRecord。

delete-replication-set

以下程式碼範例顯示如何使用 delete-replication-set。

AWS CLI

刪除複寫集

下列delete-replication-set範例會從 Amazon Web Services 帳戶刪除複寫集。刪除複寫集也會刪除所有 Incident Manager 資料。這無法復原。

```
aws ssm-incidents delete-replication-set \
    --arn "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-bb3f-413c-08df53673b57"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Event Manager 使用者指南》中的使用 Incident Manager 複寫集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteReplicationSet。

delete-resource-policy

以下程式碼範例顯示如何使用 delete-resource-policy。

AWS CLI

刪除資源政策

下列delete-resource-policy範例會從回應計劃刪除資源政策。這將撤銷與回應計劃共用的委託人或組織的存取權。

```
aws ssm-incidents delete-resource-policy \
    --policy-id "be8b57191f0371f1c6827341aa3f0a03" \
    --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的使用共用聯絡人和回應計劃。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteResourcePolicy。

delete-response-plan

以下程式碼範例顯示如何使用 delete-response-plan。

AWS CLI

刪除回應計劃

下列delete-response-plan範例會刪除指定的回應計劃。

```
aws ssm-incidents delete-response-plan \
    --arn "arn:aws:ssm-incidents::111122223333:response-plan/example-response"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的事件準備。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteResponsePlan。

delete-timeline-event

以下程式碼範例顯示如何使用 delete-timeline-event。

AWS CLI

刪除時間軸事件

下列delete-timeline-event範例會從指定的事件記錄中刪除自訂時間軸事件。

```
aws ssm-incidents delete-timeline-event \
    --event-id "c0bcc885-a41d-eb01-b4ab-9d2de193643c" \
    --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的事件詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTimelineEvent。

get-incident-record

以下程式碼範例顯示如何使用 get-incident-record。

AWS CLI

取得事件記錄

下列get-incident-record範例會取得指定事件記錄的詳細資訊。

```
aws ssm-incidents get-incident-record \
    --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-
Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

輸出：

```
{
    "incidentRecord": {
        "arn": "arn:aws:ssm-incidents::111122223333:incident-record/Example-
Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308",
        "automationExecutions": [],
```

```
        "creationTime": "2021-05-21T18:16:57.579000+00:00",
        "dedupeString": "c4bcc812-85e7-938d-2b78-17181176ee1a",
        "impact": 5,
        "incidentRecordSource": {
            "createdBy": "arn:aws:iam::111122223333:user/draliatp",
            "invokedBy": "arn:aws:iam::111122223333:user/draliatp",
            "source": "aws.ssm-incidents.custom"
        },
        "lastModifiedBy": "arn:aws:iam::111122223333:user/draliatp",
        "lastModifiedTime": "2021-05-21T18:16:59.149000+00:00",
        "notificationTargets": [],
        "status": "OPEN",
        "title": "Example-Incident"
    }
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的事件詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIncidentRecord。

get-replication-set

以下程式碼範例顯示如何使用 get-replication-set。

AWS CLI

若要取得複寫集

下列get-replication-set範例取得 Incident Manager 用來複寫和加密 Amazon Web Services 帳戶中資料之複寫集的詳細資訊。

```
aws ssm-incidents get-replication-set \
    --arn "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-bb3f-413c-08df53673b57"
```

輸出：

```
{
    "replicationSet": {
        "createdBy": "arn:aws:sts::111122223333:assumed-role/Admin/username",
        "createdTime": "2021-05-14T17:57:22.010000+00:00",
        "deletionProtected": false,
```

```
        "lastModifiedBy": "arn:aws:sts::111122223333:assumed-role/Admin/username",
        "lastModifiedTime": "2021-05-14T17:57:22.010000+00:00",
        "regionMap": {
            "us-east-1": {
                "sseKmsKeyId": "DefaultKey",
                "status": "ACTIVE"
            },
            "us-east-2": {
                "sseKmsKeyId": "DefaultKey",
                "status": "ACTIVE",
                "statusMessage": "Tagging inaccessible"
            }
        },
        "status": "ACTIVE"
    }
}
```

如需詳細資訊，請參閱《Event Manager 使用者指南》中的使用 Incident Manager 複寫集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetReplicationSet。

get-resource-policies

以下程式碼範例顯示如何使用 get-resource-policies。

AWS CLI

列出回應計劃的資源政策

下列command-name範例列出與指定回應計劃相關聯的資源政策。

```
aws ssm-incidents get-resource-policies \
--resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

輸出：

```
{
    "resourcePolicies": [
        {
            "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid
\":\"d901b37a-dbb0-458a-8842-75575c464219-external-principals\",\"Effect\":
\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::222233334444:root\"},\"Action
```

```
\":[\"ssm-incidents:GetResponsePlan\",\"ssm-incidents:StartIncident\",\"ssm-
incidents:UpdateIncidentRecord\",\"ssm-incidents:GetIncidentRecord\",\"ssm-
incidents:CreateTimelineEvent\",\"ssm-incidents:UpdateTimelineEvent\",\"ssm-
incidents:GetTimelineEvent\",\"ssm-incidents:ListTimelineEvents\",\"ssm-
incidents:UpdateRelatedItems\",\"ssm-incidents:ListRelatedItems\"],\"Resource\":
[\"arn:aws:ssm-incidents:*:111122223333:response-plan/Example-Response-Plan\",
\"arn:aws:ssm-incidents:*:111122223333:incident-record/Example-Response-Plan/*
\"]}]}",
            "policyId": "be8b57191f0371f1c6827341aa3f0a03",
            "ramResourceShareRegion": "us-east-1"
        }
    ]
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的使用共用聯絡人和回應計劃。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResourcePolicies。

get-response-plan

以下程式碼範例顯示如何使用 get-response-plan。

AWS CLI

取得回應計劃的詳細資訊

下列command-name範例會取得您 AWS 帳戶中指定回應計劃的詳細資訊。

```
aws ssm-incidents get-response-plan \
    --arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

輸出：

```
{
    "actions": [
        {
            "ssmAutomation": {
                "documentName": "AWSIncidents-CriticalIncidentRunbookTemplate",
                "documentVersion": "$DEFAULT",
                "roleArn": "arn:aws:iam::111122223333:role/aws-service-role/ssm-
incidents.amazonaws.com/AWSServiceRoleForIncidentManager",
                "targetAccount": "RESPONSE_PLAN_OWNER_ACCOUNT"
            }
```

```
        }
    ],
    "arn": "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-
Plan",
    "chatChannel": {
        "chatbotSns": [
            "arn:aws:sns:us-east-1:111122223333:Standard_User"
        ]
    },
    "displayName": "Example response plan",
    "engagements": [
        "arn:aws:ssm-contacts:us-east-1:111122223333:contact/example"
    ],
    "incidentTemplate": {
        "impact": 5,
        "title": "Example-Incident"
    },
    "name": "Example-Response-Plan"
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的事件準備。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResponsePlan。

get-timeline-event

以下程式碼範例顯示如何使用 get-timeline-event。

AWS CLI

取得時間軸事件的詳細資訊

下列get-timeline-event範例會傳回指定時間軸事件的詳細資訊。

```
aws ssm-incidents get-timeline-event \
    --event-id 20bcc812-8a94-4cd7-520c-0ff742111424 \
    --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

輸出：

```
{
    "event": {
```

```
        "eventData": "\"Incident Started\"",
        "eventId": "20bcc812-8a94-4cd7-520c-0ff742111424",
        "eventTime": "2021-05-21T18:16:57+00:00",
        "eventType": "Custom Event",
        "eventUpdatedTime": "2021-05-21T18:16:59.944000+00:00",
        "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
    }
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的事件詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTimelineEvent。

`list-incident-records`

以下程式碼範例顯示如何使用 `list-incident-records`。

AWS CLI

列出事件記錄

下列command-name範例列出 Amazon Web Services 帳戶中的事件記錄。

```
aws ssm-incidents list-incident-records
```

輸出：

```
{
    "incidentRecordSummaries": [
        {
            "arn": "arn:aws:ssm-incidents::111122223333:incident-record/Example-
Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308",
            "creationTime": "2021-05-21T18:16:57.579000+00:00",
            "impact": 5,
            "incidentRecordSource": {
                "createdBy": "arn:aws:iam::111122223333:user/draliatp",
                "invokedBy": "arn:aws:iam::111122223333:user/draliatp",
                "source": "aws.ssm-incidents.custom"
            },
            "status": "OPEN",
            "title": "Example-Incident"
        }
```

```
    ]
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的事件清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListIncidentRecords。

`list-related-items`

以下程式碼範例顯示如何使用 `list-related-items`。

AWS CLI

列出相關項目

下列`list-related-items`範例列出指定事件的相關項目。

```
aws ssm-incidents list-related-items \
    --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

輸出：

```
{
    "relatedItems": [
        {
            "identifier": {
                "type": "OTHER",
                "value": {
                    "url": "https://console.aws.amazon.com/systems-manager/opsitems/
oi-8ef82158e190/workbench?region=us-east-1"
                }
            },
            "title": "Example related item"
        },
        {
            "identifier": {
                "type": "PARENT",
                "value": {
                    "arn": "arn:aws:ssm:us-east-1:111122223333:opsitem/
oi-8084126392ac"
                }
            },
```

```
            "title": "parentItem"
        }
    ]
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的事件詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRelatedItems。

list-replication-sets

以下程式碼範例顯示如何使用 list-replication-sets。

AWS CLI

列出複寫集

下列list-replication-set範例列出 Incident Manager 用來複寫和加密 AWS 帳戶中資料的複寫集。

```
aws ssm-incidents list-replication-sets
```

輸出：

```
{
    "replicationSetArns": [
        "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-
bb3f-413c-08df53673b57"
    ]
}
```

如需詳細資訊，請參閱《Event Manager 使用者指南》中的使用 Incident Manager 複寫集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListReplicationSets。

list-response-plans

以下程式碼範例顯示如何使用 list-response-plans。

AWS CLI

列出可用的回應計畫

下列list-response-plans範例列出 Amazon Web Services 帳戶中可用的回應計劃。

```
aws ssm-incidents list-response-plans
```

輸出：

```
{
    "responsePlanSummaries": [
        {
            "arn": "arn:aws:ssm-incidents::111122223333:response-plan/Example-
Response-Plan",
            "displayName": "Example response plan",
            "name": "Example-Response-Plan"
        }
    ]
}
```

如需詳細資訊，請參閱《Incident Manager 使用者指南》中的事件準備。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListResponsePlans。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出回應計劃的標籤

下列list-tags-for-resource範例列出與指定回應計劃相關聯的標籤。

```
aws ssm-incidents list-tags-for-resource \
    --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-
Response-Plan"
```

輸出：

```
{
    "tags": {
    "group1": "1"
    }
```

```
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的標記。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`list-timeline-events`

以下程式碼範例顯示如何使用 `list-timeline-events`。

AWS CLI

列出事件的時間軸事件

下列command-name範例列出指定事件的時間軸事件。

```
aws ssm-incidents list-timeline-events \
    --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

輸出：

```
{
    "eventSummaries": [
        {
            "eventId": "8cbcc889-35e1-a42d-2429-d6f100799915",
            "eventTime": "2021-05-21T22:36:13.766000+00:00",
            "eventType": "SSM Incident Record Update",
            "eventUpdatedTime": "2021-05-21T22:36:13.766000+00:00",
            "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
        },
        {
            "eventId": "a2bcc825-aab5-1787-c605-f9bb2640d85b",
            "eventTime": "2021-05-21T18:58:46.443000+00:00",
            "eventType": "SSM Incident Record Update",
            "eventUpdatedTime": "2021-05-21T18:58:46.443000+00:00",
            "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
        },
        {
            "eventId": "5abcc812-89c0-b0a8-9437-1c74223d4685",
            "eventTime": "2021-05-21T18:16:59.149000+00:00",
```

```
            "eventType": "SSM Incident Record Update",
            "eventUpdatedTime": "2021-05-21T18:16:59.149000+00:00",
            "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
        },
        {
            "eventId": "06bcc812-8820-405e-4065-8d2b14d29b92",
            "eventTime": "2021-05-21T18:16:58+00:00",
            "eventType": "SSM Automation Execution Start Failure for Incident",
            "eventUpdatedTime": "2021-05-21T18:16:58.689000+00:00",
            "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
        },
        {
            "eventId": "20bcc812-8a94-4cd7-520c-0ff742111424",
            "eventTime": "2021-05-21T18:16:57+00:00",
            "eventType": "Custom Event",
            "eventUpdatedTime": "2021-05-21T18:16:59.944000+00:00",
            "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
        },
        {
            "eventId": "c0bcc885-a41d-eb01-b4ab-9d2de193643c",
            "eventTime": "2020-10-01T20:30:00+00:00",
            "eventType": "Custom Event",
            "eventUpdatedTime": "2021-05-21T22:28:26.299000+00:00",
            "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
        }
    ]
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的事件詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTimelineEvents。

put-resource-policy

以下程式碼範例顯示如何使用 put-resource-policy。

AWS CLI

共用回應計劃和事件

下列command-name範例會將資源政策新增至 Example-Response-Plan，與指定的委託人共用回應計劃和相關事件。

```
aws ssm-incidents put-resource-policy \
    --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan" \
    --policy "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"ExampleResourcePolciy\",\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::222233334444:root\"},\"Action\":[\"ssm-incidents:GetResponsePlan\",\"ssm-incidents:StartIncident\",\"ssm-incidents:UpdateIncidentRecord\",\"ssm-incidents:GetIncidentRecord\",\"ssm-incidents:CreateTimelineEvent\",\"ssm-incidents:UpdateTimelineEvent\",\"ssm-incidents:GetTimelineEvent\",\"ssm-incidents:ListTimelineEvents\",\"ssm-incidents:UpdateRelatedItems\",\"ssm-incidents:ListRelatedItems\"],\"Resource\":[\"arn:aws:ssm-incidents:*:111122223333:response-plan/Example-Response-Plan\",\"arn:aws:ssm-incidents:*:111122223333:incident-record/Example-Response-Plan/*\"]}]}"
```

輸出：

```
{
    "policyId": "be8b57191f0371f1c6827341aa3f0a03"
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的使用共用聯絡人和回應計劃。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutResourcePolicy。

start-incident

以下程式碼範例顯示如何使用 start-incident。

AWS CLI

啟動事件

下列start-incident範例使用指定的回應計劃啟動事件。

```
aws ssm-incidents start-incident \
    --response-plan-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

輸出：

```
{
    "incidentRecordArn": "arn:aws:ssm-incidents::682428703967:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的事件建立。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartIncident。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

標記回應計劃

下列tag-resource範例使用提供的標籤鍵值對來標記指定的回應計劃。

```
aws ssm-incidents tag-resource \
    --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-
Response-Plan" \
    --tags '{"group1":"1"}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的標記。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從回應計劃中移除標籤

下列untag-resource範例會從回應計劃中移除指定的標籤。

```
aws ssm-incidents untag-resource \
    --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-
Response-Plan" \
```

```
    --tag-keys '["group1"]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的標記。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-deletion-protection

以下程式碼範例顯示如何使用 update-deletion-protection。

AWS CLI

更新複寫集刪除保護

下列update-deletion-protection範例會更新您帳戶中的刪除保護，以防止您刪除複寫集中的最後一個區域。

```
aws ssm-incidents update-deletion-protection \
    --arn "arn:aws:ssm-incidents::111122223333:replication-set/
a2bcc5c9-0f53-8047-7fef-c20749989b40" \
    --deletion-protected
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Event Manager 使用者指南》中的使用 Incident Manager 複寫集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDeletionProtection。

update-incident-record

以下程式碼範例顯示如何使用 update-incident-record。

AWS CLI

更新事件記錄

下列command-name範例會解析指定的事件。

```
aws ssm-incidents update-incident-record \
    --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-
Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308" \
```

```
    --status "RESOLVED"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的事件詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateIncidentRecord。

update-related-items

以下程式碼範例顯示如何使用 update-related-items。

AWS CLI

更新事件相關項目

下列update-related-item範例會從指定的事件記錄中移除相關項目。

```
aws ssm-incidents update-related-items \
    --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308" \
    --related-items-update '{"itemToRemove": {"type": "OTHER", "value": {"url":
 "https://console.aws.amazon.com/systems-manager/opsitems/oi-8ef82158e190/workbench?
region=us-east-1"}}}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Event Manager 使用者指南》中的事件詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRelatedItems。

update-replication-set

以下程式碼範例顯示如何使用 update-replication-set。

AWS CLI

更新複寫集

下列command-name範例會從複寫集刪除 us-east-2 區域。

```
aws ssm-incidents update-replication-set \
```

```
    --arn "arn:aws:ssm-incidents::111122223333:replication-set/
a2bcc5c9-0f53-8047-7fef-c20749989b40" \
    --actions '[{"deleteRegionAction": {"regionName": "us-east-2"}}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Event Manager 使用者指南》中的使用 Incident Manager 複寫集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateReplicationSet。

update-response-plan

以下程式碼範例顯示如何使用 update-response-plan。

AWS CLI

更新回應計畫

下列update-response-plan範例會從指定的回應計劃中移除聊天頻道。

```
aws ssm-incidents update-response-plan \
    --arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
 \
    --chat-channel '{"empty":{}}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的事件準備。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateResponsePlan。

update-timeline-event

以下程式碼範例顯示如何使用 update-timeline-event。

AWS CLI

更新時間軸事件

下列update-timeline-event範例會更新事件發生的時間。

```
aws ssm-incidents update-timeline-event \
    --event-id 20bcc812-8a94-4cd7-520c-0ff742111424 \
```

```
    --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308" \
    --event-time "2021-05-21T18:10:57+00:00"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的事件詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateTimelineEvent。

使用 的 Incident Manager Contacts 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Incident Manager Contacts 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`accept-page`

以下程式碼範例顯示如何使用 `accept-page`。

AWS CLI

在 和 參與期間接受頁面

下列`accept-page`範例使用傳送至聯絡管道的接受代碼來接受頁面。

```
aws ssm-contacts accept-page \
    --page-id "arn:aws:ssm-contacts:us-east-2:682428703967:page/
akuam/94ea0c7b-56d9-46c3-b84a-a37c8b067ad3" \
    --accept-type READ \
    --accept-code 425440
```

此命令不會產生輸出

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptPage。

activate-contact-channel

以下程式碼範例顯示如何使用 activate-contact-channel。

AWS CLI

啟用聯絡人的聯絡管道

下列activate-contact-channel範例會啟用聯絡管道，並將其做為事件的一部分使用。

```
aws ssm-contacts activate-contact-channel \
    --contact-channel-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d" \
    --activation-code "466136"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ActivateContactChannel。

command-name

以下程式碼範例顯示如何使用 command-name。

AWS CLI

刪除聯絡人

下列command-name範例會刪除聯絡人。無法再從任何與其相關的呈報計畫聯絡該聯絡人。

```
aws ssm-contacts delete-contact \
    --contact-id "arn:aws:ssm-contacts:us-east-1:682428703967:contact/alejr"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 命令參考》中的 CommandName。 AWS CLI

create-contact-channel

以下程式碼範例顯示如何使用 create-contact-channel。

AWS CLI

建立聯絡管道

為聯絡人 Akua Mansa 建立 SMS 類型的聯絡管道。您可以建立 SMS、EMAIL 或 VOICE 類型的聯絡管道。

```
aws ssm-contacts create-contact-channel \
    --contact-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \
    --name "akuas sms-test" \
    --type SMS \
    --delivery-address '{"SimpleAddress": "+15005550199"}'
```

輸出：

```
{
    "ContactChannelArn": "arn:aws:ssm-contacts:us-east-1:111122223333:contact-
channel/akuam/02f506b9-ea5d-4764-af89-2daa793ff024"
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateContactChannel。

create-contact

以下程式碼範例顯示如何使用 create-contact。

AWS CLI

建立聯絡人

下列create-contact範例會在您環境中使用空白計劃建立聯絡人。計畫可在建立聯絡管道後更新。使用此 create-contact-channel 命令搭配此命令的輸出 ARN。在您為此聯絡人建立聯絡管道之後，請使用 update-contact 來更新計劃。

```
aws ssm-contacts create-contact \
    --alias "akuam" \
```

```
    --display-name "Akua Mansa" \
    --type PERSONAL \
    --plan '{"Stages": []}'
```

輸出：

```
{
    "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateContact。

deactivate-contact-channel

以下程式碼範例顯示如何使用 deactivate-contact-channel。

AWS CLI

停用聯絡管道

下列deactivate-contact-channel範例會停用聯絡管道。停用聯絡管道表示在事件期間不會再傳呼聯絡管道。您也可以使用 activate-contact-channel命令隨時重新啟用聯絡管道。

```
aws ssm-contacts deactivate-contact-channel \
    --contact-channel-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeactivateContactChannel。

delete-contact-channel

以下程式碼範例顯示如何使用 delete-contact-channel。

AWS CLI

刪除聯絡管道

下列delete-contact-channel範例會刪除聯絡管道。刪除聯絡管道可確保在事件期間不會分頁聯絡管道。

```
aws ssm-contacts delete-contact-channel \
    --contact-channel-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact-channel/akuam/13149bad-52ee-45ea-ae1e-45857f78f9b2"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteContactChannel。

delete-contact

以下程式碼範例顯示如何使用 delete-contact。

AWS CLI

刪除聯絡人

下列delete-contact範例會刪除聯絡人。無法再從任何與其相關的呈報計畫聯絡該聯絡人。

```
aws ssm-contacts delete-contact \
    --contact-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact/alejr"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteContact。

describe-engagement

以下程式碼範例顯示如何使用 describe-engagement。

AWS CLI

描述業務開發的詳細資訊

下列describe-engagement範例列出與聯絡人或呈報計劃互動的詳細資訊。主旨和內容會傳送至聯絡管道。

```
aws ssm-contacts describe-engagement \
    --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
```

輸出：

```
{
    "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation",
    "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356",
    "Sender": "cli",
    "Subject": "cli-test",
    "Content": "Testing engagements via CLI",
    "PublicSubject": "cli-test",
    "PublicContent": "Testing engagements va CLI",
    "StartTime": "2021-05-18T18:25:41.151000+00:00"
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEngagement。

describe-page

以下程式碼範例顯示如何使用 describe-page。

AWS CLI

列出聯絡管道頁面的詳細資訊

下列describe-page範例會列出聯絡管道頁面的詳細資訊。此頁面將包含提供的主題和內容。

```
aws ssm-contacts describe-page \
    --page-id "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/ad0052bd-
e606-498a-861b-25726292eb93"
```

輸出：

```
{
    "PageArn": "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/ad0052bd-
e606-498a-861b-25726292eb93",
```

```
    "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
akuam/78a29753-3674-4ac5-9f83-0468563567f0",
    "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam",
    "Sender": "cli",
    "Subject": "cli-test",
    "Content": "Testing engagements via CLI",
    "PublicSubject": "cli-test",
    "PublicContent": "Testing engagements va CLI",
    "SentTime": "2021-05-18T18:43:29.301000+00:00",
    "ReadTime": "2021-05-18T18:43:55.708000+00:00",
    "DeliveryTime": "2021-05-18T18:43:55.265000+00:00"
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePage。

get-contact-channel

以下程式碼範例顯示如何使用 get-contact-channel。

AWS CLI

列出聯絡管道的詳細資訊

下列get-contact-channel範例列出聯絡管道的詳細資訊。

```
aws ssm-contacts get-contact-channel \
    --contact-channel-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact-
channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d"
```

輸出：

```
{
    "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam",
    "ContactChannelArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact-
channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
    "Name": "akuas sms",
    "Type": "SMS",
    "DeliveryAddress": {
        "SimpleAddress": "+15005550199"
    },
    "ActivationStatus": "ACTIVATED"
```

```
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetContactChannel。

get-contact-policy

以下程式碼範例顯示如何使用 get-contact-policy。

AWS CLI

列出聯絡人的資源政策

下列get-contact-policy範例列出與指定聯絡人相關聯的資源政策。

```
aws ssm-contacts get-contact-policy \
    --contact-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam"
```

輸出：

```
{
    "ContactArn": "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam",
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":
\"SharePolicyForDocumentationDralia\",\"Effect\":\"Allow\",\"Principal\":
{\"AWS\":\"222233334444\"},\"Action\":[\"ssm-contacts:GetContact\",\"ssm-
contacts:StartEngagement\",\"ssm-contacts:DescribeEngagement\",\"ssm-
contacts:ListPagesByEngagement\",\"ssm-contacts:StopEngagement\"],\"Resource
\":[\"arn:aws:ssm-contacts:*:111122223333:contact/akuam\",\"arn:aws:ssm-
contacts:*:111122223333:engagement/akuam/*\"]}]}"
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的使用共用聯絡人和回應計劃。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetContactPolicy。

get-contact

以下程式碼範例顯示如何使用 get-contact。

AWS CLI

範例 1：描述聯絡計畫

下列get-contact範例說明聯絡人。

```
aws ssm-contacts get-contact \
    --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
```

輸出：

```
{
    "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam",
    "Alias": "akuam",
    "DisplayName": "Akua Mansa",
    "Type": "PERSONAL",
    "Plan": {
        "Stages": [
            {
                "DurationInMinutes": 5,
                "Targets": [
                    {
                        "ChannelTargetInfo": {
                            "ContactChannelId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/beb25840-5ac8-4644-95cc-7a8de390fa65",
                            "RetryIntervalInMinutes": 1
                        }
                    }
                ]
            },
            {
                "DurationInMinutes": 5,
                "Targets": [
                    {
                        "ChannelTargetInfo": {
                            "ContactChannelId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/49f3c24d-5f9f-4638-ae25-3f49e04229ad",
                            "RetryIntervalInMinutes": 1
                        }
                    }
                ]
            },
            {
                "DurationInMinutes": 5,
                "Targets": [
                    {
                        "ChannelTargetInfo": {
```

```
                            "ContactChannelId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/77d4f447-f619-4954-afff-85551e369c2a",
                            "RetryIntervalInMinutes": 1
                        }
                    }
                ]
            }
        ]
    }
}
```

範例 2：描述呈報計畫

下列get-contact範例說明呈報計畫。

```
aws ssm-contacts get-contact \
--contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation"
```

輸出：

```
{
    "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation",
    "Alias": "example_escalation",
    "DisplayName": "Example Escalation",
    "Type": "ESCALATION",
    "Plan": {
        "Stages": [
            {
                "DurationInMinutes": 5,
                "Targets": [
                    {
                        "ContactTargetInfo": {
                            "ContactId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact/akuam",
                            "IsEssential": true
                        }
                    }
                ]
            },
            {
                "DurationInMinutes": 5,
```

```
                "Targets": [
                    {
                        "ContactTargetInfo": {
                            "ContactId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact/alejr",
                            "IsEssential": false
                        }
                    }
                ]
            },
            {
                "DurationInMinutes": 0,
                "Targets": [
                    {
                        "ContactTargetInfo": {
                            "ContactId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact/anasi",
                            "IsEssential": false
                        }
                    }
                ]
            }
        ]
    }
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetContact。

`list-contact-channels`

以下程式碼範例顯示如何使用 `list-contact-channels`。

AWS CLI

列出聯絡人的聯絡管道

下列`list-contact-channels`範例列出指定聯絡人的可用聯絡管道。

```
aws ssm-contacts list-contact-channels \
    --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
```

輸出：

```
{
    [
        {
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
            "Name": "akuas email",
            "Type": "EMAIL",
            "DeliveryAddress": {
                "SimpleAddress": "akuam@example.com"
            },
            "ActivationStatus": "NOT_ACTIVATED"
        },
        {
            "ContactChannelArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
            "Name": "akuas sms",
            "Type": "SMS",
            "DeliveryAddress": {
                "SimpleAddress": "+15005550100"
            },
            "ActivationStatus": "ACTIVATED"
        }
    ]
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListContactChannels。

`list-contacts`

以下程式碼範例顯示如何使用 `list-contacts`。

AWS CLI

列出所有呈報計畫和聯絡人

下列`list-contacts`範例列出您帳戶中的聯絡人和呈報計畫。

```
aws ssm-contacts list-contacts
```

輸出：

```
{
    "Contacts": [
        {
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
            "Alias": "akuam",
            "DisplayName": "Akua Mansa",
            "Type": "PERSONAL"
        },
        {
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
alejr",
            "Alias": "alejr",
            "DisplayName": "Alejandro Rosalez",
            "Type": "PERSONAL"
        },
        {
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
anasi",
            "Alias": "anasi",
            "DisplayName": "Ana Carolina Silva",
            "Type": "PERSONAL"
        },
        {
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation",
            "Alias": "example_escalation",
            "DisplayName": "Example Escalation",
            "Type": "ESCALATION"
        }
    ]
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListContacts。

list-engagements

以下程式碼範例顯示如何使用 list-engagements。

AWS CLI

列出所有業務開發

下列list-engagements範例列出呈報計畫和聯絡人的參與。您也可以列出單一事件的參與。

```
aws ssm-contacts list-engagements
```

輸出：

```
{
    "Engagements": [
        {
            "EngagementArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:engagement/akuam/91792571-0b53-4821-9f73-d25d13d9e529",
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
            "Sender": "cli",
            "StartTime": "2021-05-18T20:37:50.300000+00:00"
        },
        {
            "EngagementArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:engagement/akuam/78a29753-3674-4ac5-9f83-0468563567f0",
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
            "Sender": "cli",
            "StartTime": "2021-05-18T18:40:26.666000+00:00"
        },
        {
            "EngagementArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:engagement/
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356",
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation",
            "Sender": "cli",
            "StartTime": "2021-05-18T18:25:41.151000+00:00"
        },
        {
            "EngagementArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:engagement/akuam/607ced0e-e8fa-4ea7-8958-a237b8803f8f",
```

```
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
            "Sender": "cli",
            "StartTime": "2021-05-18T18:20:58.093000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEngagements。

list-page-receipts

以下程式碼範例顯示如何使用 list-page-receipts。

AWS CLI

列出頁面回條

下列command-name範例會列出聯絡人是否收到頁面。

```
aws ssm-contacts list-page-receipts \
    --page-id "arn:aws:ssm-contacts:us-east-2:111122223333:page/
akuam/94ea0c7b-56d9-46c3-b84a-a37c8b067ad3"
```

輸出：

```
{
    "Receipts": [
        {
            "ContactChannelArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
            "ReceiptType": "DELIVERED",
            "ReceiptInfo": "425440",
            "ReceiptTime": "2021-05-18T20:42:57.485000+00:00"
        },
        {
            "ContactChannelArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
            "ReceiptType": "READ",
            "ReceiptInfo": "425440",
```

```
            "ReceiptTime": "2021-05-18T20:42:57.907000+00:00"
        },
        {
            "ContactChannelArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
            "ReceiptType": "SENT",
            "ReceiptInfo": "SM6656c19132f1465f9c9c1123a5dde7c9",
            "ReceiptTime": "2021-05-18T20:40:52.962000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPageReceipts。

`list-pages-by-contact`

以下程式碼範例顯示如何使用 `list-pages-by-contact`。

AWS CLI

依聯絡人列出頁面

下列`list-pages-by-contact`範例會列出指定聯絡人的所有頁面。

```
aws ssm-contacts list-pages-by-contact \
    --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
```

輸出：

```
{
    "Pages": [
        {
            "PageArn": "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/
ad0052bd-e606-498a-861b-25726292eb93",
            "EngagementArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:engagement/akuam/78a29753-3674-4ac5-9f83-0468563567f0",
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
            "Sender": "cli",
            "SentTime": "2021-05-18T18:43:29.301000+00:00",
```

```
            "DeliveryTime": "2021-05-18T18:43:55.265000+00:00",
            "ReadTime": "2021-05-18T18:43:55.708000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPagesByContact。

`list-pages-by-engagement`

以下程式碼範例顯示如何使用 `list-pages-by-engagement`。

AWS CLI

列出從業務開發開始的聯絡管道頁面。

下列`list-pages-by-engagement`範例列出參與已定義的參與計劃時所發生的頁面。

```
aws ssm-contacts list-pages-by-engagement \
    --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
akuam/78a29753-3674-4ac5-9f83-0468563567f0"
```

輸出：

```
{
    "Pages": [
        {
            "PageArn": "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/
ad0052bd-e606-498a-861b-25726292eb93",
            "EngagementArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:engagement/akuam/78a29753-3674-4ac5-9f83-0468563567f0",
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
            "Sender": "cli",
            "SentTime": "2021-05-18T18:40:27.245000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPagesByEngagement。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出聯絡人的標籤

下列`list-tags-for-resource`範例列出指定聯絡人的標籤。

```
aws ssm-contacts list-tags-for-resource \
    --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam"
```

輸出：

```
{
    "Tags": [
        {
            "Key": "group1",
            "Value": "1"
        }
    ]
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的標記。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`put-contact-policy`

以下程式碼範例顯示如何使用 `put-contact-policy`。

AWS CLI

共用聯絡人和業務開發

下列`put-contact-policy`範例會將資源政策新增至與委託人共用聯絡人和相關業務的聯絡人 Akua。

```
aws ssm-contacts put-contact-policy \
```

```
    --contact-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \
    --policy "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":
\"ExampleResourcePolicy\",\"Action\":[\"ssm-contacts:GetContact\",\"ssm-
contacts:StartEngagement\",\"ssm-contacts:DescribeEngagement\",\"ssm-
contacts:ListPagesByEngagement\",\"ssm-contacts:StopEngagement\"],
\"Principal\":{\"AWS\":\"222233334444\"},\"Effect\":\"Allow\",\"Resource
\":[\"arn:aws:ssm-contacts:*:111122223333:contact\/akuam\",\"arn:aws:ssm-
contacts:*:111122223333:engagement\/akuam\/*\"]}]}"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的使用共用聯絡人和回應計劃。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutContactPolicy。

send-activation-code

以下程式碼範例顯示如何使用 send-activation-code。

AWS CLI

傳送啟用碼

下列send-activation-code範例會將啟用碼和訊息傳送至指定的聯絡管道。

```
aws ssm-contacts send-activation-code \
    --contact-channel-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact-
channel/akuam/8ddae2d1-12c8-4e45-b852-c8587266c400"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SendActivationCode。

start-engagement

以下程式碼範例顯示如何使用 start-engagement。

AWS CLI

範例 1：呼叫聯絡人的聯絡管道

以下start-engagement頁面是聯絡人的聯絡管道。寄件者、主體、公有主體和公有內容都不含欄位。Incident Manager 會將主旨和內容傳送至提供的 VOICE 或 EMAIL 聯絡管道。Incident Manager 會將公有主體和公有內容傳送至提供的簡訊聯絡管道。寄件者用於追蹤誰開始了互動。

```
aws ssm-contacts start-engagement \
    --contact-id  "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam" \
    --sender "cli" \
    --subject "cli-test" \
    --content "Testing engagements via CLI" \
    --public-subject "cli-test" \
    --public-content "Testing engagements va CLI"
```

輸出：

```
{
    "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
akuam/607ced0e-e8fa-4ea7-8958-a237b8803f8f"
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

範例 2：在提供的呈報計畫中呼叫聯絡人。

以下透過呈報計畫來start-engagement吸引聯絡人的 。每個聯絡人都會根據他們的互動計畫進行分頁。

```
aws ssm-contacts start-engagement \
    --contact-id  "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation" \
    --sender "cli" \
    --subject "cli-test" \
    --content "Testing engagements via CLI" \
    --public-subject "cli-test" \
    --public-content "Testing engagements va CLI"
```

輸出：

```
{
    "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
}
```

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartEngagement。

stop-engagement

以下程式碼範例顯示如何使用 stop-engagement。

AWS CLI

停止參與

下列stop-engagement範例會停止對進一步聯絡人和聯絡管道進行分頁的參與。

```
aws ssm-contacts stop-engagement \
    --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopEngagement。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

標記聯絡人

下列tag-resource範例使用提供的標籤鍵值對來標記指定的聯絡人。

```
aws ssm-contacts tag-resource \
    --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \
    --tags '[{"Key":"group1","Value":"1"}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的標記。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從聯絡人移除標籤

下列untag-resource範例會從指定的聯絡人移除 group1 標籤。

```
aws ssm-contacts untag-resource \
    --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \
    --tag-keys "group1"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的標記。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-contact-channel

以下程式碼範例顯示如何使用 update-contact-channel。

AWS CLI

更新聯絡管道

下列update-contact-channel範例會更新聯絡管道的名稱和交付地址。

```
aws ssm-contacts update-contact-channel \
    --contact-channel-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact-
channel/akuam/49f3c24d-5f9f-4638-ae25-3f49e04229ad" \
    --name "akuas voice channel" \
    --delivery-address '{"SimpleAddress": "+15005550198"}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateContactChannel。

update-contact

以下程式碼範例顯示如何使用 update-contact。

AWS CLI

更新聯絡的參與計劃

下列update-contact範例會更新聯絡 Akua 的參與計劃，以包含三種類型的聯絡管道。這會在建立 Akua 的聯絡管道之後完成。

```
aws ssm-contacts update-contact \
    --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam" \
    --plan '{"Stages": [{"DurationInMinutes": 5, "Targets": [{"ChannelTargetInfo":
 {"ContactChannelId": "arn:aws:ssm-contacts:us-east-2:111122223333:contact-
channel/akuam/beb25840-5ac8-4644-95cc-7a8de390fa65","RetryIntervalInMinutes":
 1 }}]}, {"DurationInMinutes": 5, "Targets": [{"ChannelTargetInfo":
{"ContactChannelId": "arn:aws:ssm-contacts:us-east-2:111122223333:contact-channel/
akuam/49f3c24d-5f9f-4638-ae25-3f49e04229ad", "RetryIntervalInMinutes": 1}}]},
 {"DurationInMinutes": 5, "Targets": [{"ChannelTargetInfo": {"ContactChannelId":
 "arn:aws:ssm-contacts:us-east-2:111122223333:contact-channel/akuam/77d4f447-
f619-4954-afff-85551e369c2a","RetryIntervalInMinutes": 1 }}]}]}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 Incident Manager 使用者指南》中的聯絡人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateContact。

使用 的 Amazon Inspector 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon Inspector 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

add-attributes-to-findings

以下程式碼範例顯示如何使用 add-attributes-to-findings。

AWS CLI

將屬性新增至問題清單

下列add-attribute-to-finding命令會將索引鍵為 Example且值example為 的屬性指派給 ARN 為 的調查結果arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-8l1VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU：

```
aws inspector add-attributes-to-findings --finding-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-8l1VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU --attributes key=Example,value=example
```

輸出：

```
{
    "failedItems": {}
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 調查結果。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddAttributesToFindings。

associate-member

以下程式碼範例顯示如何使用 associate-member。

AWS CLI

範例：將 AWS 帳戶與 Amazon Inspector 委派管理員建立關聯

下列associate-member範例會將 AWS 帳戶與 Amazon Inspector 委派管理員建立關聯。

```
aws inspector2 associate-member \
    --account-id 123456789012
```

輸出：

```
{
    "accountId": "123456789012"
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的使用 AWS Organizations 在 Amazon Inspector 中管理多個帳戶。 Amazon Inspector

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateMember。

create-assessment-target

以下程式碼範例顯示如何使用 create-assessment-target。

AWS CLI

建立評估目標

下列create-assessment-target命令ExampleAssessmentTarget會使用具有 之 ARN 的資源群組來建立名為 的評估目標arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-AB6DMKnv：

```
aws inspector create-assessment-target --assessment-target-
name ExampleAssessmentTarget --resource-group-arn arn:aws:inspector:us-
west-2:123456789012:resourcegroup/0-AB6DMKnv
```

輸出：

```
{
    "assessmentTargetArn": "arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX"
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估目標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAssessmentTarget。

create-assessment-template

以下程式碼範例顯示如何使用 create-assessment-template。

AWS CLI

建立評估範本

下列create-assessment-template命令會使用 的 ARN 建立名為 ExampleAssessmentTemplate 的評估目標評估範本arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX：

```
aws inspector create-assessment-template --assessment-target-
arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX --assessment-
template-name ExampleAssessmentTemplate --duration-in-seconds 180 --rules-package-
arns arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p --user-
attributes-for-findings key=ExampleTag,value=examplevalue
```

輸出：

```
{
    "assessmentTemplateArn": "arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX/template/0-it5r2S4T"
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估範本和評估執行。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAssessmentTemplate。

create-filter

以下程式碼範例顯示如何使用 create-filter。

AWS CLI

建立篩選條件

下列create-filter範例會建立隱藏規則，省略 ECR 執行個體類型問題清單。

```
aws inspector2 create-filter \
    --name "ExampleSuppressionRuleECR" \
    --description "This suppression rule omits ECR instance type findings" \
```

```
    --action SUPPRESS \
    --filter-criteria 'resourceType=[{comparison="EQUALS",
 value="AWS_ECR_INSTANCE"}]'
```

輸出：

```
{
    "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/
EXAMPLE444444444"
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的篩選 Amazon Inspector 問題清單。 Amazon Inspector

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFilter。

create-findings-report

以下程式碼範例顯示如何使用 create-findings-report。

AWS CLI

建立問題清單報告

下列create-findings-report範例會建立調查結果報告。

```
aws inspector2 create-findings-report \
    --report-format CSV \
    --s3-destination bucketName=inspector-sbom-123456789012,keyPrefix=sbom-
key,kmsKeyArn=arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE33333 \
    --filter-criteria '{"ecrImageRepositoryName":
[{"comparison":"EQUALS","value":"debian"}]}'
```

輸出：

```
{
    "reportId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的管理 Amazon Inspector 中的問題清單。 Amazon Inspector

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFindingsReport。

create-resource-group

以下程式碼範例顯示如何使用 create-resource-group。

AWS CLI

建立資源群組

下列create-resource-group命令會使用 的標籤索引鍵Name和 的值來建立資源群組example：

```
aws inspector create-resource-group --resource-group-tags key=Name,value=example
```

輸出：

```
{
   "resourceGroupArn": "arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-
AB6DMKnv"
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估目標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateResourceGroup。

create-sbom-export

以下程式碼範例顯示如何使用 create-sbom-export。

AWS CLI

建立軟體物料清單 (SBOM) 報告

下列create-sbom-export範例會建立軟體物料清單 (SBOM) 報告。

```
aws inspector2 create-sbom-export \
    --report-format SPDX_2_3 \
    --resource-filter-criteria
 'ecrRepositoryName=[{comparison="EQUALS",value="debian"}]' \
```

```
    --s3-destination bucketName=inspector-sbom-123456789012,keyPrefix=sbom-
key,kmsKeyArn=arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE33333
```

輸出：

```
{
    "reportId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的使用 Amazon Inspector 匯出 SBOMs。 Amazon Inspector

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSbomExport。

delete-assessment-run

以下程式碼範例顯示如何使用 delete-assessment-run。

AWS CLI

刪除評估執行

下列delete-assessment-run命令會使用 的 ARN 刪除評估執行arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-11LMTAVe：

```
aws inspector delete-assessment-run --assessment-run-arn arn:aws:inspector:us-
west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-11LMTAVe
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估範本和評估執行。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAssessmentRun。

delete-assessment-target

以下程式碼範例顯示如何使用 delete-assessment-target。

AWS CLI

刪除評估目標

下列delete-assessment-target命令會使用 的 ARN 刪除評估目標arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq：

```
aws inspector delete-assessment-target --assessment-target-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估目標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAssessmentTarget。

delete-assessment-template

以下程式碼範例顯示如何使用 delete-assessment-template。

AWS CLI

刪除評估範本

下列delete-assessment-template命令會使用 的 ARN 刪除評估範本arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T：

```
aws inspector delete-assessment-template --assessment-template-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估範本和評估執行。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAssessmentTemplate。

delete-filter

以下程式碼範例顯示如何使用 delete-filter。

AWS CLI

刪除篩選條件

下列delete-filter範例會刪除篩選條件。

```
aws inspector2 delete-filter \
    --arn "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/EXAMPLE444444444"
```

輸出：

```
{
    "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/
EXAMPLE444444444"
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的篩選 Amazon Inspector 問題清單。Amazon Inspector

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFilter。

describe-assessment-runs

以下程式碼範例顯示如何使用 describe-assessment-runs。

AWS CLI

描述評估執行

下列describe-assessment-run命令說明使用 的 ARN 執行的評估arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE：

```
aws inspector describe-assessment-runs --assessment-run-arns arn:aws:inspector:us-
west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE
```

輸出：

```
{
        "assessmentRuns": [
          {
                "arn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-MKkpXXPE",
                "assessmentTemplateArn": "arn:aws:inspector:us-
west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw",
                "completedAt": 1458680301.4,
                "createdAt": 1458680170.035,
                "dataCollected": true,
                "durationInSeconds": 3600,
                "name": "Run 1 for ExampleAssessmentTemplate",
                "notifications": [],
```

```
            "rulesPackageArns": [
              "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-X1KXtawP"
            ],
            "startedAt": 1458680170.161,
            "state": "COMPLETED",
            "stateChangedAt": 1458680301.4,
            "stateChanges": [
              {
                    "state": "CREATED",
                    "stateChangedAt": 1458680170.035
              },
              {
                    "state": "START_DATA_COLLECTION_PENDING",
                    "stateChangedAt": 1458680170.065
              },
              {
                    "state": "START_DATA_COLLECTION_IN_PROGRESS",
                    "stateChangedAt": 1458680170.096
              },
              {
                    "state": "COLLECTING_DATA",
                    "stateChangedAt": 1458680170.161
              },
              {
                    "state": "STOP_DATA_COLLECTION_PENDING",
                    "stateChangedAt": 1458680239.883
              },
              {
                    "state": "DATA_COLLECTED",
                    "stateChangedAt": 1458680299.847
              },
              {
                    "state": "EVALUATING_RULES",
                    "stateChangedAt": 1458680300.099
              },
              {
                    "state": "COMPLETED",
                    "stateChangedAt": 1458680301.4
              }
            ],
            "userAttributesForFindings": []
      }
    ],
    "failedItems": {}
```

```
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估範本和評估執行。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAssessmentRuns。

`describe-assessment-targets`

以下程式碼範例顯示如何使用 `describe-assessment-targets`。

AWS CLI

描述評估目標

下列`describe-assessment-targets`命令使用 的 ARN 描述評估目標`arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq`：

```
aws inspector describe-assessment-targets --assessment-target-
arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq
```

輸出：

```
{
      "assessmentTargets": [
        {
              "arn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq",
              "createdAt": 1458074191.459,
              "name": "ExampleAssessmentTarget",
              "resourceGroupArn": "arn:aws:inspector:us-
west-2:123456789012:resourcegroup/0-PyGXopAI",
              "updatedAt": 1458074191.459
        }
      ],
      "failedItems": {}
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估目標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAssessmentTargets。

describe-assessment-templates

以下程式碼範例顯示如何使用 describe-assessment-templates。

AWS CLI

描述評估範本

下列describe-assessment-templates命令描述具有 之 ARN 的評估範本arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw：

```
aws inspector describe-assessment-templates --assessment-template-
arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw
```

輸出：

```
{
      "assessmentTemplates": [
        {
              "arn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw",
              "assessmentTargetArn": "arn:aws:inspector:us-
west-2:123456789012:target/0-0kFIPusq",
              "createdAt": 1458074191.844,
              "durationInSeconds": 3600,
              "name": "ExampleAssessmentTemplate",
              "rulesPackageArns": [
                "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-X1KXtawP"
              ],
              "userAttributesForFindings": []
        }
      ],
      "failedItems": {}
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估範本和評估執行。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAssessmentTemplates。

describe-cross-account-access-role

以下程式碼範例顯示如何使用 describe-cross-account-access-role。

AWS CLI

描述跨帳戶存取角色

下列describe-cross-account-access-role命令說明可讓 Amazon Inspector 存取您 AWS 帳戶的 IAM 角色：

```
aws inspector describe-cross-account-access-role
```

輸出：

```
{
        "registeredAt": 1458069182.826,
        "roleArn": "arn:aws:iam::123456789012:role/inspector",
        "valid": true
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的設定 Amazon Inspector。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCrossAccountAccessRole。

describe-findings

以下程式碼範例顯示如何使用 describe-findings。

AWS CLI

描述問題清單

下列describe-findings命令使用 的 ARN 描述問題清單arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE/finding/0-HwPnsDm4：

```
aws inspector describe-findings --finding-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE/finding/0-HwPnsDm4
```

輸出：

```
{
      "failedItems": {},
      "findings": [
```

```
        {
              "arn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-MKkpXXPE/finding/0-HwPnsDm4",
              "assetAttributes": {
                "ipv4Addresses": [],
                "schemaVersion": 1
              },
              "assetType": "ec2-instance",
              "attributes": [],
              "confidence": 10,
              "createdAt": 1458680301.37,
              "description": "Amazon Inspector did not find any potential security
 issues during this assessment.",
              "indicatorOfCompromise": false,
              "numericSeverity": 0,
              "recommendation": "No remediation needed.",
              "schemaVersion": 1,
              "service": "Inspector",
              "serviceAttributes": {
                "assessmentRunArn": "arn:aws:inspector:us-
west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE",
                "rulesPackageArn": "arn:aws:inspector:us-
west-2:758058086616:rulespackage/0-X1KXtawP",
                "schemaVersion": 1
              },
              "severity": "Informational",
              "title": "No potential security issues found",
              "updatedAt": 1458680301.37,
              "userAttributes": []
        }
      ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 調查結果。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFindings。

describe-resource-groups

以下程式碼範例顯示如何使用 describe-resource-groups。

AWS CLI

描述資源群組

下列describe-resource-groups命令說明具有 之 ARN 的資源群組arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-PyGXopAI：

```
aws inspector describe-resource-groups --resource-group-arns arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-PyGXopAI
```

輸出：

```
{
      "failedItems": {},
      "resourceGroups": [
        {
              "arn": "arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-PyGXopAI",
              "createdAt": 1458074191.098,
              "tags": [
                {
                      "key": "Name",
                      "value": "example"
                }
              ]
        }
      ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估目標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeResourceGroups。

describe-rules-packages

以下程式碼範例顯示如何使用 describe-rules-packages。

AWS CLI

描述規則套件

下列describe-rules-packages命令說明具有 之 ARN 的規則套件arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p：

```
aws inspector describe-rules-packages --rules-package-arns arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p
```

輸出：

```
{
      "failedItems": {},
      "rulesPackages": [
        {
              "arn": "arn:aws:inspector:us-
west-2:758058086616:rulespackage/0-9hgA516p",
              "description": "The rules in this package help verify whether the EC2
 instances in your application are exposed to Common Vulnerabilities and
              Exposures (CVEs). Attacks can exploit unpatched vulnerabilities to
 compromise the confidentiality, integrity, or availability of your service
              or data. The CVE system provides a reference for publicly known
 information security vulnerabilities and exposures. For more information, see
              [https://cve.mitre.org/](https://cve.mitre.org/). If a particular CVE
 appears in one of the produced Findings at the end of a completed
              Inspector assessment, you can search [https://cve.mitre.org/](https://
cve.mitre.org/) using the CVE's ID (for example, \"CVE-2009-0021\") to
              find detailed information about this CVE, its severity, and how to
 mitigate it. ",
              "name": "Common Vulnerabilities and Exposures",
              "provider": "Amazon Web Services, Inc.",
              "version": "1.1"
        }
      ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 規則套件和規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRulesPackages。

disassociate-member

以下程式碼範例顯示如何使用 disassociate-member。

AWS CLI

範例：取消成員帳戶與 Amazon Inspector 委派管理員的關聯

下列disassociate-member範例會取消 AWS 帳戶與 Amazon Inspector 委派管理員的關聯。

```
aws inspector2 disassociate-member \
    --account-id 123456789012
```

輸出：

```
{
    "accountId": "123456789012"
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的使用 AWS Organizations 在 Amazon Inspector 中管理多個帳戶。 Amazon Inspector

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateMember。

get-configuration

以下程式碼範例顯示如何使用 get-configuration。

AWS CLI

取得 Inspector 掃描的設定組態

下列get-configuration範例取得 Inspector 掃描的設定組態。

```
aws inspector2 get-configuration
```

輸出：

```
{
    "ec2Configuration": {
        "scanModeState": {
            "scanMode": "EC2_HYBRID",
            "scanModeStatus": "SUCCESS"
        }
    },
    "ecrConfiguration": {
        "rescanDurationState": {
            "pullDateRescanDuration": "DAYS_90",
            "rescanDuration": "DAYS_30",
            "status": "SUCCESS",
            "updatedAt": "2024-05-14T21:16:20.237000+00:00"
        }
     }
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的使用 Amazon Inspector 自動化資源掃描。 Amazon Inspector

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetConfiguration。

`get-member`

以下程式碼範例顯示如何使用 `get-member`。

AWS CLI

範例：取得組織的成員資訊

aws inspector2 get-member --account-id 123456789012

輸出：

```
{
        "member": {
        "accountId": "123456789012",
        "delegatedAdminAccountId": "123456789012",
        "relationshipStatus": "ENABLED",
        "updatedAt": "2023-09-11T09:57:20.520000-07:00"
    }
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的使用 AWS Organizations 在 Amazon Inspector 中管理多個帳戶。 Amazon Inspector

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMember。

`get-telemetry-metadata`

以下程式碼範例顯示如何使用 `get-telemetry-metadata`。

AWS CLI

取得遙測中繼資料

下列`get-telemetry-metadata`命令會產生使用 的 ARN 為評估執行所收集的資料的相關資訊`arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE`：

```
aws inspector get-telemetry-metadata --assessment-run-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE
```

輸出：

```
{
      "telemetryMetadata": [
        {
              "count": 2,
              "dataSize": 345,
              "messageType": "InspectorDuplicateProcess"
        },
        {
              "count": 3,
              "dataSize": 255,
              "messageType": "InspectorTimeEventMsg"
        },
        {
              "count": 4,
              "dataSize": 1082,
              "messageType": "InspectorNetworkInterface"
        },
        {
              "count": 2,
              "dataSize": 349,
              "messageType": "InspectorDnsEntry"
        },
        {
              "count": 11,
              "dataSize": 2514,
              "messageType": "InspectorDirectoryInfoMsg"
        },
        {
              "count": 1,
              "dataSize": 179,
              "messageType": "InspectorTcpV6ListeningPort"
        },
        {
              "count": 101,
              "dataSize": 10949,
              "messageType": "InspectorTerminal"
        },
        {
```

```
            "count": 26,
            "dataSize": 5916,
            "messageType": "InspectorUser"
        },
        {
            "count": 282,
            "dataSize": 32148,
            "messageType": "InspectorDynamicallyLoadedCodeModule"
        },
        {
            "count": 18,
            "dataSize": 10172,
            "messageType": "InspectorCreateProcess"
        },
        {
            "count": 3,
            "dataSize": 8001,
            "messageType": "InspectorProcessPerformance"
        },
        {
            "count": 1,
            "dataSize": 360,
            "messageType": "InspectorOperatingSystem"
        },
        {
            "count": 6,
            "dataSize": 546,
            "messageType": "InspectorStopProcess"
        },
        {
            "count": 1,
            "dataSize": 1553,
            "messageType": "InspectorInstanceMetaData"
        },
        {
            "count": 2,
            "dataSize": 434,
            "messageType": "InspectorTcpV4Connection"
        },
        {
            "count": 474,
            "dataSize": 2960322,
            "messageType": "InspectorPackageInfo"
        },
```

```
        {
              "count": 3,
              "dataSize": 2235,
              "messageType": "InspectorSystemPerformance"
        },
        {
              "count": 105,
              "dataSize": 46048,
              "messageType": "InspectorCodeModule"
        },
        {
              "count": 1,
              "dataSize": 182,
              "messageType": "InspectorUdpV6ListeningPort"
        },
        {
              "count": 2,
              "dataSize": 371,
              "messageType": "InspectorUdpV4ListeningPort"
        },
        {
              "count": 18,
              "dataSize": 8362,
              "messageType": "InspectorKernelModule"
        },
        {
              "count": 29,
              "dataSize": 48788,
              "messageType": "InspectorConfigurationInfo"
        },
        {
              "count": 1,
              "dataSize": 79,
              "messageType": "InspectorMonitoringStart"
        },
        {
              "count": 5,
              "dataSize": 0,
              "messageType": "InspectorSplitMsgBegin"
        },
        {
              "count": 51,
              "dataSize": 4593,
              "messageType": "InspectorGroup"
```

```
        },
        {
              "count": 1,
              "dataSize": 184,
              "messageType": "InspectorTcpV4ListeningPort"
        },
        {
              "count": 1159,
              "dataSize": 3146579,
              "messageType": "Total"
        },
        {
              "count": 5,
              "dataSize": 0,
              "messageType": "InspectorSplitMsgEnd"
        },
        {
              "count": 1,
              "dataSize": 612,
              "messageType": "InspectorLoadImageInProcess"
        }
      ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTelemetryMetadata。

`list-account-permissions`

以下程式碼範例顯示如何使用 `list-account-permissions`。

AWS CLI

列出帳戶許可

下列`list-account-permissions`範例列出您的帳戶許可。

```
aws inspector2 list-account-permissions
```

輸出：

```
{
    "permissions": [
```

```
        {
            "operation": "ENABLE_SCANNING",
            "service": "ECR"
        },
        {
            "operation": "DISABLE_SCANNING",
            "service": "ECR"
        },
        {
            "operation": "ENABLE_REPOSITORY",
            "service": "ECR"
        },
        {
            "operation": "DISABLE_REPOSITORY",
            "service": "ECR"
        },
        {
            "operation": "ENABLE_SCANNING",
            "service": "EC2"
        },
        {
            "operation": "DISABLE_SCANNING",
            "service": "EC2"
        },
        {
            "operation": "ENABLE_SCANNING",
            "service": "LAMBDA"
        },
        {
            "operation": "DISABLE_SCANNING",
            "service": "LAMBDA"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的 Amazon Inspector 的 Identity and Access Management。 Amazon Inspector

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAccountPermissions。

`list-assessment-run-agents`

以下程式碼範例顯示如何使用 `list-assessment-run-agents`。

AWS CLI

列出評估執行代理程式

下列list-assessment-run-agents命令列出使用指定 ARN 執行評估的代理程式。

```
aws inspector list-assessment-run-agents \
    --assessment-run-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-MKkpXXPE
```

輸出：

```
{
    "assessmentRunAgents": [
        {
            "agentHealth": "HEALTHY",
            "agentHealthCode": "HEALTHY",
            "agentId": "i-49113b93",
            "assessmentRunArn": "arn:aws:inspector:us-
west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE",
            "telemetryMetadata": [
                {
                    "count": 2,
                    "dataSize": 345,
                    "messageType": "InspectorDuplicateProcess"
                },
                {
                    "count": 3,
                    "dataSize": 255,
                    "messageType": "InspectorTimeEventMsg"
                },
                {
                    "count": 4,
                    "dataSize": 1082,
                    "messageType": "InspectorNetworkInterface"
                },
                {
                    "count": 2,
                    "dataSize": 349,
                    "messageType": "InspectorDnsEntry"
                },
                {
                    "count": 11,
```

```
                    "dataSize": 2514,
                    "messageType": "InspectorDirectoryInfoMsg"
                },
                {
                    "count": 1,
                    "dataSize": 179,
                    "messageType": "InspectorTcpV6ListeningPort"
                },
                {
                    "count": 101,
                    "dataSize": 10949,
                    "messageType": "InspectorTerminal"
                },
                {
                    "count": 26,
                    "dataSize": 5916,
                    "messageType": "InspectorUser"
                },
                {
                    "count": 282,
                    "dataSize": 32148,
                    "messageType": "InspectorDynamicallyLoadedCodeModule"
                },
                {
                    "count": 18,
                    "dataSize": 10172,
                    "messageType": "InspectorCreateProcess"
                },
                {
                    "count": 3,
                    "dataSize": 8001,
                    "messageType": "InspectorProcessPerformance"
                },
                {
                    "count": 1,
                    "dataSize": 360,
                    "messageType": "InspectorOperatingSystem"
                },
                {
                    "count": 6,
                    "dataSize": 546,
                    "messageType": "InspectorStopProcess"
                },
                {
```

```
                    "count": 1,
                    "dataSize": 1553,
                    "messageType": "InspectorInstanceMetaData"
                },
                {
                    "count": 2,
                    "dataSize": 434,
                    "messageType": "InspectorTcpV4Connection"
                },
                {
                    "count": 474,
                    "dataSize": 2960322,
                    "messageType": "InspectorPackageInfo"
                },
                {
                    "count": 3,
                    "dataSize": 2235,
                    "messageType": "InspectorSystemPerformance"
                },
                {
                    "count": 105,
                    "dataSize": 46048,
                    "messageType": "InspectorCodeModule"
                },
                {
                    "count": 1,
                    "dataSize": 182,
                    "messageType": "InspectorUdpV6ListeningPort"
                },
                {
                    "count": 2,
                    "dataSize": 371,
                    "messageType": "InspectorUdpV4ListeningPort"
                },
                {
                    "count": 18,
                    "dataSize": 8362,
                    "messageType": "InspectorKernelModule"
                },
                {
                    "count": 29,
                    "dataSize": 48788,
                    "messageType": "InspectorConfigurationInfo"
                },
```

```
                {
                    "count": 1,
                    "dataSize": 79,
                    "messageType": "InspectorMonitoringStart"
                },
                {
                    "count": 5,
                    "dataSize": 0,
                    "messageType": "InspectorSplitMsgBegin"
                },
                {
                    "count": 51,
                    "dataSize": 4593,
                    "messageType": "InspectorGroup"
                },
                {
                    "count": 1,
                    "dataSize": 184,
                    "messageType": "InspectorTcpV4ListeningPort"
                },
                {
                    "count": 1159,
                    "dataSize": 3146579,
                    "messageType": "Total"
                },
                {
                    "count": 5,
                    "dataSize": 0,
                    "messageType": "InspectorSplitMsgEnd"
                },
                {
                    "count": 1,
                    "dataSize": 612,
                    "messageType": "InspectorLoadImageInProcess"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的AWS 客服人員。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAssessmentRunAgents。

list-assessment-runs

以下程式碼範例顯示如何使用 list-assessment-runs。

AWS CLI

列出評估執行

下列list-assessment-runs命令會列出所有現有的評估執行。

```
aws inspector list-assessment-runs
```

輸出：

```
{
    "assessmentRunArns": [
        "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-MKkpXXPE",
        "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-v5D6fI3v"
    ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的 Amazon Inspector 評估範本和評估執行。 Amazon Inspector

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAssessmentRuns。

list-assessment-targets

以下程式碼範例顯示如何使用 list-assessment-targets。

AWS CLI

列出評估目標

下列list-assessment-targets命令會列出所有現有的評估目標：

```
aws inspector list-assessment-targets
```

輸出：

```
{
```

```
        "assessmentTargetArns": [
        "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq"
        ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估目標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAssessmentTargets。

`list-assessment-templates`

以下程式碼範例顯示如何使用 `list-assessment-templates`。

AWS CLI

列出評估範本

下列`list-assessment-templates`命令會列出所有現有的評估範本：

```
aws inspector list-assessment-templates
```

輸出：

```
{
        "assessmentTemplateArns": [
        "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw",
        "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-
Uza6ihLh"
        ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估範本和評估執行。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAssessmentTemplates。

`list-coverage-statistics`

以下程式碼範例顯示如何使用 `list-coverage-statistics`。

AWS CLI

範例 1：依群組列出涵蓋範圍統計資料

下列list-coverage-statistics範例會依群組列出您 AWS 環境的涵蓋範圍統計資料。

```
aws inspector2 list-coverage-statistics \
   --group-by RESOURCE_TYPE
```

輸出：

```
{
    "countsByGroup": [
        {
            "count": 56,
            "groupKey": "AWS_LAMBDA_FUNCTION"
        },
        {
            "count": 27,
            "groupKey": "AWS_ECR_REPOSITORY"
        },
        {
            "count": 18,
            "groupKey": "AWS_EC2_INSTANCE"
        },
        {
            "count": 3,
            "groupKey": "AWS_ECR_CONTAINER_IMAGE"
        },
        {
            "count": 1,
            "groupKey": "AWS_ACCOUNT"
        }
    ],
    "totalCounts": 105
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的評估 AWS 環境的 Amazon Inspector 涵蓋範圍。 Amazon Inspector

範例 2：依資源類型列出涵蓋範圍統計資料

下列list-coverage-statistics範例會依資源類型列出您 AWS 環境的涵蓋範圍統計資料。

```
aws inspector2 list-coverage-statistics
    --filter-criteria '{"resourceType":
[{"comparison":"EQUALS","value":"AWS_ECR_REPOSITORY"}]}'
```

```
    --group-by SCAN_STATUS_REASON
```

輸出：

```
{
    "countsByGroup": [
        {
            "count": 27,
            "groupKey": "SUCCESSFUL"
        }
    ],
    "totalCounts": 27
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的評估 AWS 環境的 Amazon Inspector 涵蓋範圍。 Amazon Inspector

範例 3：依 ECR 儲存庫名稱列出涵蓋範圍統計資料

下列list-coverage-statistics範例會依 ECR 儲存庫名稱列出您 AWS 環境的涵蓋範圍統計資料。

```
aws inspector2 list-coverage-statistics
   --filter-criteria '{"ecrRepositoryName":
[{"comparison":"EQUALS","value":"debian"}]}'
   --group-by SCAN_STATUS_REASON
```

輸出：

```
{
    "countsByGroup": [
        {
            "count": 3,
            "groupKey": "SUCCESSFUL"
        }
    ],
    "totalCounts": 3
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的評估 AWS 環境的 Amazon Inspector 涵蓋範圍。 Amazon Inspector

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCoverageStatistics。

`list-coverage`

以下程式碼範例顯示如何使用 `list-coverage`。

AWS CLI

範例 1：列出您環境的涵蓋範圍詳細資訊

下列`list-coverage`範例列出您環境的涵蓋範圍詳細資訊。

```
aws inspector2 list-coverage
```

輸出：

```
{
    "coveredResources": [
        {
            "accountId": "123456789012",
            "lastScannedAt": "2024-05-20T16:23:20-07:00",
            "resourceId": "i-EXAMPLE55555555555",
            "resourceMetadata": {
                "ec2": {
                    "amiId": "ami-EXAMPLE6666666666",
                    "platform": "LINUX"
                }
            },
            "resourceType": "AWS_EC2_INSTANCE",
            "scanStatus": {
                "reason": "SUCCESSFUL",
                "statusCode": "ACTIVE"
            },
            "scanType": "PACKAGE"
        }
    ]
}
```

範例 2：列出 Lambda 函數資源類型的涵蓋範圍詳細資訊

下列`list-coverage`範例列出您的 Lamda 函數資源類型詳細資訊。

```
aws inspector2 list-coverage
    --filter-criteria '{"resourceType":
[{"comparison":"EQUALS","value":"AWS_LAMBDA_FUNCTION"}]}'
```

輸出：

```
{
    "coveredResources": [
        {
            "accountId": "123456789012",
            "resourceId": "arn:aws:lambda:us-west-2:123456789012:function:Eval-
container-scan-results:$LATEST",
            "resourceMetadata": {
                "lambdaFunction": {
                    "functionName": "Eval-container-scan-results",
                    "functionTags": {},
                    "layers": [],
                    "runtime": "PYTHON_3_7"
                }
            },
            "resourceType": "AWS_LAMBDA_FUNCTION",
            "scanStatus": {
                "reason": "SUCCESSFUL",
                "statusCode": "ACTIVE"
            },
            "scanType": "CODE"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCoverage。

list-delegated-admin-accounts

以下程式碼範例顯示如何使用 list-delegated-admin-accounts。

AWS CLI

列出組織委派管理員帳戶的相關資訊

下列list-delegated-admin-accounts範例列出組織委派管理員帳戶的相關資訊。

```
aws inspector2 list-delegated-admin-accounts
```

輸出：

```
{
    "delegatedAdminAccounts": [
        {
            "accountId": "123456789012",
            "status": "ENABLED"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的為 Amazon Inspector 指定委派管理員。 Amazon Inspector

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDelegatedAdminAccounts。

`list-event-subscriptions`

以下程式碼範例顯示如何使用 `list-event-subscriptions`。

AWS CLI

列出事件訂閱

下列`list-event-subscriptions`命令會列出評估範本的所有事件訂閱，其 ARN 為 `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0`：

```
aws inspector list-event-subscriptions --resource-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0
```

輸出：

```
{
      "subscriptions": [
        {
              "eventSubscriptions": [
                {
                      "event": "ASSESSMENT_RUN_COMPLETED",
```

```
                    "subscribedAt": 1459455440.867
                }
            ],
            "resourceArn": "arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX/template/0-7sbz2Kz0",
            "topicArn": "arn:aws:sns:us-west-2:123456789012:exampletopic"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估範本和評估執行。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEventSubscriptions。

list-filters

以下程式碼範例顯示如何使用 `list-filters`。

AWS CLI

列出與您用來啟用 Amazon Inspector 之帳戶相關聯的篩選條件

下列`list-filters`範例列出與您用來啟用 Amazon Inspector 之帳戶相關聯的篩選條件。

```
aws inspector2 list-filters
```

輸出：

```
{
    "filters": [
        {
            "action": "SUPPRESS",
            "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/
filter/EXAMPLE444444444",
            "createdAt": "2024-05-15T21:11:08.602000+00:00",
            "criteria": {
                "resourceType": [
                    {
                        "comparison": "EQUALS",
                        "value": "AWS_EC2_INSTANCE"
                    },
                ]
```

```
            },
            "description": "This suppression rule omits EC2 instance type findings",
            "name": "ExampleSuppressionRuleEC2",
            "ownerId": "o-EXAMPLE222",
            "tags": {},
            "updatedAt": "2024-05-15T21:11:08.602000+00:00"
        },
        {
            "action": "SUPPRESS",
            "arn": "arn:aws:inspector2:us-east-1:813737243517:owner/o-EXAMPLE222/
filter/EXAMPLE444444444",
            "createdAt": "2024-05-15T21:28:27.054000+00:00",
            "criteria": {
                "resourceType": [
                    {
                        "comparison": "EQUALS",
                        "value": "AWS_ECR_INSTANCE"
                    }
                ]
            },
            "description": "This suppression rule omits ECR instance type findings",
            "name": "ExampleSuppressionRuleECR",
            "ownerId": "o-EXAMPLE222",
            "tags": {},
            "updatedAt": "2024-05-15T21:28:27.054000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的篩選 Amazon Inspector 問題清單。Amazon Inspector

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFilters。

`list-findings`

以下程式碼範例顯示如何使用 `list-findings`。

AWS CLI

列出問題清單

下列`list-findings`命令會列出所有產生的調查結果：

```
aws inspector list-findings
```

輸出：

```
{
        "findingArns": [
        "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-MKkpXXPE/finding/0-HwPnsDm4",
        "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-v5D6fI3v/finding/0-tyvmqBLy"
      ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 調查結果。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFindings。

`list-members`

以下程式碼範例顯示如何使用 `list-members`。

AWS CLI

範例 1：列出與組織 Amazon Inspector 委派管理員相關聯的所有成員帳戶

aws inspector2 list-members --only-associated

輸出：

```
{
        {
             "members": [
        {
             "accountId": "123456789012",
             "delegatedAdminAccountId": "123456789012",
             "relationshipStatus": "ENABLED",
             "updatedAt": "2023-09-11T09:57:20.520000-07:00"
        },
        {
             "accountId": "123456789012",
             "delegatedAdminAccountId": "123456789012",
             "relationshipStatus": "ENABLED",
```

```
            "updatedAt": "2024-08-12T10:13:01.472000-07:00"
        },
        {
            "accountId": "625032911453",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "ENABLED",
            "updatedAt": "2023-09-11T09:57:20.438000-07:00"
        },
        {
            "accountId": "715411239211",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "ENABLED",
            "updatedAt": "2024-04-24T09:14:57.471000-07:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的使用 AWS Organizations 在 Amazon Inspector 中管理多個帳戶。 Amazon Inspector

範例 2：列出與您組織 Amazon Inspector 委派管理員關聯和取消關聯的所有成員帳戶

aws inspector2 list-members --no-only-associated

輸出：

```
{
        {
            "members": [
        {
            "accountId": "123456789012",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "REMOVED",
            "updatedAt": "2024-05-15T11:34:53.326000-07:00"
        },
        {
            "accountId": "123456789012",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "ENABLED",
            "updatedAt": "2023-09-11T09:57:20.520000-07:00"
        },
        {
            "accountId": "123456789012",
```

```
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "ENABLED",
            "updatedAt": "2024-08-12T10:13:01.472000-07:00"
        },
        {
            "accountId": "123456789012",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "ENABLED",
            "updatedAt": "2023-09-11T09:57:20.438000-07:00"
        },
        {
            "accountId": "123456789012",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "ENABLED",
            "updatedAt": "2024-04-24T09:14:57.471000-07:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的使用 AWS Organizations 管理 Amazon Inspector 中的多個帳戶。 Amazon Inspector

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMembers。

`list-rules-packages`

以下程式碼範例顯示如何使用 `list-rules-packages`。

AWS CLI

列出規則套件

下列`list-rules-packages`命令列出所有可用的 Inspector 規則套件：

```
aws inspector list-rules-packages
```

輸出：

```
{
       "rulesPackageArns": [
         "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p",
         "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-H5hpSawc",
```

```
        "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-JJOtZiqQ",
        "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-vg5GGHSD"
      ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 規則套件和規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRulesPackages。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出資源的標籤

下列list-tags-for-resource命令會列出與評估範本相關聯的所有標籤，其 ARN 為 arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-gcwFliYu：

```
aws inspector list-tags-for-resource --resource-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-gcwFliYu
```

輸出：

```
{
      "tags": [
        {
              "key": "Name",
              "value": "Example"
        }
      ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估範本和評估執行。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

list-usage-totals

以下程式碼範例顯示如何使用 list-usage-totals。

AWS CLI

列出過去 30 天的用量總計

下列list-usage-totals範例列出過去 30 天的用量總計。

```
aws inspector2 list-usage-totals
```

輸出：

```
{
    "totals": [
        {
            "accountId": "123456789012",
            "usage": [
                {
                    "currency": "USD",
                    "estimatedMonthlyCost": 4.6022044647,
                    "total": 1893.4784083333334,
                    "type": "EC2_AGENTLESS_INSTANCE_HOURS"
                },
                {
                    "currency": "USD",
                    "estimatedMonthlyCost": 18.892449279,
                    "total": 10882.050784722222,
                    "type": "EC2_INSTANCE_HOURS"
                },
                {
                    "currency": "USD",
                    "estimatedMonthlyCost": 5.4525363736,
                    "total": 6543.043648333333,
                    "type": "LAMBDA_FUNCTION_CODE_HOURS"
                },
                {
                    "currency": "USD",
                    "estimatedMonthlyCost": 3.9064080309,
                    "total": 9375.379274166668,
                    "type": "LAMBDA_FUNCTION_HOURS"
                },
                {
                    "currency": "USD",
                    "estimatedMonthlyCost": 0.06,
                    "total": 6.0,
```

```
                "type": "ECR_RESCAN"
            },
            {
                "currency": "USD",
                "estimatedMonthlyCost": 0.09,
                "total": 1.0,
                "type": "ECR_INITIAL_SCAN"
            }
        ]
    }
  ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的監控 Amazon Inspector 中的用量和成本。 Amazon Inspector

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListUsageTotals。

`preview-agents`

以下程式碼範例顯示如何使用 `preview-agents`。

AWS CLI

預覽客服人員

下列`preview-agents`命令會使用 的 ARN 預覽 EC2 執行個體上安裝的代理程式，這些執行個體屬於評估目標的一部分`arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq`：

```
aws inspector preview-agents --preview-agents-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq
```

輸出：

```
{
      "agentPreviews": [
        {
              "agentId": "i-49113b93"
        }
      ]
```

```
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估目標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PreviewAgents。

register-cross-account-access-role

以下程式碼範例顯示如何使用 register-cross-account-access-role。

AWS CLI

註冊跨帳戶存取角色

當您呼叫 preview-agents 命令時，下列register-cross-account-access-role命令會將 IAM 角色註冊到 arn:aws:iam::123456789012:role/inspector Amazon Inspector 用來在評估執行開始時列出 EC2 執行個體的 ARN：

```
aws inspector register-cross-account-access-role --role-
arn arn:aws:iam::123456789012:role/inspector
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的設定 Amazon Inspector。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterCrossAccountAccessRole。

remove-attributes-from-findings

以下程式碼範例顯示如何使用 remove-attributes-from-findings。

AWS CLI

從問題清單移除屬性

下列remove-attributes-from-finding命令會將索引鍵為 Example且值為 的屬性example從 ARN 為 的調查結果中移除arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-8l1VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU：

```
aws inspector remove-attributes-from-findings --finding-arns arn:aws:inspector:us-
west-2:123456789012:target/0-0kFIPusq/template/0-8l1VIE0D/run/0-Z02cjjug/finding/0-
T8yM9mEU --attribute-keys key=Example,value=example
```

輸出：

```
{
      "failedItems": {}
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 調查結果。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveAttributesFromFindings。

set-tags-for-resource

以下程式碼範例顯示如何使用 set-tags-for-resource。

AWS CLI

設定資源的標籤

下列set-tags-for-resource命令會將索引鍵為 Example且值為 的標籤example設定為 ARN 為 的評估範本arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0：

```
aws inspector set-tags-for-resource --resource-arn arn:aws:inspector:us-
west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0 --
tags key=Example,value=example
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估範本和評估執行。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetTagsForResource。

start-assessment-run

以下程式碼範例顯示如何使用 start-assessment-run。

AWS CLI

啟動評估執行

下列start-assessment-run命令examplerun會使用 ARN 為 的評估範本啟動名為 的評估執行arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T：

```
aws inspector start-assessment-run --assessment-run-name examplerun --assessment-
template-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-
it5r2S4T
```

輸出：

```
{
      "assessmentRunArn": "arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX/template/0-it5r2S4T/run/0-jOoroxyY"
}
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估範本和評估執行。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartAssessmentRun。

stop-assessment-run

以下程式碼範例顯示如何使用 stop-assessment-run。

AWS CLI

停止評估執行

下列stop-assessment-run命令會停止使用 的 ARN 執行評估arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-jOoroxyY：

```
aws inspector stop-assessment-run --assessment-run-arn arn:aws:inspector:us-
west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-jOoroxyY
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估範本和評估執行。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopAssessmentRun。

subscribe-to-event

以下程式碼範例顯示如何使用 subscribe-to-event。

AWS CLI

訂閱事件

下列範例會啟用使用 的 ARN 將ASSESSMENT_RUN_COMPLETED事件相關 Amazon SNS 通知傳送至 主題的程序 arn:aws:sns:us-west-2:123456789012:exampletopic

```
aws inspector subscribe-to-event \
    --event ASSESSMENT_RUN_COMPLETED \
    --resource-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/
template/0-7sbz2Kz0 \
    --topic-arn arn:aws:sns:us-west-2:123456789012:exampletopic
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估範本和評估執行。Amazon Inspector

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SubscribeToEvent。

unsubscribe-from-event

以下程式碼範例顯示如何使用 unsubscribe-from-event。

AWS CLI

取消訂閱事件

下列unsubscribe-from-event命令會停用使用 的 ARN 將ASSESSMENT_RUN_COMPLETED事件相關 Amazon SNS 通知傳送至 主題的程序arn:aws:sns:us-west-2:123456789012:exampletopic：

```
aws inspector unsubscribe-from-event --event ASSESSMENT_RUN_COMPLETED --resource-
arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0
 --topic arn:aws:sns:us-west-2:123456789012:exampletopic
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估範本和評估執行。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UnsubscribeFromEvent。

update-assessment-target

以下程式碼範例顯示如何使用 update-assessment-target。

AWS CLI

更新評估目標

下列update-assessment-target命令會使用 的 ARN arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX和 的名稱更新評估目標Example，並使用 的 ARN 更新資源群組arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-yNbgL5Pt：

```
aws inspector update-assessment-target --assessment-target-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX --assessment-target-name Example --resource-group-arn arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-yNbgL5Pt
```

如需詳細資訊，請參閱《Amazon Inspector 指南》中的 Amazon Inspector 評估目標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAssessmentTarget。

update-filter

以下程式碼範例顯示如何使用 update-filter。

AWS CLI

更新篩選條件

下列update-filter範例會更新篩選條件，以省略 Lambda 調查結果，而非 ECR 執行個體調查結果。

```
aws inspector2 update-filter \
   --filter-arn "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/EXAMPLE444444444" \
   --name "ExampleSuppressionRuleLambda" \
   --description "This suppression rule omits Lambda instance findings" \
   --reason "Updating filter to omit Lambda instance findings instead of ECR instance findings"
```

輸出：

```
{
    "filters": [
```

```
        {
            "action": "SUPPRESS",
            "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/
filter/EXAMPLE444444444",
            "createdAt": "2024-05-15T21:28:27.054000+00:00",
            "criteria": {
                "resourceType": [
                    {
                        "comparison": "EQUALS",
                        "value": "AWS_ECR_INSTANCE"
                    }
                ]
            },
            "description": "This suppression rule omits Lambda instance findings",
            "name": "ExampleSuppressionRuleLambda",
            "ownerId": "o-EXAMPLE222",
            "reason": "Updating filter to omit Lambda instance findings instead of
 ECR instance findings",
            "tags": {},
            "updatedAt": "2024-05-15T22:23:13.665000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Inspector 使用者指南》中的管理 Amazon Inspector 中的問題清單。 Amazon Inspector

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateFilter。

AWS IoT 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS IoT。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

accept-certificate-transfer

以下程式碼範例顯示如何使用 accept-certificate-transfer。

AWS CLI

接受從不同 AWS 帳戶轉移的裝置憑證

下列accept-certificate-transfer範例接受從另一個 AWS 帳戶轉移的裝置憑證。憑證會以其 ID 識別。

```
aws iot accept-certificate-transfer \
    --certificate-
id 488b6a7f2acdeb00a77384e63c4e40b18bEXAMPLEe57b7272ba44c45e3448142
```

此命令不會產生任何輸出。

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的將憑證轉移到另一個帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptCertificateTransfer。

add-thing-to-billing-group

以下程式碼範例顯示如何使用 add-thing-to-billing-group。

AWS CLI

範例 1：依名稱將物件新增至帳單群組

下列add-thing-to-billing-group範例會將名為 的物件新增至名為 MyLightBulb的帳單群組GroupOne。

```
aws iot add-thing-to-billing-group \
    --billing-group-name GroupOne \
    --thing-name MyLightBulb
```

此命令不會產生輸出。

範例 2：依 ARN 將物件新增至帳單群組

下列add-thing-to-billing-group範例會將具有指定 ARN 的物件新增至具有指定 ARN 的帳單群組。如果您使用多個 AWS 區域或帳戶，指定 ARN 很有幫助。它有助於確保您將 新增至正確的區域和帳戶。

```
aws iot add-thing-to-thing-group \
    --billing-group-arn "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne" \
    --thing-arn "arn:aws:iot:us-west-2:123456789012:thing/MyOtherLightBulb"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的帳單群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddThingToBillingGroup。

add-thing-to-thing-group

以下程式碼範例顯示如何使用 add-thing-to-thing-group。

AWS CLI

將物件新增至群組

下列add-thing-to-thing-group範例會將指定的物件新增至指定的物件群組。

```
aws iot add-thing-to-thing-group \
    --thing-name MyLightBulb \
    --thing-group-name LightBulbs
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddThingToThingGroup。

associate-targets-with-job

以下程式碼範例顯示如何使用 associate-targets-with-job。

AWS CLI

將物件群組與連續任務建立關聯

下列associate-targets-with-job範例會將指定的物件群組與指定的連續任務建立關聯。

```
aws iot associate-targets-with-job \
    --targets "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \
    --job-id "example-job-04"
```

輸出：

```
{
    "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-04",
    "jobId": "example-job-04",
    "description": "example continuous job"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的建立和管理任務 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateTargetsWithJob。

attach-policy

以下程式碼範例顯示如何使用 attach-policy。

AWS CLI

範例 1：將政策連接至物件群組

下列attach-policy範例會將指定的政策連接至其 ARN 所識別的物件群組。

```
aws iot attach-policy \
    --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \
    --policy-name "UpdateDeviceCertPolicy"
```

此命令不會產生任何輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件群組。

範例 2：將政策連接至憑證

下列attach-policy範例會將政策連接至憑證指定的UpdateDeviceCertPolicy委託人。

```
aws iot attach-policy \
```

```
    --policy-name UpdateDeviceCertPolicy \
    --target "arn:aws:iot:us-
west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e"
```

此命令不會產生任何輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將 IoT 政策連接至裝置憑證。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachPolicy。

attach-security-profile

以下程式碼範例顯示如何使用 attach-security-profile。

AWS CLI

將安全性描述檔與所有未註冊裝置建立關聯

下列attach-security-profile範例會將名為 的 AWS IoT Device Defender 安全性描述檔Testprofile與此 AWS 帳戶 us-west-2區域中所有未註冊的裝置建立關聯。

```
aws iot attach-security-profile \
    --security-profile-name Testprofile \
    --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/
unregistered-things"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Detect 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachSecurityProfile。

attach-thing-principal

以下程式碼範例顯示如何使用 attach-thing-principal。

AWS CLI

將憑證連接至您的物件

下列attach-thing-principal範例會將憑證連接至 MyTemperatureSensor 物件。憑證由 ARN 識別。您可以在 AWS IoT 主控台中找到憑證的 ARN。

```
aws iot attach-thing-principal \
    --thing-name MyTemperatureSensor \
    --principal arn:aws:iot:us-
west-2:123456789012:cert/2e1eb273792174ec2b9bf4e9b37e6c6c692345499506002a351597670552 78e8
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的如何使用登錄檔管理物件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachThingPrincipal。

cancel-audit-mitigation-actions-task

以下程式碼範例顯示如何使用 cancel-audit-mitigation-actions-task。

AWS CLI

取消稽核緩解動作任務

下列cancel-audit-mitigations-action-task範例會取消指定任務的緩解動作應用程式。您無法取消已完成的任務。

```
aws iot cancel-audit-mitigation-actions-task
    --task-id "myActionsTaskId"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 CancelAuditMitigationActionsTask （緩解動作命令）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelAuditMitigationActionsTask。

cancel-audit-task

以下程式碼範例顯示如何使用 cancel-audit-task。

AWS CLI

取消稽核任務

下列cancel-audit-task範例會取消具有指定任務 ID 的稽核任務。您無法取消已完成的任務。

```
aws iot cancel-audit-task \
    --task-id a3aea009955e501a31b764abe1bebd3d
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelAuditTask。

cancel-certificate-transfer

以下程式碼範例顯示如何使用 cancel-certificate-transfer。

AWS CLI

若要取消將憑證轉移到其他 AWS 帳戶

下列cancel-certificate-transfer範例會取消指定憑證傳輸的傳輸。憑證是以憑證 ID 識別。您可以在 AWS IoT 主控台中找到憑證的 ID。

```
aws iot cancel-certificate-transfer \
    --certificate-
id f0f33678c7c9a046e5cc87b2b1a58dfa0beec26db78addd5e605d630e05c7fc8
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的將憑證轉移到另一個帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelCertificateTransfer。

cancel-job-execution

以下程式碼範例顯示如何使用 cancel-job-execution。

AWS CLI

取消裝置上的任務執行

下列cancel-job-execution範例會取消在裝置上執行指定的任務。如果任務未處於 QUEUED 狀態，您必須新增 --force 參數。

```
aws iot cancel-job-execution \
```

```
    --job-id "example-job-03" \
    --thing-name "MyRPi"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的建立和管理任務 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelJobExecution。

cancel-job

以下程式碼範例顯示如何使用 cancel-job。

AWS CLI

取消任務

下列cancel-job範例會取消指定的任務。

```
aws iot cancel-job \
    --job-job "example-job-03"
```

輸出：

```
{
    "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-03",
    "jobId": "example-job-03",
    "description": "example job test"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的建立和管理任務 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelJob。

clear-default-authorizer

以下程式碼範例顯示如何使用 clear-default-authorizer。

AWS CLI

清除預設授權方

下列clear-default-authorizer範例會清除目前設定的預設自訂授權方。執行此命令後，沒有預設授權方。當您使用自訂授權方時，您必須在 HTTP 請求標頭中依名稱指定它。

```
aws iot clear-default-authorizer
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT API 參考中的 ClearDefaultAuthorizer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ClearDefaultAuthorizer。

confirm-topic-rule-destination

以下程式碼範例顯示如何使用 confirm-topic-rule-destination。

AWS CLI

確認主題規則目的地

下列confirm-topic-rule-destination範例使用 HTTP 端點收到的確認字符來確認主題規則目的地。

```
aws iot confirm-topic-rule-destination \
    --confirmation-token "AYADeIcmtq-
ZkxfpiWIQqHWM5ucAXwABABVhd3MtY3J5cHRvLXB1YmxpYy1rZXkAREFxY1E0UmlGeDg0V21BZWZ1VjZtZWFRVUJJUkt
aywpPqg8YEsa1lD4B40aJ2s1wEHKMybiF1RoOZzYisI0IvslzQY5UmCkqq3tV-3f7-
nKfosgIAAAADAAAEAAAAAAAAAAAAAAAAAAAi9RMgy-
V19V9m6Iw2xfbw_____wAAAAEAAAAAAAAAAAAAAAEAAAB1hw4SokgUcxiJ3gTO6n50NLJVpzyQR1UmPIj5sShqXEQGc0
iufgrzTePl8RZYOWr0O6Aj9DiVzJZx-1iD6Pu-
G6PUw1kaO7Knzs2B4AD0qfrHUF4pYRTvyUgBnMGUCMQC8ZRmhKqntd_c6Kgrow3bMUDBvNqo2qZr8Z8Jm2rzgseROlAn
PIetJ803Z4IlIlF8xXlcdPGP-PV1dOXFemyL8g"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的確認主題規則目的地。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ConfirmTopicRuleDestination。

create-audit-suppression

以下程式碼範例顯示如何使用 create-audit-suppression。

AWS CLI

建立稽核問題清單抑制

下列create-audit-suppression範例會為名為 "virtualMachinePolicy" 的政策建立稽核調查結果抑制，該政策已標記為過度寬鬆。

```
aws iot create-audit-suppression \
    --check-name IOT_POLICY_OVERLY_PERMISSIVE_CHECK \
    --resource-identifier
 policyVersionIdentifier={"policyName"="virtualMachinePolicy","policyVersionId"="1"}
 \
    --no-suppress-indefinitely \
    --expiration-date 2020-10-20
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核調查結果抑制。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAuditSuppression。

create-authorizer

以下程式碼範例顯示如何使用 create-authorizer。

AWS CLI

建立自訂授權方

下列create-authorizer範例會建立自訂授權方，使用指定的 Lambda 函數做為自訂身分驗證服務的一部分。

```
   aws iot create-authorizer \
       --authorizer-name "CustomAuthorizer" \
       --authorizer-function-arn "arn:aws:lambda:us-
west-2:123456789012:function:CustomAuthorizerFunction" \
       --token-key-name "MyAuthToken" \
       --status ACTIVE \
       --token-signing-public-keys FIRST_KEY="-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA1uJOB4lQPgG/lM6ZfIwo
Z+7ENxAio9q6QD4FFqjGZsvjtYwjoe1RKK0U8Eq9xb5O3kRSmyIwTzwzm/f4Gf0Y
ZUloJ+t3PUUwHrmbYTAgTrCUgRFygjfgVwGCPs5ZAX4Eyqt5cr+AIHIiUDbxSa7p
```

```
zwOBKPeic0asNJpqT8PkBbRaKyleJh5oo81NDHHmVtbBm5A5YiJjqYXLaVAowKzZ
+GqsNvAQ9Jy1wI2VrEa1OfL8flDB/BJLm7zjpfPOHDJQgID0XnZwAlNnZcOhCwIx
50g2LW2Oy9R/dmqtDmJiVP97Z4GykxPvwlYHrUXY0iW1R3AR/Ac1NhCTGZMwVDB1
lQIDAQAB
-----END PUBLIC KEY-----"
```

輸出：

```
{
    "authorizerName": "CustomAuthorizer",
    "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer2"
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 CreateAuthorizer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAuthorizer。

create-billing-group

以下程式碼範例顯示如何使用 create-billing-group。

AWS CLI

建立帳單群組

下列create-billing-group範例會建立名為 的簡單帳單群組GroupOne。

```
aws iot create-billing-group \
    --billing-group-name GroupOne
```

輸出：

```
{
    "billingGroupName": "GroupOne",
    "billingGroupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne",
    "billingGroupId": "103de383-114b-4f51-8266-18f209ef5562"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的帳單群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateBillingGroup。

create-certificate-from-csr

以下程式碼範例顯示如何使用 create-certificate-from-csr。

AWS CLI

從憑證簽署請求 (CSR) 建立裝置憑證

下列create-certificate-from-csr範例會從 CSR 建立裝置憑證。您可以使用 openssl命令來建立 CSR。

```
aws iot create-certificate-from-csr \
    --certificate-signing-request=file://certificate.csr
```

輸出：

```
{
    "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/
c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbee1428d216d54d53ac9",
        "certificateId":
 "c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbee1428d216d54d53ac9",
        "certificatePem": "<certificate-text>"
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 CreateCertificateFromCSR。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCertificateFromCsr。

create-custom-metric

以下程式碼範例顯示如何使用 create-custom-metric。

AWS CLI

建立裝置發佈至 Device Defender 的自訂指標

下列create-custom-metric範例會建立測量電池百分比的自訂指標。

```
aws iot create-custom-metric \
    --metric-name "batteryPercentage" \
    --metric-type "number" \
```

```
    --display-name "Remaining battery percentage." \
    --region us-east-1 \
    --client-request-token "02ccb92b-33e8-4dfa-a0c1-35b181ed26b0"
```

輸出：

```
{
    "metricName": "batteryPercentage",
    "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/
batteryPercentage"
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的自訂指標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCustomMetric。

create-dimension

以下程式碼範例顯示如何使用 create-dimension。

AWS CLI

建立維度

以下create-dimension使用名為 的單一主題篩選條件建立維度TopicFilterForAuthMessages。

```
aws iot create-dimension \
    --name TopicFilterForAuthMessages \
    --type TOPIC_FILTER \
    --string-values device/+/auth
```

輸出：

```
{
    "name": "TopicFilterForAuthMessages",
    "arn": "arn:aws:iot:eu-west-2:123456789012:dimension/TopicFilterForAuthMessages"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Detect 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDimension。

create-domain-configuration

以下程式碼範例顯示如何使用 create-domain-configuration。

AWS CLI

建立網域組態

下列create-domain-configuration範例會建立服務類型為 的受 AWS管網域組態DATA。

```
aws iot create-domain-configuration \
    --domain-configuration-name "additionalDataDomain" \
    --service-type "DATA"
```

輸出：

```
{
    "domainConfigurationName": "additionalDataDomain",
    "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的可設定端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDomainConfiguration。

create-dynamic-thing-group

以下程式碼範例顯示如何使用 create-dynamic-thing-group。

AWS CLI

建立動態物件群組

下列create-dynamic-thing-group範例會建立動態物件群組，其中包含溫度屬性大於 60 度的任何物件。您必須啟用 AWS IoT 機群索引，才能使用動態物件群組。

```
aws iot create-dynamic-thing-group \
    --thing-group-name "RoomTooWarm" \
```

```
    --query-string "attributes.temperature>60"
```

輸出：

```
{
    "thingGroupName": "RoomTooWarm",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RoomTooWarm",
    "thingGroupId": "9d52492a-fc87-43f4-b6e2-e571d2ffcad1",
    "indexName": "AWS_Things",
    "queryString": "attributes.temperature>60",
    "queryVersion": "2017-09-30"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的動態物件群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDynamicThingGroup。

create-job

以下程式碼範例顯示如何使用 create-job。

AWS CLI

範例 1：建立任務

下列create-job範例會建立將 JSON 文件傳送至MyRaspberryPi裝置的 Simple AWS IoT 任務。

```
aws iot create-job \
    --job-id "example-job-01" \
    --targets "arn:aws:iot:us-west-2:123456789012:thing/MyRaspberryPi" \
    --document file://example-job.json \
    --description "example job test" \
    --target-selection SNAPSHOT
```

輸出：

```
{
    "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",
    "jobId": "example-job-01",
    "description": "example job test"
```

```
}
```

範例 2：建立連續任務

下列create-job範例會建立任務，該任務會在指定為目標的物件完成任務後繼續執行。在此範例中，目標為物件群組，因此當新裝置新增至群組時，連續任務會在這些新物件上執行。

aws iot create-job --job-id "example-job-04" --targets "arn：aws：iot：us-west-2：123456789012：thinggroup/DeadBulbs" --document file：/example-job.json --description "example continuous job" --target-selection CONTINUOUS

輸出：

```
{
    "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-04",
    "jobId": "example-job-04",
    "description": "example continuous job"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的建立和管理任務 (CLI)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateJob。

create-keys-and-certificate

以下程式碼範例顯示如何使用 create-keys-and-certificate。

AWS CLI

建立 RSA 金鑰對並發行 X.509 憑證

以下內容create-keys-and-certificate會建立 2048 位元 RSA 金鑰對，並使用發行的公有金鑰發行 X.509 憑證。由於這是 AWS IoT 為此憑證提供私有金鑰的唯一時間，請務必將其保存在安全的位置。

```
aws iot create-keys-and-certificate \
    --certificate-pem-outfile "myTest.cert.pem" \
    --public-key-outfile "myTest.public.key" \
    --private-key-outfile "myTest.private.key"
```

輸出：

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
    "certificateId":
 "9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
    "certificatePem": "
-----BEGIN CERTIFICATE-----
MIICiTCCEXAMPLE6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgEXAMPLEAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSEXAMPLE2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYEXAMPLEb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCEXAMPLEJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDAEXAMPLEsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEXAMPLE25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+aEXAMPLE
EXAMPLEfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZEXAMPLELG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAEXAMPLEWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9qEXAMPLEyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDEXAMPLEBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkEXAMPLEQEFAAOCAQ8AMIIBCgKCAQEAEXAMPLE1nnyJwKSMHw4h\nMMEXAMPLEuuN/
dMAS3fyce8DW/4+EXAMPLEyjmoF/YVF/gHr99VEEXAMPLE5VF13\n59VK7cEXAMPLE67GK+y+jikqXOgHh/
xJTwo
+sGpWEXAMPLEDz18xOd2ka4tCzuWEXAMPLEahJbYkCPUBSU8opVkR7qkEXAMPLE1DR6sx2HocliOOLtu6Fkw91swQWEX
\GB3ZPrNh0PzQYvjUStZeccyNCx2EXAMPLEvp9mQOUXP6plfgxwKRX2fEXAMPLEDa
\nhJLXkX3rHU2xbxJSq7D+XEXAMPLEcw+LyFhI5mgFRl88eGdsAEXAMPLElnI9EesG\nFQIDAQAB\n-----
END PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----\nkey omittted for security
 reasons\n-----END RSA PRIVATE KEY-----\n"
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的建立和註冊 IoT 裝置憑證。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateKeysAndCertificate。

create-mitigation-action

以下程式碼範例顯示如何使用 create-mitigation-action。

AWS CLI

建立緩解動作

下列create-mitigation-action範例定義了名為 的緩解動作，在套用時，AddThingsToQuarantineGroup1Action會將物件移至名為 的物件群組QuarantineGroup1。此動作會覆寫動態物件群組。

```
aws iot create-mitigation-action --cli-input-json file::params.json
```

params.json 的內容：

```
{
    "actionName": "AddThingsToQuarantineGroup1Action",
    "actionParams": {
        "addThingsToThingGroupParams": {
            "thingGroupNames": [
                "QuarantineGroup1"
            ],
            "overrideDynamicGroups": true
        }
    },
    "roleArn": "arn:aws:iam::123456789012:role/service-role/
MoveThingsToQuarantineGroupRole"
}
```

輸出：

```
{
    "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
AddThingsToQuarantineGroup1Action",
    "actionId": "992e9a63-a899-439a-aa50-4e20c52367e1"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 CreateMitigationAction（緩解動作命令）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateMitigationAction。

create-ota-update

以下程式碼範例顯示如何使用 create-ota-update。

AWS CLI

建立 OTA 更新以搭配 Amazon FreeRTOS 使用

下列create-ota-update範例會在物件或群組的目標群組上建立 AWS IoT OTAUpdate。這是 Amazon FreeRTOS over-the-air更新的一部分，可讓您將新的韌體映像部署到單一裝置或一組裝置。

```
aws iot create-ota-update \
    --cli-input-json file://create-ota-update.json
```

create-ota-update.json 的內容：

```
{
    "otaUpdateId": "ota12345",
    "description": "A critical update needed right away.",
    "targets": [
        "device1",
        "device2",
        "device3",
        "device4"
    ],
    "targetSelection": "SNAPSHOT",
    "awsJobExecutionsRolloutConfig": {
        "maximumPerMinute": 10
    },
    "files": [
        {
          "fileName": "firmware.bin",
          "fileLocation": {
            "stream": {
              "streamId": "004",
              "fileId":123
            }
          },
          "codeSigning": {
            "awsSignerJobId": "48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"
          }
        }
    ]
    "roleArn": "arn:aws:iam:123456789012:role/service-role/my_ota_role"
}
```

輸出：

```
{
     "otaUpdateId": "ota12345",
     "awsIotJobId": "job54321",
     "otaUpdateArn": "arn:aws:iot:us-west-2:123456789012:otaupdate/itsaupdate",
     "awsIotJobArn": "arn:aws:iot:us-west-2:123456789012:job/itsajob",
     "otaUpdateStatus": "CREATE_IN_PROGRESS"
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 CreateOTAUpdate。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateOtaUpdate。

create-policy-version

以下程式碼範例顯示如何使用 create-policy-version。

AWS CLI

使用新版本更新政策

下列create-policy-version範例會更新政策定義，建立新的政策版本。此範例也會將新版本設為預設值。

```
aws iot create-policy-version \
    --policy-name UpdateDeviceCertPolicy \
    --policy-document file://policy.json \
    --set-as-default
```

policy.json 的內容：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action":  "iot:UpdateCertificate",
            "Resource": "*"
        }
    ]
}
```

輸出：

```
{
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/UpdateDeviceCertPolicy",
    "policyDocument": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\":
 \"Allow\", \"Action\":  \"iot:UpdateCertificate\", \"Resource\": \"*\" } ] }",
    "policyVersionId": "2",
    "isDefaultVersion": true
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 IoT 政策。 AWS IoT

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreatePolicyVersion。

create-policy

以下程式碼範例顯示如何使用 create-policy。

AWS CLI

建立 AWS IoT 政策

下列create-policy範例會建立名為 TemperatureSensorPolicy 的 AWS IoT 政策。policy.json 檔案包含允許 AWS IoT 政策動作的陳述式。

```
aws iot create-policy \
    --policy-name TemperatureSensorPolicy \
    --policy-document file://policy.json
```

policy.json 的內容：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish",
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:us-west-2:123456789012:topic/topic_1",
```

```
                "arn:aws:iot:us-west-2:123456789012:topic/topic_2"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-west-2:123456789012:topicfilter/topic_1",
                "arn:aws:iot:us-west-2:123456789012:topicfilter/topic_2"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-west-2:123456789012:client/basicPubSub"
            ]
        }
    ]
}
```

輸出：

```
{
    "policyName": "TemperatureSensorPolicy",
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TemperatureSensorPolicy",
    "policyDocument": "{
        \"Version\": \"2012-10-17\",
        \"Statement\": [
            {
                \"Effect\": \"Allow\",
                \"Action\": [
                    \"iot:Publish\",
                    \"iot:Receive\"
                ],
                \"Resource\": [
                    \"arn:aws:iot:us-west-2:123456789012:topic/topic_1\",
                    \"arn:aws:iot:us-west-2:123456789012:topic/topic_2\"
```

```
                ]
            },
            {
                \"Effect\": \"Allow\",
                \"Action\": [
                    \"iot:Subscribe\"
                ],
                \"Resource\": [
                    \"arn:aws:iot:us-west-2:123456789012:topicfilter/topic_1\",
                    \"arn:aws:iot:us-west-2:123456789012:topicfilter/topic_2\"
                ]
            },
            {
                \"Effect\": \"Allow\",
                \"Action\": [
                    \"iot:Connect\"
                ],
                \"Resource\": [
                    \"arn:aws:iot:us-west-2:123456789012:client/basicPubSub\"
                ]
            }
        ]
    }",
    "policyVersionId": "1"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 IoT 政策。 AWS IoT

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreatePolicy。

create-provisioning-claim

以下程式碼範例顯示如何使用 create-provisioning-claim。

AWS CLI

建立佈建宣告

下列create-provisioning-claim範例會從佈建範本建立佈建宣告。

```
aws iot create-provisioning-claim \
    --template-name MyTestProvisioningTemplate
```

輸出：


```
{
    "certificateId":
 "78de02184b2ce80cf8fb709bda59e62b19fb83513590483eb0434589476ab09f",
    "certificatePem": "-----BEGIN CERTIFICATE-----\nMIIDdzCCAl
+gAwIBAgIUXSZhEBLztMLZ2fHG
14gV0NymYY0wDQYJKoZIhvcNAQEL
\nBQAwfjELMAkGA1UEBhMCVVMxEzARBgNVBAgMCldhc2hpbmd0b24xEDAOBg
VBAcM\nB1NlYXR0bGUxGDAWBgNVBAoMD0FtYXpvbi5jb20gSW5jLjEgMB4GA1UECwwXQW1h
\nem9uIElvVCBQcm9
2aXNpb25pbmcxDDAKBgNVBAUTAzEuMDAeFw0yMDA3MjgxNjQ0\nMDZaFw0yMDA3MjgxNjUxMDZaMEsxSTBHBgNVB
AMMQDFhNDEyM2VkNmIxYjU3MzE3\nZTgzMTJmY2MzN2FiNTdhY2MzYTZkZGVjOGQ5OGY3NzUwMWRlMjc0YjhmYTQ
xN2Iw\nggEiMA0GCSqGSIb3EXAMPLEAA4IBDwAwggEKAoIBAQDBhKI94ktKLqTwnj+ayOq1\nTAJt/
N6s6IJDZvl
rYjkC0E7wzaeY3TprWk03S29vUzVuEOXHXQXZbihgpg2m6fza\nkWm9/
wpjzE9ny5+xkPGVH4Wnwz7yK5m8S0agL
T96cRBSWnWmonOWdY0GKVzni0CA\n+iyGudgrFKm7Eae/
v18oXrf82KtOAGO4xG0KE2WKYHsT1fx3c9xZhlXP/eX
Lhv00\n+lGp0WVw9PbhKfrxliKJ5q6sL5nVUaUHq6hlQPYwsATeOvAp3u0ak5zgTyL0fg7Y
\nPyKk6VYwLW62r+V
YBSForEMOAhkq3LsP/rjxpEKmi2W4lPVS6oFZRKcD+H1Kyil5\nAgMBAAGjIDAeMAwGA1UdEwEB/
wQCMAAwDgYDV
R0PAQH/BAQDAgeAMA0GCSqGSIb3\nDQEBCwUAA4IBAQAGgix2k6nVqbZFKq97/fZBzLGS0dyz5rT/
E41cDIRX+1j
EPW41\nw0D+2sXheCZLZZnSkvIiP74IToNeXDrjdcaodeGFVHIElRjhMIq+4ZebPbRLtidF
\nRc2hfcTAlqq9Z6v
5Vk6BeM1tu0RqH1wPoVUccLPya8EjNCbnJZUmGdOfrN/Y9pho\n5ikV+HPeZhG/k6dhE2GsQJyKFVHL/
uBgKSily
1bRyWU1r6qcpWBNBHjUoD7HgOwD
\nnzMh4XRb2FQDsqFalkCSYmeL8IVC49sgPD9Otyp5uteGMTy62usAAUQdq/f
ZvrWg\nOkFpwMVnGKVKT7Kg0kKOLzKWOBB2Jm4/gmrJ\n-----END CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCg
KCAQEAwYSiPeJLSi6k8J4/msjq
\ntUwCbfzerOiCQ2b5a2I5AtBO8M2nmN06a1pNN0tvb1M1bhDlx10F2W4oYKYN
pun8\n2pFpvf8KY8xPZ8ufsZDxlR+Fp8M+8iuZvEtGoC0/enEQUlp1pqJzlnWNBilc54tA
\ngPoshrnYKxSpuxGn
v79fKF63/NirTgBjuMRtChNlimEXAMPLE3PcWYZVz/3ly4b9\nNPpRqdFlcPT24Sn68ZYiieaurC
+Z1VGlB6uoZU
D2MLAE3jrwKd7tGpOc4E8i9H4O\n2D8ipOlWMC1utq/
lWAUhaKxDDgIZKty7D/648aRCpotluJT1UuqBWUSnA/h9
Ssop\neQIDAQAB\n-----END PUBLIC KEY-----\n",
```

```
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQEAwYSiPeJLSi6k8J4/
msjqtUwCbfzerOiCQ2b5a2I5AtBO8M2n
\nmN06a1pNN0tvb1M1bhDlx10F2W4oYKYNpun82pFpvf8KY8xPZ8ufsZ
DxlR+Fp8M+\n8iuZvEtGoC0/enEQUlp1pqJzlnWNBilc54tAgPoshrnYKxSpuxGnv79fKF63/Nir
\nTgBjuMRtCh
NlimB7E9X8d3PcWYZVz/3ly4b9NPpRqdFlcPT24Sn68ZYiieaurC+Z
\n1VGlB6uoZUD2MLAE3jrwKd7tGpOc4E8i
9H4O2D8ipOlWMC1utq/lWAUhaKxDDgIZ\nKty7D/648aRCpotluJT1UuqBWUSnA/
h9SsopeQIDAQABAoIBAEAybN
QUtx9T2/nK\ntZT2pA4iugecxI4dz+DmT0XVXs5VJmrx/
nBSq6ejXExEpSIMO4RY7LE3ZdJcnd56\nF7tQkkY7yR
VzfxHeXFU1krOIPuxWebNOrRoPZr+1RSer+wv2aBC525+88pVuR6tM
\nm3pgkrR2ycCj9FdOUoQxdjHBHaM5PDmJ
9aSxCKdg3nReepeGwsR2TQA+m2vVxWk7\nou0+91eTOP+/QfP7P8ZjOIkO2XivlRcVDyN/
E4QXPKuIkM/8vS8VK+
E9pATQ0MtB\n2lw8R/YU5AJd6jlEXAMPLEGU2UzRzInNWiLtkPPPqgqXXhxOf+mxByjcMalVJk0L
\nhOG2ROUCgY
EA+ROcHNHy/XbsP7FihOhEh+6Q2QxQ2ncBUPYbBazrR8Hn+7SCICQK
\nVyYfd8Ajfq3e7RsKVL5SlMBp7Slidxak
bIn28fKfPn62DaemGCIoyDgLpF+eUxBx
\ngzbCiBZga8brfurza43UZjKZLpg3hq721+FeAiXi1Nma4Yr9YWEHEN
8CgYEAxuWt\npzdWWmsiFzfsAw0sy9ySDA/xr5WRWzJyAqUsjsks6rxNzWebpufnYHcmtW7pLdqM
\nkboHwN2pXa
kmZvrk2nKkEMq5brBYGDXuxDe+V369Bianx8aZFyIsckA7OwXW1w1h
\ngRC5rQ4XOgp3+Jmw7eAO8LRYDjaN846+
QbtO2KcCgYAWS0UL51bijQR0ZwI0dz27\nFQVuCAYsp748aurcRTACCj8jbnK/
QbqTNlxWsaH7ssBjZKo2D5sAqY
BRtASWODab\naHXsDhVm2Jye+ESLoHMaCLoyCkT3ll8yqXIcEDStMO7fO1Ryag164EiJvSIrMfny\nNL/
fXVjCSH
/udCxdzPt+7QKBgQC+LAD7rxdr4J9538hTqpc4XK9vxRbrMXEH55XH
\nHbMa2xONZXpmeTgEQBukyohCVceyRhK9
i0e6irZTjVXghOeoTpC8VXkzcnzouTiQ
\neFQQSGfnp7Ioe6UIz23715pKduzSNkMSKrG924ktv7CyDBF1gBQI5g
aDoHnddJBJ\nPRTIZQKBgA8MASXtTxQntRwXXzR92U0vAighiuRkB/mx9jQpUcK1qiqHbkAMqgNF
\nPFCBYIUbFT
iYKKKeJNbyJQvjfsJCkAnaFJ+RnTxk0Q6Wjm20peJ/ii4QiDdnigoE\nvdlc5cFQewWb4/
zqAtPdinkPlN94ileI
79XQdc7RlJ0jpgTimL+V\n-----END RSA PRIVATE KEY-----\n"
    },
    "expiration": 1595955066.0
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的由信任的使用者佈建。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateProvisioningClaim。

create-provisioning-template-version

以下程式碼範例顯示如何使用 create-provisioning-template-version。

AWS CLI

建立佈建範本版本

下列範例會為指定的佈建範本建立版本。檔案 提供新版本的內文template.json。

```
aws iot create-provisioning-template-version \
    --template-name widget-template \
    --template-body file://template.json
```

template.json 的內容：

```
{
    "Parameters" : {
        "DeviceLocation": {
            "Type": "String"
        }
    },
    "Mappings": {
        "LocationTable": {
            "Seattle": {
                "LocationUrl": "https://example.aws"
            }
        }
    },
    "Resources" : {
        "thing" : {
            "Type" : "AWS::IoT::Thing",
            "Properties" : {
                "AttributePayload" : {
                    "version" : "v1",
                    "serialNumber" : "serialNumber"
                },
                "ThingName" : {"Fn::Join":["",["ThingPrefix_",
{"Ref":"SerialNumber"}]]},
```

```
                "ThingTypeName" : {"Fn::Join":["",["ThingTypePrefix_",
{"Ref":"SerialNumber"}]]},
                "ThingGroups" : ["widgets", "WA"],
                "BillingGroup": "BillingGroup"
            },
            "OverrideSettings" : {
                "AttributePayload" : "MERGE",
                "ThingTypeName" : "REPLACE",
                "ThingGroups" : "DO_NOTHING"
            }
        },
        "certificate" : {
            "Type" : "AWS::IoT::Certificate",
            "Properties" : {
                "CertificateId": {"Ref": "AWS::IoT::Certificate::Id"},
                "Status" : "Active"
            }
        },
        "policy" : {
            "Type" : "AWS::IoT::Policy",
            "Properties" : {
                "PolicyDocument" : {
                    "Version": "2012-10-17",
                    "Statement": [{
                        "Effect": "Allow",
                        "Action":["iot:Publish"],
                        "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/foo/
bar"]
                    }]
                }
            }
        }
    },
    "DeviceConfiguration": {
        "FallbackUrl": "https://www.example.com/test-site",
        "LocationUrl": {
            "Fn::FindInMap": ["LocationTable",{"Ref": "DeviceLocation"},
 "LocationUrl"]}
        }
    }
}
```

輸出：

```
{
    "templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/widget-
template",
    "templateName": "widget-template",
    "versionId": 2,
    "isDefaultVersion": false
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的 IoT 安全通道。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateProvisioningTemplateVersion。

create-provisioning-template

以下程式碼範例顯示如何使用 create-provisioning-template。

AWS CLI

建立佈建範本

下列create-provisioning-template範例會建立檔案 定義的佈建範本template.json。

```
aws iot create-provisioning-template \
    --template-name widget-template \
    --description "A provisioning template for widgets" \
    --provisioning-role-arn arn:aws:iam::123456789012:role/Provision_role \
    --template-body file://template.json
```

template.json 的內容：

```
{
    "Parameters" : {
        "DeviceLocation": {
            "Type": "String"
        }
    },
    "Mappings": {
        "LocationTable": {
            "Seattle": {
                "LocationUrl": "https://example.aws"
            }
        }
    },
```

```
    "Resources" : {
        "thing" : {
            "Type" : "AWS::IoT::Thing",
            "Properties" : {
                "AttributePayload" : {
                    "version" : "v1",
                    "serialNumber" : "serialNumber"
                },
                "ThingName" : {"Fn::Join":["",["ThingPrefix_",
{"Ref":"SerialNumber"}]]},
                "ThingTypeName" : {"Fn::Join":["",["ThingTypePrefix_",
{"Ref":"SerialNumber"}]]},
                "ThingGroups" : ["widgets", "WA"],
                "BillingGroup": "BillingGroup"
            },
            "OverrideSettings" : {
                "AttributePayload" : "MERGE",
                "ThingTypeName" : "REPLACE",
                "ThingGroups" : "DO_NOTHING"
            }
        },
        "certificate" : {
            "Type" : "AWS::IoT::Certificate",
            "Properties" : {
                "CertificateId": {"Ref": "AWS::IoT::Certificate::Id"},
                "Status" : "Active"
            }
        },
        "policy" : {
            "Type" : "AWS::IoT::Policy",
            "Properties" : {
                "PolicyDocument" : {
                    "Version": "2012-10-17",
                    "Statement": [{
                        "Effect": "Allow",
                        "Action":["iot:Publish"],
                        "Resource": ["arn:aws:iot:us-east-1:504350838278:topic/foo/
bar"]
                    }]
                }
            }
        }
    },
    "DeviceConfiguration": {
```

```
        "FallbackUrl": "https://www.example.com/test-site",
        "LocationUrl": {
            "Fn::FindInMap": ["LocationTable",{"Ref": "DeviceLocation"},
 "LocationUrl"]}
        }
    }
}
```

輸出：

```
{
    "templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/widget-
template",
    "templateName": "widget-template",
    "defaultVersionId": 1
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的 IoT 安全通道。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateProvisioningTemplate。

create-role-alias

以下程式碼範例顯示如何使用 create-role-alias。

AWS CLI

建立角色別名

下列create-role-alias範例會LightBulbRole為指定的角色建立名為 的角色別名。

```
aws iot create-role-alias \
    --role-alias LightBulbRole \
    --role-arn arn:aws:iam::123456789012:role/lightbulbrole-001
```

輸出：

```
{
    "roleAlias": "LightBulbRole",
    "roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/LightBulbRole"
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 CreateRoleAlias。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRoleAlias。

create-scheduled-audit

以下程式碼範例顯示如何使用 create-scheduled-audit。

AWS CLI

建立排定的稽核

下列create-scheduled-audit範例會建立排程稽核，每週在星期三執行，以檢查 CA 憑證或裝置憑證是否即將到期。

```
aws iot create-scheduled-audit \
    --scheduled-audit-name WednesdayCertCheck \
    --frequency WEEKLY \
    --day-of-week WED \
    --target-check-
names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK
```

輸出：

```
{
    "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
WednesdayCertCheck"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateScheduledAudit。

create-security-profile

以下程式碼範例顯示如何使用 create-security-profile。

AWS CLI

建立安全性描述檔

下列create-security-profile範例會建立安全性描述檔，檢查行動頻寬是否超過閾值，或在五分鐘內發生超過 10 次授權失敗。

```
aws iot create-security-profile \
    --security-profile-name PossibleIssue \
    --security-profile-description "Check to see if authorization fails 10 times in
 5 minutes or if cellular bandwidth exceeds 128"  \
    --behaviors "[{\"name\":\"CellularBandwidth\",\"metric\":\"aws:message-byte-size
\",\"criteria\":{\"comparisonOperator\":\"greater-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}},{\"name
\":\"Authorization\",\"metric\":\"aws:num-authorization-failures\",\"criteria\":
{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":10},\"durationSeconds
\":300,\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]"
```

輸出：

```
{
    "securityProfileName": "PossibleIssue",
    "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/
PossibleIssue"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Detect 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSecurityProfile。

create-stream

以下程式碼範例顯示如何使用 create-stream。

AWS CLI

建立串流，以透過 MQTT 在區塊中交付一或多個大型檔案

下列create-stream範例會建立串流，以透過 MQTT 在區塊中交付一或多個大型檔案。串流自 S3 等來源傳輸的資料位元組區塊，會封包為 MQTT 訊息。一個串流可與一個或多個檔案建立關聯。

```
aws iot create-stream \
    --cli-input-json file://create-stream.json
```

create-stream.json 的內容：

```
{
    "streamId": "stream12345",
```

```
    "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",
    "files": [
        {
            "fileId": 123,
            "s3Location": {
                "bucket":"codesign-ota-bucket",
                "key":"48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"
            }
        }
    ],
    "roleArn": "arn:aws:iam:123456789012:role/service-role/my_ota_stream_role"
}
```

輸出：

```
{
     "streamId": "stream12345",
     "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream12345",
     "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",
     "streamVersion": "1"
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 CreateStream。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateStream。

create-thing-group

以下程式碼範例顯示如何使用 create-thing-group。

AWS CLI

範例 1：建立物件群組

下列create-thing-group範例會建立名為 的物件群組，LightBulbs其中包含描述和兩個屬性。

```
aws iot create-thing-group \
    --thing-group-name LightBulbs \
    --thing-group-properties "thingGroupDescription=\"Generic bulb group\",
 attributePayload={attributes={Manufacturer=AnyCompany,wattage=60}}"
```

輸出：

```
{
    "thingGroupName": "LightBulbs",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs",
    "thingGroupId": "9198bf9f-1e76-4a88-8e8c-e7140142c331"
}
```

範例 2：建立屬於父群組的物件群組

以下內容會create-thing-group建立名為 的物件群組HalogenBulbs，其父物件群組名為 LightBulbs。

```
aws iot create-thing-group \
    --thing-group-name HalogenBulbs \
    --parent-group-name LightBulbs
```

輸出：

```
{
    "thingGroupName": "HalogenBulbs",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/HalogenBulbs",
    "thingGroupId": "f4ec6b84-b42b-499d-9ce1-4dbd4d4f6f6e"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateThingGroup。

create-thing-type

以下程式碼範例顯示如何使用 create-thing-type。

AWS CLI

定義物件類型

下列create-thing-type範例定義物件類型和相關聯的屬性。

```
aws iot create-thing-type \
    --thing-type-name "LightBulb" \
    --thing-type-properties "thingTypeDescription=light bulb type,
 searchableAttributes=wattage,model"
```

輸出：

```
{
    "thingTypeName": "LightBulb",
    "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb",
    "thingTypeId": "ce3573b0-0a3c-45a7-ac93-4e0ce14cd190"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件類型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateThingType。

create-thing

以下程式碼範例顯示如何使用 create-thing。

AWS CLI

範例 1：在登錄檔中建立物件記錄

下列create-thing範例會在 AWS IoT 物件登錄檔中為裝置建立項目。

```
aws iot create-thing \
    --thing-name SampleIoTThing
```

輸出：

```
{
    "thingName": "SampleIoTThing",
    "thingArn": "arn:aws:iot:us-west-2: 123456789012:thing/SampleIoTThing",
    "thingId": " EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE "
}
```

範例 2：定義與物件類型相關聯的物件

下列create-thing範例會建立具有指定物件類型及其屬性的物件。

```
aws iot create-thing \
    --thing-name "MyLightBulb" \
    --thing-type-name "LightBulb" \
    --attribute-payload "{"attributes": {"wattage":"75", "model":"123"}}"
```

輸出：

```
{
    "thingName": "MyLightBulb",
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
    "thingId": "40da2e73-c6af-406e-b415-15acae538797"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的如何使用登錄檔和物件類型管理物件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateThing。

create-topic-rule-destination

以下程式碼範例顯示如何使用 create-topic-rule-destination。

AWS CLI

建立主題規則目的地

下列create-topic-rule-destination範例會建立 HTTP 端點的主題規則目的地。

```
aws iot create-topic-rule-destination \
    --destination-configuration httpUrlConfiguration={confirmationUrl=https://
example.com}
```

輸出：

```
{
    "topicRuleDestination": {
        "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "status": "IN_PROGRESS",
        "statusReason": "Awaiting confirmation. Confirmation message sent on
 2020-07-09T22:47:54.154Z; no response received from the endpoint.",
        "httpUrlProperties": {
            "confirmationUrl": "https://example.com"
        }
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的建立主題規則目的地。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTopicRuleDestination。

create-topic-rule

以下程式碼範例顯示如何使用 create-topic-rule。

AWS CLI

建立傳送 Amazon SNS 提醒的規則

下列create-topic-rule範例會建立規則，當裝置陰影中找到的土壤濕度等級讀數偏低時，傳送 Amazon SNS 訊息。

```
aws iot create-topic-rule \
    --rule-name "LowMoistureRule" \
    --topic-rule-payload file://plant-rule.json
```

此範例需要將下列 JSON 程式碼儲存至名為 的檔案plant-rule.json：

```
{
    "sql": "SELECT * FROM '$aws/things/MyRPi/shadow/update/accepted' WHERE
 state.reported.moisture = 'low'\n",
    "description": "Sends an alert whenever soil moisture level readings are too
 low.",
    "ruleDisabled": false,
    "awsIotSqlVersion": "2016-03-23",
    "actions": [{
            "sns": {
                "targetArn": "arn:aws:sns:us-
west-2:123456789012:MyRPiLowMoistureTopic",
                "roleArn": "arn:aws:iam::123456789012:role/service-role/
MyRPiLowMoistureTopicRole",
                "messageFormat": "RAW"
            }
    }]
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的建立 IoT 規則。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTopicRule。

delete-account-audit-configuration

以下程式碼範例顯示如何使用 delete-account-audit-configuration。

AWS CLI

停用您 AWS 帳戶的所有稽核檢查

下列delete-account-audit-configuration範例會還原此帳戶的 AWS IoT Device Defender 預設設定，停用所有稽核檢查並清除組態資料。它也會刪除此帳戶的任何排程稽核。請謹慎使用此命令。

```
aws iot delete-account-audit-configuration \
    --delete-scheduled-audits
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAccountAuditConfiguration。

delete-audit-suppression

以下程式碼範例顯示如何使用 delete-audit-suppression。

AWS CLI

刪除稽核問題清單禁止

下列delete-audit-suppression範例會刪除 DEVICE_CERTIFICATE_EXPIRING_CHECK 的稽核調查結果抑制。

```
aws iot delete-audit-suppression \
    --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
    --resource-identifier deviceCertificateId="c7691e<shortened>"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核調查結果抑制。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAuditSuppression。

delete-authorizer

以下程式碼範例顯示如何使用 delete-authorizer。

AWS CLI

刪除自訂授權方

下列delete-authorizer範例會刪除名為 的授權方CustomAuthorizer。自訂授權方必須處於 INACTIVE 狀態，您才能將其刪除。

```
aws iot delete-authorizer \
    --authorizer-name CustomAuthorizer
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 DeleteAuthorizer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAuthorizer。

delete-billing-group

以下程式碼範例顯示如何使用 delete-billing-group。

AWS CLI

刪除帳單群組

下列delete-billing-group範例會刪除指定的帳單群組。即使帳單群組包含一或多個物件，您也可以將其刪除。

```
aws iot delete-billing-group \
    --billing-group-name BillingGroupTwo
```

此命令不會產生任何輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的帳單群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBillingGroup。

delete-ca-certificate

以下程式碼範例顯示如何使用 delete-ca-certificate。

AWS CLI

刪除 CA 憑證

下列delete-ca-certificate範例會刪除具有指定憑證 ID 的 CA 憑證。

```
aws iot delete-ca-certificate \
    --certificate-
id f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT API 參考中的 DeleteCACertificate。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCaCertificate。

delete-certificate

以下程式碼範例顯示如何使用 delete-certificate。

AWS CLI

刪除裝置憑證

下列delete-certificate範例會刪除具有指定 ID 的裝置憑證。

```
aws iot delete-certificate \
    --certificate-
id c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbee1428d216d54d53ac9
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT API 參考中的 DeleteCertificate。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCertificate。

delete-custom-metric

以下程式碼範例顯示如何使用 delete-custom-metric。

AWS CLI

刪除自訂指標

下列delete-custom-metric範例會刪除自訂指標。

```
aws iot delete-custom-metric \
    --metric-name batteryPercentage \
    --region us-east-1
```

輸出：

```
HTTP 200
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的自訂指標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCustomMetric。

delete-dimension

以下程式碼範例顯示如何使用 delete-dimension。

AWS CLI

刪除維度

下列delete-dimension範例會刪除稱為 的維度TopicFilterForAuthMessages。

```
aws iot delete-dimension \
    --name TopicFilterForAuthMessages
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Detect 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDimension。

delete-domain-configuration

以下程式碼範例顯示如何使用 delete-domain-configuration。

AWS CLI

刪除網域組態

下列delete-domain-configuration範例additionalDataDomain會從您的帳戶刪除名為 的網域組態 AWS 。

```
aws iot delete-domain-configuration \
    --domain-configuration-name "additionalDataDomain" \
    --domain-configuration-status "OK"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的可設定端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDomainConfiguration。

delete-dynamic-thing-group

以下程式碼範例顯示如何使用 delete-dynamic-thing-group。

AWS CLI

刪除動態物件群組

下列delete-dynamic-thing-group範例會刪除指定的動態物件群組。

```
aws iot delete-dynamic-thing-group \
    --thing-group-name "RoomTooWarm"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的動態物件群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDynamicThingGroup。

delete-job-execution

以下程式碼範例顯示如何使用 delete-job-execution。

AWS CLI

刪除任務執行

下列delete-job-execution範例會刪除裝置上指定任務的任務執行。使用 describe-job-execution取得執行編號。

```
aws iot delete-job-execution
    --job-id "example-job-02"
    --thing-name "MyRaspberryPi"
    --execution-number 1
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的建立和管理任務 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteJobExecution。

delete-job

以下程式碼範例顯示如何使用 delete-job。

AWS CLI

若要刪除工作

下列delete-job範例會刪除指定的任務。透過指定 --force選項，即使狀態為 ，也會刪除任務IN_PROGRESS。

```
aws iot delete-job \
    --job-id "example-job-04" \
    --force
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的建立和管理任務 (CLI)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteJob。

delete-mitigation-action

以下程式碼範例顯示如何使用 delete-mitigation-action。

AWS CLI

刪除緩解動作

下列delete-mitigation-action範例會刪除指定的緩解動作。

```
aws iot delete-mitigation-action \
```

```
    --action-name AddThingsToQuarantineGroup1Action
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 DeleteMitigationAction （緩解動作命令）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteMitigationAction。

delete-ota-update

以下程式碼範例顯示如何使用 delete-ota-update。

AWS CLI

刪除 OTA 更新

下列delete-ota-update範例會刪除指定的 OTA 更新。

```
aws iot delete-ota-update \
    --ota-update-id ota12345 \
    --delete-stream \
    --force-delete-aws-job
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT API 參考中的 DeleteOTAUpdate。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteOtaUpdate。

delete-policy-version

以下程式碼範例顯示如何使用 delete-policy-version。

AWS CLI

刪除政策的版本

下列delete-policy-version範例會從您的帳戶刪除指定政策的第 2 版 AWS 。

```
aws iot delete-policy-version \
    --policy-name UpdateDeviceCertPolicy \
    --policy-version-id 2
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 IoT 政策。 AWS IoT

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 DeletePolicyVersion。

delete-policy

以下程式碼範例顯示如何使用 delete-policy。

AWS CLI

刪除政策

下列delete-policy範例會從您的帳戶刪除指定的政策 AWS 。

```
aws iot delete-policy --policy-name UpdateDeviceCertPolicy
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 IoT 政策。 AWS IoT

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeletePolicy。

delete-provisioning-template-version

以下程式碼範例顯示如何使用 delete-provisioning-template-version。

AWS CLI

刪除佈建範本版本

下列delete-provisioning-template-version範例會刪除指定佈建範本的第 2 版。

```
aws iot delete-provisioning-template-version \
    --version-id 2 \
    --template-name "widget-template"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的 IoT 安全通道。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteProvisioningTemplateVersion。

delete-provisioning-template

以下程式碼範例顯示如何使用 delete-provisioning-template。

AWS CLI

刪除佈建範本

下列delete-provisioning-template範例會刪除指定的佈建範本。

```
aws iot delete-provisioning-template \
    --template-name widget-template
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的 IoT 安全通道。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteProvisioningTemplate。

delete-registration-code

以下程式碼範例顯示如何使用 delete-registration-code。

AWS CLI

刪除您的註冊 cod

下列delete-registration-code範例會刪除 an AWS IoT 帳戶特定的註冊碼。

```
aws iot delete-registration-code
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的使用您自己的憑證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRegistrationCode。

delete-role-alias

以下程式碼範例顯示如何使用 delete-role-alias。

AWS CLI

刪除 an AWS IoT 角色別名

下列delete-role-alias範例會刪除名為 的 AWS IoT 角色別名LightBulbRole。

```
aws iot delete-role-alias \
    --role-alias LightBulbRole
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的授權直接呼叫 AWS 服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRoleAlias。

delete-scheduled-audit

以下程式碼範例顯示如何使用 delete-scheduled-audit。

AWS CLI

刪除排定的稽核

下列delete-scheduled-audit範例會刪除名為 的 AWS IoT Device Defender 排程稽核AWSIoTDeviceDefenderDailyAudit。

```
aws iot delete-scheduled-audit \
    --scheduled-audit-name AWSIoTDeviceDefenderDailyAudit
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteScheduledAudit。

delete-security-profile

以下程式碼範例顯示如何使用 delete-security-profile。

AWS CLI

刪除安全性描述檔

下列delete-security-profile範例會刪除名為 的安全設定檔PossibleIssue。

```
aws iot delete-security-profile \
```

```
    --security-profile-name PossibleIssue
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Detect 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSecurityProfile。

delete-stream

以下程式碼範例顯示如何使用 delete-stream。

AWS CLI

刪除串流

下列delete-stream範例會刪除指定的串流。

```
aws iot delete-stream \
    --stream-id stream12345
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT API 參考中的 DeleteStream。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteStream。

delete-thing-group

以下程式碼範例顯示如何使用 delete-thing-group。

AWS CLI

刪除物件群組

下列delete-thing-group範例會刪除指定的物件群組。如果物件群組包含子物件群組，則無法刪除該物件群組。

```
aws iot delete-thing-group \
    --thing-group-name DefectiveBulbs
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteThingGroup。

delete-thing-type

以下程式碼範例顯示如何使用 delete-thing-type。

AWS CLI

範例 1：刪除物件類型

下列delete-thing-type範例會刪除已棄用的物件類型。

```
aws iot delete-thing-type \
    --thing-type-name "obsoleteThingType"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件類型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteThingType。

delete-thing

以下程式碼範例顯示如何使用 delete-thing。

AWS CLI

顯示物件的詳細資訊

下列delete-thing範例會從您 AWS 帳戶的 AWS IoT 登錄檔中刪除物件。

aws iot delete-thing --thing-name "FourthBulb"

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的如何使用登錄管理物件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteThing。

delete-topic-rule-destination

以下程式碼範例顯示如何使用 delete-topic-rule-destination。

AWS CLI

刪除主題規則目的地

下列delete-topic-rule-destination範例會刪除指定的主題規則目的地。

```
aws iot delete-topic-rule-destination \
    --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的刪除主題規則目的地。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTopicRuleDestination。

delete-topic-rule

以下程式碼範例顯示如何使用 delete-topic-rule。

AWS CLI

刪除規則

下列delete-topic-rule範例會刪除指定的規則。

```
aws iot delete-topic-rule \
    --rule-name "LowMoistureRule"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的刪除規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTopicRule。

delete-v2-logging-level

以下程式碼範例顯示如何使用 delete-v2-logging-level。

AWS CLI

刪除物件群組的記錄層級

下列delete-v2-logging-level範例會刪除指定物件群組的記錄層級。

```
aws iot delete-v2-logging-level \
    --target-type THING_GROUP \
    --target-name LightBulbs
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteV2LoggingLevel。

deprecate-thing-type

以下程式碼範例顯示如何使用 deprecate-thing-type。

AWS CLI

範例 1：棄用物件類型

下列deprecate-thing-type範例會棄用物件類型，讓使用者無法與其建立任何新物件的關聯。

```
aws iot deprecate-thing-type \
    --thing-type-name "obsoleteThingType"
```

此命令不會產生輸出。

範例 2：反轉物件類型的棄用

下列deprecate-thing-type範例會反轉物件類型的棄用，讓使用者可以再次將新物件與其建立關聯。

```
aws iot deprecate-thing-type \
    --thing-type-name "obsoleteThingType" \
    --undo-deprecate
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件類型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeprecateThingType。

describe-account-audit-configuration

以下程式碼範例顯示如何使用 describe-account-audit-configuration。

AWS CLI

檢視目前的稽核組態設定

下列describe-account-audit-configuration範例列出 IoT AWS IoT Device Defender 稽核組態的目前設定。

```
aws iot describe-account-audit-configuration
```

輸出：

```
{
    "roleArn": "arn:aws:iam::123456789012:role/service-role/
AWSIoTDeviceDefenderAudit_1551201085996",
    "auditNotificationTargetConfigurations": {
        "SNS": {
            "targetArn": "arn:aws:sns:us-west-2:123456789012:ddaudits",
            "roleArn": "arn:aws:iam::123456789012:role/service-role/
AWSIoTDeviceDefenderAudit",
            "enabled": true
        }
    },
    "auditCheckConfigurations": {
        "AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK": {
            "enabled": true
        },
        "CA_CERTIFICATE_EXPIRING_CHECK": {
            "enabled": true
        },
        "CONFLICTING_CLIENT_IDS_CHECK": {
            "enabled": true
        },
        "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
            "enabled": true
        },
        "DEVICE_CERTIFICATE_SHARED_CHECK": {
            "enabled": true
        },
        "IOT_POLICY_OVERLY_PERMISSIVE_CHECK": {
            "enabled": true
        },
        "LOGGING_DISABLED_CHECK": {
            "enabled": true
```

```
        },
        "REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK": {
            "enabled": true
        },
        "REVOKED_DEVICE_CERTIFICATE_STILL_ACTIVE_CHECK": {
            "enabled": true
        },
        "UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK": {
            "enabled": true
        }
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAccountAuditConfiguration。

describe-audit-finding

以下程式碼範例顯示如何使用 describe-audit-finding。

AWS CLI

列出稽核問題清單的詳細資訊

下列describe-audit-finding範例列出指定 AWS IoT Device Defender 稽核調查結果的詳細資訊。稽核可以產生多個問題清單。使用 list-audit-findings命令從稽核取得問題清單，以取得 findingId。

```
aws iot describe-audit-finding \
    --finding-id "ef4826b8-e55a-44b9-b460-5c485355371b"
```

輸出：

```
{
    "finding": {
        "findingId": "ef4826b8-e55a-44b9-b460-5c485355371b",
        "taskId": "873ed69c74a9ec8fa9b8e88e9abc4661",
        "checkName": "IOT_POLICY_OVERLY_PERMISSIVE_CHECK",
        "taskStartTime": 1576012045.745,
        "findingTime": 1576012046.168,
```

```
        "severity": "CRITICAL",
        "nonCompliantResource": {
            "resourceType": "IOT_POLICY",
            "resourceIdentifier": {
                "policyVersionIdentifier": {
                    "policyName": "smp-ggrass-group_Core-policy",
                    "policyVersionId": "1"
                }
            }
         },
        "reasonForNonCompliance": "Policy allows broad access to IoT data plane
 actions: [iot:Subscribe, iot:Connect, iot:GetThingShadow, iot:DeleteThingShadow,
 iot:UpdateThingShadow, iot:Publish].",
        "reasonForNonComplianceCode":
 "ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS"
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的檢查稽核結果（稽核命令）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAuditFinding。

describe-audit-mitigation-actions-task

以下程式碼範例顯示如何使用 describe-audit-mitigation-actions-task。

AWS CLI

顯示稽核緩解動作任務的詳細資訊

下列describe-audit-mitigation-actions-task範例顯示指定任務的詳細資訊，其中ResetPolicyVersionAction已套用至問題清單。結果包括任務開始和結束的時間、目標調查結果數量（和結果），以及在此任務中套用的動作定義。

```
aws iot describe-audit-mitigation-actions-task \
    --task-id ResetPolicyTask01
```

輸出：

```
{
    "taskStatus": "COMPLETED",
    "startTime": "2019-12-10T15:13:19.457000-08:00",
```

```
    "endTime": "2019-12-10T15:13:19.947000-08:00",
    "taskStatistics": {
        "IOT_POLICY_OVERLY_PERMISSIVE_CHECK": {
            "totalFindingsCount": 1,
            "failedFindingsCount": 0,
            "succeededFindingsCount": 1,
            "skippedFindingsCount": 0,
            "canceledFindingsCount": 0
        }
    },
    "target": {
        "findingIds": [
            "ef4826b8-e55a-44b9-b460-5c485355371b"
        ]
    },
    "auditCheckToActionsMapping": {
        "IOT_POLICY_OVERLY_PERMISSIVE_CHECK": [
            "ResetPolicyVersionAction"
        ]
    },
    "actionsDefinition": [
        {
            "name": "ResetPolicyVersionAction",
            "id": "1ea0b415-bef1-4a01-bd13-72fb63c59afb",
            "roleArn": "arn:aws:iam::123456789012:role/service-role/
ReplacePolicyVersionRole",
            "actionParams": {
                "replaceDefaultPolicyVersionParams": {
                    "templateName": "BLANK_POLICY"
                }
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 DescribeAuditMitigationActionsTask （緩解動作命令）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAuditMitigationActionsTask。

describe-audit-suppression

以下程式碼範例顯示如何使用 describe-audit-suppression。

AWS CLI

取得稽核問題清單禁止的詳細資訊

下列describe-audit-suppression範例列出有關稽核問題清單禁止的詳細資訊。

```
aws iot describe-audit-task \
    --task-id "787ed873b69cb4d6cdbae6ddd06996c5"
```

輸出：

```
{
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK",
    "taskStartTime": 1596168096.157,
    "taskStatistics": {
        "totalChecks": 1,
        "inProgressChecks": 0,
        "waitingForDataCollectionChecks": 0,
        "compliantChecks": 0,
        "nonCompliantChecks": 1,
        "failedChecks": 0,
        "canceledChecks": 0
    },
    "scheduledAuditName": "AWSIoTDeviceDefenderDailyAudit",
    "auditDetails": {
        "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
            "checkRunStatus": "COMPLETED_NON_COMPLIANT",
            "checkCompliant": false,
            "totalResourcesCount": 195,
            "nonCompliantResourcesCount": 2
        }
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核調查結果抑制。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAuditSuppression。

describe-audit-task

以下程式碼範例顯示如何使用 describe-audit-task。

AWS CLI

取得稽核執行個體的相關資訊

下列`describe-audit-task`範例取得 an AWS IoT Device Defender 稽核執行個體的相關資訊。如果稽核完成，則執行的摘要統計資料會包含在結果中。

```
aws iot describe-audit-task \
    --task-id a3aea009955e501a31b764abe1bebd3d
```

輸出：

```
{
   "taskStatus": "COMPLETED",
   "taskType": "ON_DEMAND_AUDIT_TASK",
   "taskStartTime": 1560356923.434,
   "taskStatistics": {
       "totalChecks": 3,
       "inProgressChecks": 0,
       "waitingForDataCollectionChecks": 0,
       "compliantChecks": 3,
       "nonCompliantChecks": 0,
       "failedChecks": 0,
       "canceledChecks": 0
   },
   "auditDetails": {
       "CA_CERTIFICATE_EXPIRING_CHECK": {
           "checkRunStatus": "COMPLETED_COMPLIANT",
           "checkCompliant": true,
           "totalResourcesCount": 0,
           "nonCompliantResourcesCount": 0
       },
       "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
           "checkRunStatus": "COMPLETED_COMPLIANT",
           "checkCompliant": true,
           "totalResourcesCount": 6,
           "nonCompliantResourcesCount": 0
       },
       "REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK": {
           "checkRunStatus": "COMPLETED_COMPLIANT",
           "checkCompliant": true,
           "totalResourcesCount": 0,
           "nonCompliantResourcesCount": 0
```

```
        }
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAuditTask。

describe-authorizer

以下程式碼範例顯示如何使用 describe-authorizer。

AWS CLI

取得自訂授權方的相關資訊

下列describe-authorizer範例顯示指定自訂授權方的詳細資訊。

```
aws iot describe-authorizer \
    --authorizer-name CustomAuthorizer
```

輸出：

```
{
    "authorizerDescription": {
        "authorizerName": "CustomAuthorizer",
        "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer",
        "authorizerFunctionArn": "arn:aws:lambda:us-
west-2:123456789012:function:CustomAuthorizerFunction",
        "tokenKeyName": "MyAuthToken",
        "tokenSigningPublicKeys": {
            "FIRST_KEY": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA1uJOB4lQPgG/lM6ZfIwo
\nZ+7ENxAio9q6QD4FFqjGZsvjtYwjoe1RKK0U8Eq9xb5O3kRSmyIwTzwzm/f4Gf0Y
\nZUloJ+t3PUUwHrmbYTAgTrCUgRFygjfgVwGCPs5ZAX4Eyqt5cr+AIHIiUDbxSa7p
\nzwOBKPeic0asNJpqT8PkBbRaKyleJh5oo81NDHHmVtbBm5A5YiJjqYXLaVAowKzZ\n
+GqsNvAQ9Jy1wI2VrEa1OfL8flDB/BJLm7zjpfPOHDJQgID0XnZwAlNnZcOhCwIx\n50g2LW2Oy9R/
dmqtDmJiVP97Z4GykxPvwlYHrUXY0iW1R3AR/Ac1NhCTGZMwVDB1\nlQIDAQAB\n-----END PUBLIC
 KEY-----"
        },
        "status": "ACTIVE",
```

```
        "creationDate": 1571245658.069,
        "lastModifiedDate": 1571245658.069
    }
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 DescribeAuthorizer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAuthorizer。

describe-billing-group

以下程式碼範例顯示如何使用 describe-billing-group。

AWS CLI

取得帳單群組的相關資訊

下列describe-billing-group範例會取得指定帳單群組的資訊。

```
aws iot describe-billing-group --billing-group-name GroupOne
```

輸出：

```
{
    "billingGroupName": "GroupOne",
    "billingGroupId": "103de383-114b-4f51-8266-18f209ef5562",
    "billingGroupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne",
    "version": 1,
    "billingGroupProperties": {},
    "billingGroupMetadata": {
        "creationDate": 1560199355.378
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的帳單群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeBillingGroup。

describe-ca-certificate

以下程式碼範例顯示如何使用 describe-ca-certificate。

AWS CLI

取得 CA 憑證的詳細資訊

下列describe-ca-certificate範例顯示指定 CA 憑證的詳細資訊。

```
aws iot describe-ca-certificate \
    --certificate-
id f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467
```

輸出：

```
{
    "certificateDescription": {
        "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467",
        "certificateId":
 "f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467",
        "status": "INACTIVE",
        "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIICzzCCAbegEXAMPLEJANVEPWXl8taPMA0GCSqGSIb3DQEBBQUAMB4xCzAJBgNV
\nBAYTAlVTMQ8wDQYDVQQKDAZBbWF6b24wHhcNMTkwOTI0MjEzMTE1WhcNMjkwOTIx
\nMjEzMTE1WjAeMQswCQYDVQQGEwJVUzEPMA0GA1UECgwGQW1hem9uMIIBIjANBgkq
\nhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzd3R3ioalCS0MhFWfBrVGR036EK07UAf
\nVdz9EXAMPLE1VczICbADnATK522kEIB51/18VzlFtAhQL5V5eybXKnB7QebNer5m
\n4Yibx7shR5oqNzFsrXWxuugN5+w5gEfqNMawOjhF4LsculKG49yuqjcDU19/13ua
\n3B2gxs1Pe7TiWWvUskzxnbO1F2WCshbEJvqY8fIWtGYCjTeJAgQ9hvZx/69XhKen
\nwV9LJwOQxrsUS0Ty8IHwbB8fRy72VM3u7fJoaU+nO4jD5cqaoEPtzoeFUEXAMPLE
\nyVAJpqHwgbYbcUfn7V+AB6yh1+0Fa1rEQGuZDPGyJslxwr5vh8nRewIDAQABoxAw
\nDjAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBQUAA4IBAQA+3a5CV3IJgOnd0AgI
\nBgVMtmYzTvqAngx26aG9/spvCjXckh2SBF+EcBlCFwH1yakwjJL1dR4yarnrfxgI
\nEqP4AOYVimAVoQ5FBwnloHe16+3qtDiblU9DeXBUCtS55EcfrEXAMPLEYtXdqU5C
\nU9ia4KAjV0dxW1+EFYMwX5eGeb0gDTNHBylV6B/fOSZiQAwDYp4x3B+gAP+a/bWB
\nu1umOqtBdWe6L6/83L+JhaTByqV25iVJ4c/UZUnG8926wUlDM9zQvEXuEVvzZ7+m\n4PSNqst/
nVOvnLpoG4e0WgcJgANuB33CSWtjWSuYsbhmqQRknGhREXAMPLEZT4fm\nfo0e\n-----END
 CERTIFICATE-----\n",
        "ownedBy": "123456789012",
        "creationDate": 1569365372.053,
        "autoRegistrationStatus": "DISABLE",
        "lastModifiedDate": 1569365372.053,
        "customerVersion": 1,
        "generationId": "c5c2eb95-140b-4f49-9393-6aaac85b2a90",
        "validity": {
```

```
            "notBefore": 1569360675.0,
            "notAfter": 1884720675.0
        }
    }
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 DescribeCACertificate。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCaCertificate。

describe-certificate

以下程式碼範例顯示如何使用 describe-certificate。

AWS CLI

取得憑證的相關資訊

下列describe-certificate範例顯示指定憑證的詳細資訊。

```
aws iot describe-certificate \
    --certificate-
id "4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e"
```

輸出：

```
{
    "certificateDescription": {
        "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
        "certificateId":
 "4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
        "status": "ACTIVE",
        "certificatePem": "-----BEGIN CERTIFICATE-----
MIICiTEXAMPLEQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBEXAMPLEMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDEXAMPLElMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5EXAMPLEcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNEXAMPLEdBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBEXAMPLEz
b2xEXAMPLEYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8EXAMPLEZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
```

```
21uUSfwfEvySWtC2XADZ4nB+BLYEXAMPLEpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7EXAMPLEGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFEXAMPLEAtCu4
nUhVVxYUnEXAMPLE8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GEXAMPLEl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
        "ownedBy": "123456789012",
        "creationDate": 1541022751.983,
        "lastModifiedDate": 1541022751.983,
        "customerVersion": 1,
        "transferData": {},
        "generationId": "6974fbed-2e61-4114-bc5e-4204cc79b045",
        "validity": {
            "notBefore": 1541022631.0,
            "notAfter": 2524607999.0
        }
    }
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 DescribeCertificate。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCertificate。

describe-custom-metric

以下程式碼範例顯示如何使用 `describe-custom-metric`。

AWS CLI

取得 Device Defender 自訂指標的相關資訊

下列`describe-custom-metric`範例取得名為 之自訂指標的相關資訊`myCustomMetric`。

```
aws iot describe-custom-metric \
    --metric-name myCustomMetric
```

輸出：

```
{
    "metricName": "myCustomMetric",
    "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/myCustomMetric",
```

```
    "metricType": "number",
    "displayName": "My custom metric",
    "creationDate": 2020-11-17T23:02:12.879000-09:00,
    "lastModifiedDate": 2020-11-17T23:02:12.879000-09:00
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的自訂指標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCustomMetric。

describe-default-authorizer

以下程式碼範例顯示如何使用 describe-default-authorizer。

AWS CLI

取得預設自訂授權方的相關資訊

下列describe-default-authorizer範例顯示預設自訂授權方的詳細資訊。

```
aws iot describe-default-authorizer
```

輸出：

```
{
    "authorizerName": "CustomAuthorizer",
    "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer"
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 DescribeDefaultAuthorizer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDefaultAuthorizer。

describe-dimension

以下程式碼範例顯示如何使用 describe-dimension。

AWS CLI

取得維度的相關資訊

下列describe-dimension範例取得名為 之維度的相關資訊TopicFilterForAuthMessages。

```
aws iot describe-dimension \
    --name TopicFilterForAuthMessages
```

輸出：

```
{
    "name": "TopicFilterForAuthMessages",
    "arn": "arn:aws:iot:eu-west-2:123456789012:dimension/
TopicFilterForAuthMessages",
    "type": "TOPIC_FILTER",
    "stringValues": [
        "device/+/auth"
    ],
    "creationDate": 1578620223.255,
    "lastModifiedDate": 1578620223.255
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Detect 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDimension。

describe-domain-configuration

以下程式碼範例顯示如何使用 describe-domain-configuration。

AWS CLI

描述網域組態

下列describe-domain-configuration範例顯示指定網域組態的詳細資訊。

```
aws iot describe-domain-configuration \
    --domain-configuration-name "additionalDataDomain"
```

輸出：

```
{
```

```
    "domainConfigurationName": "additionalDataDomain",
    "domainConfigurationArn": "arn:aws:iot:us-
east-1:758EXAMPLE143:domainconfiguration/additionalDataDomain/norpw",
    "domainName": "d055exampleed74y71zfd-ats.beta.us-east-1.iot.amazonaws.com",
    "serverCertificates": [],
    "domainConfigurationStatus": "ENABLED",
    "serviceType": "DATA",
    "domainType": "AWS_MANAGED",
    "lastStatusChangeDate": 1601923783.774
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的可設定端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDomainConfiguration。

`describe-endpoint`

以下程式碼範例顯示如何使用 `describe-endpoint`。

AWS CLI

範例 1：取得您目前的 AWS 端點

下列`describe-endpoint`範例會擷取套用所有命令的預設 AWS 端點。

```
aws iot describe-endpoint
```

輸出：

```
{
    "endpointAddress": "abc123defghijk.iot.us-west-2.amazonaws.com"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 DescribeEndpoint。

範例 2：取得您的 ATS 端點

下列`describe-endpoint`範例會擷取 Amazon Trust Services (ATS) 端點。

```
aws iot describe-endpoint \
```

```
    --endpoint-type iot:Data-ATS
```

輸出：

```
{
    "endpointAddress": "abc123defghijk-ats.iot.us-west-2.amazonaws.com"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 X.509 Certificates and AWS IoT。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEndpoint。

describe-event-configurations

以下程式碼範例顯示如何使用 describe-event-configurations。

AWS CLI

顯示要發佈的事件類型

下列describe-event-configurations範例列出組態，控制在新增、更新或刪除某個項目時產生哪些事件。

```
aws iot describe-event-configurations
```

輸出：

```
{
    "eventConfigurations": {
        "CA_CERTIFICATE": {
            "Enabled": false
        },
        "CERTIFICATE": {
            "Enabled": false
        },
        "JOB": {
            "Enabled": false
        },
        "JOB_EXECUTION": {
            "Enabled": false
        },
```

```
        "POLICY": {
            "Enabled": false
        },
        "THING": {
            "Enabled": false
        },
        "THING_GROUP": {
            "Enabled": false
        },
        "THING_GROUP_HIERARCHY": {
            "Enabled": false
        },
        "THING_GROUP_MEMBERSHIP": {
            "Enabled": false
        },
        "THING_TYPE": {
            "Enabled": false
        },
        "THING_TYPE_ASSOCIATION": {
            "Enabled": false
        }
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的事件訊息。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEventConfigurations。

describe-index

以下程式碼範例顯示如何使用 `describe-index`。

AWS CLI

擷取物件索引的目前狀態

下列`describe-index`範例會擷取物件索引的目前狀態。

```
aws iot describe-index \
    --index-name "AWS_Things"
```

輸出：

```
{
    "indexName": "AWS_Things",
    "indexStatus": "ACTIVE",
    "schema": "REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的管理物件索引。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeIndex。

describe-job-execution

以下程式碼範例顯示如何使用 describe-job-execution。

AWS CLI

取得裝置上的任務執行詳細資訊

下列describe-job-execution範例會取得指定任務的執行詳細資訊。

```
aws iot describe-job-execution \
    --job-id "example-job-01" \
    --thing-name "MyRaspberryPi"
```

輸出：

```
{
    "execution": {
        "jobId": "example-job-01",
        "status": "QUEUED",
        "statusDetails": {},
        "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyRaspberryPi",
        "queuedAt": 1560787023.636,
        "lastUpdatedAt": 1560787023.636,
        "executionNumber": 1,
        "versionNumber": 1
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的建立和管理任務 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeJobExecution。

describe-job

以下程式碼範例顯示如何使用 describe-job。

AWS CLI

取得任務的詳細狀態

下列describe-job範例會取得 ID 為 之任務的詳細狀態example-job-01。

```
aws iot describe-job \
    --job-id "example-job-01"
```

輸出：

```
{
    "job": {
        "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",
        "jobId": "example-job-01",
        "targetSelection": "SNAPSHOT",
        "status": "IN_PROGRESS",
        "targets": [
            "arn:aws:iot:us-west-2:123456789012:thing/MyRaspberryPi"
        ],
        "description": "example job test",
        "presignedUrlConfig": {},
        "jobExecutionsRolloutConfig": {},
        "createdAt": 1560787022.733,
        "lastUpdatedAt": 1560787026.294,
        "jobProcessDetails": {
            "numberOfCanceledThings": 0,
            "numberOfSucceededThings": 0,
            "numberOfFailedThings": 0,
            "numberOfRejectedThings": 0,
            "numberOfQueuedThings": 1,
            "numberOfInProgressThings": 0,
            "numberOfRemovedThings": 0,
            "numberOfTimedOutThings": 0
        },
        "timeoutConfig": {}
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的建立和管理任務 (CLI)。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeVpcs。

describe-mitigation-action

以下程式碼範例顯示如何使用 describe-mitigation-action。

AWS CLI

檢視已定義緩解動作的詳細資訊

下列describe-mitigation-action範例顯示指定緩解動作的詳細資訊。

```
aws iot describe-mitigation-action \
    --action-name AddThingsToQuarantineGroupAction
```

輸出：

```
{
    "actionName": "AddThingsToQuarantineGroupAction",
    "actionType": "ADD_THINGS_TO_THING_GROUP",
    "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
AddThingsToQuarantineGroupAction",
    "actionId": "2fd2726d-98e1-4abf-b10f-09465ccd6bfa",
    "roleArn": "arn:aws:iam::123456789012:role/service-role/
MoveThingsToQuarantineGroupRole",
    "actionParams": {
        "addThingsToThingGroupParams": {
            "thingGroupNames": [
                "QuarantineGroup1"
            ],
            "overrideDynamicGroups": true
        }
    },
    "creationDate": "2019-12-10T11:09:35.999000-08:00",
    "lastModifiedDate": "2019-12-10T11:09:35.999000-08:00"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 DescribeMitigationAction （緩解動作命令）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeMitigationAction。

describe-provisioning-template-version

以下程式碼範例顯示如何使用 describe-provisioning-template-version。

AWS CLI

描述佈建範本版本

下列describe-provisioning-template-version範例說明佈建範本版本。

```
aws iot describe-provisioning-template-version \
    --template-name MyTestProvisioningTemplate \
    --version-id 1
```

輸出：

```
{
    "versionId": 1,
    "creationDate": 1589308310.574,
    "templateBody": "{
        \"Parameters\":{
            \"SerialNumber\":{
                \"Type\":\"String\"
            },
            \"AWS::IoT::Certificate::Id\":{
                \"Type\":\"String\"
            }
        },
        \"Resources\":{
            \"certificate\":{
                \"Properties\":{
                    \"CertificateId\":{
                        \"Ref\":\"AWS::IoT::Certificate::Id\"
                    },
                    \"Status\":\"Active\"
                },
                \"Type\":\"AWS::IoT::Certificate\"
            },
            \"policy\":{
                \"Properties\":{
                    \"PolicyName\":\"MyIotPolicy\"
                },
                \"Type\":\"AWS::IoT::Policy\"
            },
```

```
            \"thing\":{
                \"OverrideSettings\":{
                    \"AttributePayload\":\"MERGE\",
                    \"ThingGroups\":\"DO_NOTHING\",
                    \"ThingTypeName\":\"REPLACE\"
                },
                \"Properties\":{
                    \"AttributePayload\":{},
                    \"ThingGroups\":[],
                    \"ThingName\":{
                        \"Fn::Join\":[
                            \"\",
                            [
                                \"DemoGroup_\",
                                {\"Ref\":\"SerialNumber\"}
                            ]
                        ]
                    },
                    \"ThingTypeName\":\"VirtualThings\"
                },
                \"Type\":\"AWS::IoT::Thing\"
            }
        }
    }",
    "isDefaultVersion": true
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的使用機群佈建佈建沒有裝置憑證的裝置。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeProvisioningTemplateVersion。

describe-provisioning-template

以下程式碼範例顯示如何使用 describe-provisioning-template。

AWS CLI

描述佈建範本

下列describe-provisioning-template範例說明佈建範本。

```
aws iot describe-provisioning-template \
```

```
    --template-name MyTestProvisioningTemplate
```

輸出：

```
{
    "templateArn": "arn:aws:iot:us-west-2:57EXAMPLE833:provisioningtemplate/
MyTestProvisioningTemplate",
    "templateName": "MyTestProvisioningTemplate",
    "creationDate": 1589308310.574,
    "lastModifiedDate": 1589308345.539,
    "defaultVersionId": 1,
    "templateBody": "{
        \"Parameters\":{
            \"SerialNumber\":{
                \"Type\":\"String\"
            },
            \"AWS::IoT::Certificate::Id\":{
                \"Type\":\"String\"
            }
        },
        \"Resources\":{
            \"certificate\":{
                \"Properties\":{
                    \"CertificateId\":{
                        \"Ref\":\"AWS::IoT::Certificate::Id\"
                    },
                    \"Status\":\"Active\"
                },
                \"Type\":\"AWS::IoT::Certificate\"
            },
            \"policy\":{
                \"Properties\":{
                    \"PolicyName\":\"MyIotPolicy\"
                },
                \"Type\":\"AWS::IoT::Policy\"
            },
            \"thing\":{
                \"OverrideSettings\":{
                    \"AttributePayload\":\"MERGE\",
                    \"ThingGroups\":\"DO_NOTHING\",
                    \"ThingTypeName\":\"REPLACE\"
                },
                \"Properties\":{
```

```
                    \"AttributePayload\":{},
                    \"ThingGroups\":[],
                    \"ThingName\":{
                        \"Fn::Join\":[
                            \"\",
                            [
                                \"DemoGroup_\",
                                {\"Ref\":\"SerialNumber\"}
                            ]
                        ]
                    },
                    \"ThingTypeName\":\"VirtualThings\"
                },
                \"Type\":\"AWS::IoT::Thing\"
            }
        }
    }",
    "enabled": true,
    "provisioningRoleArn": "arn:aws:iam::571032923833:role/service-role/IoT_access"
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的使用機群佈建佈建沒有裝置憑證的裝置。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeProvisioningTemplate。

describe-role-alias

以下程式碼範例顯示如何使用 `describe-role-alias`。

AWS CLI

取得 an AWS IoT 角色別名的相關資訊

下列`describe-role-alias`範例顯示指定角色別名的詳細資訊。

```
aws iot describe-role-alias \
    --role-alias LightBulbRole
```

輸出：

```
{
    "roleAliasDescription": {
        "roleAlias": "LightBulbRole",
```

```
        "roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/
LightBulbRole",
        "roleArn": "arn:aws:iam::123456789012:role/light_bulb_role_001",
        "owner": "123456789012",
        "credentialDurationSeconds": 3600,
        "creationDate": 1570558643.221,
        "lastModifiedDate": 1570558643.221
    }
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 DescribeRoleAlias。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRoleAlias。

describe-scheduled-audit

以下程式碼範例顯示如何使用 describe-scheduled-audit。

AWS CLI

取得排程稽核的相關資訊

下列describe-scheduled-audit範例取得名為 之 AWS IOT Device Defender 排程稽核的詳細資訊AWSIoTDeviceDefenderDailyAudit。

```
aws iot describe-scheduled-audit \
    --scheduled-audit-name AWSIoTDeviceDefenderDailyAudit
```

輸出：

```
{
    "frequency": "DAILY",
    "targetCheckNames": [
        "AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK",
        "CONFLICTING_CLIENT_IDS_CHECK",
        "DEVICE_CERTIFICATE_SHARED_CHECK",
        "IOT_POLICY_OVERLY_PERMISSIVE_CHECK",
        "REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK",
        "UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK"
    ],
    "scheduledAuditName": "AWSIoTDeviceDefenderDailyAudit",
    "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
AWSIoTDeviceDefenderDailyAudit"
```

```
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeScheduledAudit。

describe-security-profile

以下程式碼範例顯示如何使用 describe-security-profile。

AWS CLI

取得安全性設定檔的相關資訊

下列describe-security-profile範例取得名為 的 AWS IoT Device Defender 安全性描述檔相關資訊 PossibleIssue.

```
aws iot describe-security-profile \
    --security-profile-name PossibleIssue
```

輸出：

```
{
    "securityProfileName": "PossibleIssue",
    "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/
PossibleIssue",
    "securityProfileDescription": "check to see if authorization fails 10 times in 5
 minutes or if cellular bandwidth exceeds 128",
    "behaviors": [
        {
            "name": "CellularBandwidth",
            "metric": "aws:message-byte-size",
            "criteria": {
                "comparisonOperator": "greater-than",
                "value": {
                    "count": 128
                },
                "consecutiveDatapointsToAlarm": 1,
                "consecutiveDatapointsToClear": 1
            }
        },
        {
```

```
            "name": "Authorization",
            "metric": "aws:num-authorization-failures",
            "criteria": {
                "comparisonOperator": "greater-than",
                "value": {
                    "count": 10
                },
                "durationSeconds": 300,
                "consecutiveDatapointsToAlarm": 1,
                "consecutiveDatapointsToClear": 1
            }
        }
    ],
    "version": 1,
    "creationDate": 1560278102.528,
    "lastModifiedDate": 1560278102.528
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Detect 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSecurityProfile。

describe-stream

以下程式碼範例顯示如何使用 describe-stream。

AWS CLI

取得串流的相關資訊

下列describe-stream範例顯示指定串流的詳細資訊。

```
aws iot describe-stream \
    --stream-id stream12345
```

輸出：

```
{
    "streamInfo": {
        "streamId": "stream12345",
        "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream12345",
        "streamVersion": 1,
        "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",
```

```
        "files": [
            {
                "fileId": "123",
                "s3Location": {
                    "bucket":"codesign-ota-bucket",
                    "key":"48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"
                }
            }
        ],
        "createdAt": 1557863215.995,
        "lastUpdatedAt": 1557863215.995,
        "roleArn": "arn:aws:iam:123456789012:role/service-role/my_ota_stream_role"
    }
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 DescribeStream。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStream。

describe-thing-group

以下程式碼範例顯示如何使用 describe-thing-group。

AWS CLI

取得物件群組的相關資訊

下列describe-thing-group範例取得名為 之物件群組的相關資訊HalogenBulbs。

```
aws iot describe-thing-group \
    --thing-group-name HalogenBulbs
```

輸出：

```
{
    "thingGroupName": "HalogenBulbs",
    "thingGroupId": "f4ec6b84-b42b-499d-9ce1-4dbd4d4f6f6e",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/HalogenBulbs",
    "version": 1,
    "thingGroupProperties": {},
    "thingGroupMetadata": {
        "parentGroupName": "LightBulbs",
        "rootToParentThingGroups": [
```

```
            {
                "groupName": "LightBulbs",
                "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
LightBulbs"
            }
        ],
        "creationDate": 1559927609.897
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeThingGroup。

describe-thing-type

以下程式碼範例顯示如何使用 describe-thing-type。

AWS CLI

取得物件類型的相關資訊

下列describe-thing-type範例顯示您 AWS 帳戶中定義之指定物件類型的相關資訊。

```
aws iot describe-thing-type \
    --thing-type-name "LightBulb"
```

輸出：

```
{
    "thingTypeName": "LightBulb",
    "thingTypeId": "ce3573b0-0a3c-45a7-ac93-4e0ce14cd190",
    "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb",
    "thingTypeProperties": {
        "thingTypeDescription": "light bulb type",
        "searchableAttributes": [
            "model",
            "wattage"
        ]
    },
    "thingTypeMetadata": {
        "deprecated": false,
        "creationDate": 1559772562.498
```

```
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件類型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeThingType。

describe-thing

以下程式碼範例顯示如何使用 describe-thing。

AWS CLI

顯示物件的詳細資訊

下列describe-thing範例顯示 AWS 您的帳戶的 AWS IoT 登錄檔中定義的物件 （ 裝置 ） 的相關資訊。

aws iot describe-thing --thing-name "MyLightBulb"

輸出：

```
{
    "defaultClientId": "MyLightBulb",
    "thingName": "MyLightBulb",
    "thingId": "40da2e73-c6af-406e-b415-15acae538797",
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
    "thingTypeName": "LightBulb",
    "attributes": {
        "model": "123",
        "wattage": "75"
    },
    "version": 1
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的如何使用登錄管理物件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeThing。

detach-policy

以下程式碼範例顯示如何使用 detach-policy。

AWS CLI

範例 1：從物件群組分離 an AWS IoT 政策

下列detach-policy範例會從物件群組分離指定的政策，並依延伸與該群組和群組的任何子群組中的所有物件分離。

```
aws iot detach-policy \
    --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \
    --policy-name "MyFirstGroup_Core-policy"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件群組。

範例 2：從裝置憑證分離 an AWS IoT 政策

下列detach-policy範例會將 TemperatureSensorPolicy 政策從 ARN 識別的裝置憑證分離。

```
aws iot detach-policy \
    --policy-name TemperatureSensorPolicy \
    --target arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetachPolicy。

detach-security-profile

以下程式碼範例顯示如何使用 detach-security-profile。

AWS CLI

取消安全性設定檔與目標的關聯

下列detach-security-profile範例會移除名為 的 AWS IoT Device Defender 安全性描述檔Testprofile與所有已註冊物件目標之間的關聯。

```
aws iot detach-security-profile \
    --security-profile-name Testprofile \
```

```
    --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/
registered-things"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Detect 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetachSecurityProfile。

detach-thing-principal

以下程式碼範例顯示如何使用 detach-thing-principal。

AWS CLI

從物件分離憑證/委託人

下列detach-thing-principal範例會從指定的物件移除代表委託人的憑證。

```
aws iot detach-thing-principal \
    --thing-name "MyLightBulb" \
    --principal "arn:aws:iot:us-
west-2:123456789012:cert/604c48437a57b7d5fc5d137c5be75011c6ee67c9a6943683a1acb4b1626bac36"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的如何使用登錄管理物件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetachThingPrincipal。

disable-topic-rule

以下程式碼範例顯示如何使用 disable-topic-rule。

AWS CLI

停用主題規則

下列disable-topic-rule範例會停用指定的主題規則。

```
aws iot disable-topic-rule \
```

```
    --rule-name "MyPlantPiMoistureAlertRule"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的檢視您的規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableTopicRule。

enable-topic-rule

以下程式碼範例顯示如何使用 enable-topic-rule。

AWS CLI

啟用主題規則

下列enable-topic-rule範例會啟用 （ 或重新啟用 ） 指定的主題規則。

```
aws iot enable-topic-rule \
    --rule-name "MyPlantPiMoistureAlertRule"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的檢視您的規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableTopicRule。

get-behavior-model-training-summaries

以下程式碼範例顯示如何使用 get-behavior-model-training-summaries。

AWS CLI

列出 Device Defender 的 ML Detect Security Profile 訓練模型狀態

下列get-behavior-model-training-summaries範例列出所選安全性設定檔中已設定行為的模型訓練狀態。對於每個行為，會列出收集的資料點名稱、模型狀態和百分比。

```
aws iot get-behavior-model-training-summaries \
    --security-profile-name MySecuirtyProfileName
```

輸出：

```
{
    "summaries": [
        {
            "securityProfileName": "MySecuirtyProfileName",
            "behaviorName": "Messages_sent_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        },
        {
            "securityProfileName": "MySecuirtyProfileName",
            "behaviorName": "Messages_received_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        },
        {
            "securityProfileName": "MySecuirtyProfileName",
            "behaviorName": "Authorization_failures_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        },
        {
            "securityProfileName": "MySecuirtyProfileName",
            "behaviorName": "Message_size_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        },
        {
            "securityProfileName": "MySecuirtyProfileName",
            "behaviorName": "Connection_attempts_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        },
        {
            "securityProfileName": "MySPNoALerts",
            "behaviorName": "Disconnects_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        }
    ]
}
```


如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 GetBehaviorModelTrainingSummaries（偵測命令）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBehaviorModelTrainingSummaries。

get-cardinality

以下程式碼範例顯示如何使用 get-cardinality。

AWS CLI

傳回符合查詢之唯一值的近似計數

您可以使用下列設定指令碼來建立 10 個代表 10 個溫度感應器的物件。每個新物件都有 3 個屬性。

```
# Bash script. If in other shells, type `bash` before running
Temperatures=(70 71 72 73 74 75 47 97 98 99)
Racks=(Rack1 Rack1 Rack2 Rack2 Rack3 Rack4 Rack5 Rack6 Rack6 Rack6)
IsNormal=(true true true true true true false false false false)
for ((i=0; i<10 ; i++))
do
  thing=$(aws iot create-thing --thing-name "TempSensor$i" --attribute-payload
 attributes="{temperature=${Temperatures[i]},rackId=${Racks[i]},stateNormal=
${IsNormal[i]}}")
  aws iot describe-thing --thing-name "TempSensor$i"
done
```

設定指令碼的範例輸出：

```
{
    "version": 1,
    "thingName": "TempSensor0",
    "defaultClientId": "TempSensor0",
    "attributes": {
        "rackId": "Rack1",
        "stateNormal": "true",
        "temperature": "70"
    },
    "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/TempSensor0",
    "thingId": "example1-90ab-cdef-fedc-ba987example"
}
```

下列get-cardinality範例會查詢設定指令碼建立的 10 個感應器，並傳回具有回報異常溫度值之溫度感應器的機架數量。如果溫度值低於 60 或高於 80，則溫度感應器處於異常狀態。

```
aws iot get-cardinality \
    --aggregation-field "attributes.rackId" \
    --query-string "thingName:TempSensor* AND attributes.stateNormal:false"
```

輸出：

```
{
    "cardinality": 2
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的查詢彙總資料<https://docs.aws.amazon.com/iot/latest/developerguide/index-aggregate.html>。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCardinality。

get-effective-policies

以下程式碼範例顯示如何使用 get-effective-policies。

AWS CLI

列出影響物件的政策

下列get-effective-policies範例列出會影響指定物件的政策，包括連接至其所屬之任何群組的政策。

```
aws iot get-effective-policies \
    --thing-name TemperatureSensor-001 \
    --principal arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

輸出：

```
{
    "effectivePolicies": [
        {
            "policyName": "TemperatureSensorPolicy",
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TemperatureSensorPolicy",
            "policyDocument": "{
```

```
                \"Version\": \"2012-10-17\",
                \"Statement\": [
                    {
                        \"Effect\": \"Allow\",
                        \"Action\": [
                            \"iot:Publish\",
                            \"iot:Receive\"
                        ],
                        \"Resource\": [
                            \"arn:aws:iot:us-west-2:123456789012:topic/topic_1\",
                            \"arn:aws:iot:us-west-2:123456789012:topic/topic_2\"
                        ]
                    },
                    {
                        \"Effect\": \"Allow\",
                        \"Action\": [
                            \"iot:Subscribe\"
                        ],
                        \"Resource\": [
                            \"arn:aws:iot:us-west-2:123456789012:topicfilter/
topic_1\",
                            \"arn:aws:iot:us-west-2:123456789012:topicfilter/
topic_2\"
                        ]
                    },
                    {
                        \"Effect\": \"Allow\",
                        \"Action\": [
                            \"iot:Connect\"
                        ],
                        \"Resource\": [
                            \"arn:aws:iot:us-west-2:123456789012:client/basicPubSub
\"
                        ]
                    }
                ]
            }"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的取得物件的有效政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetEffectivePolicies。

get-indexing-configuration

以下程式碼範例顯示如何使用 get-indexing-configuration。

AWS CLI

取得物件索引組態

下列get-indexing-configuration範例會取得 AWS IoT 機群索引的目前組態資料。

```
aws iot get-indexing-configuration
```

輸出：

```
{
    "thingIndexingConfiguration": {
        "thingIndexingMode": "OFF",
        "thingConnectivityIndexingMode": "OFF"
    },
    "thingGroupIndexingConfiguration": {
        "thingGroupIndexingMode": "OFF"
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的管理物件索引。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIndexingConfiguration。

get-job-document

以下程式碼範例顯示如何使用 get-job-document。

AWS CLI

擷取任務的文件

下列get-job-document範例顯示 ID 為 之任務的文件詳細資訊example-job-01。

```
aws iot get-job-document \
    --job-id "example-job-01"
```

輸出：

```
{
    "document": "\n{\n    \"operation\":\"customJob\",\n    \"otherInfo\":
\"someValue\"\n}\n"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的建立和管理任務 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetJobDocument。

get-logging-options

以下程式碼範例顯示如何使用 get-logging-options。

AWS CLI

取得記錄選項

下列get-logging-options範例會取得您 AWS 帳戶的目前記錄選項。

```
aws iot get-logging-options
```

輸出：

```
{
    "roleArn": "arn:aws:iam::123456789012:role/service-role/iotLoggingRole",
    "logLevel": "ERROR"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的標題。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLoggingOptions。

get-ota-update

以下程式碼範例顯示如何使用 get-ota-update。

AWS CLI

擷取 OTA 更新的相關資訊

下列get-ota-update範例顯示指定 OTA 更新的詳細資訊。

```
aws iot get-ota-update \
    --ota-update-id ota12345
```

輸出：

```
{
    "otaUpdateInfo": {
        "otaUpdateId": "ota12345",
        "otaUpdateArn": "arn:aws:iot:us-west-2:123456789012:otaupdate/itsaupdate",
        "creationDate": 1557863215.995,
        "lastModifiedDate": 1557863215.995,
        "description": "A critical update needed right away.",
        "targets": [
           "device1",
           "device2",
           "device3",
           "device4"
        ],
        "targetSelection": "SNAPSHOT",
        "protocols": ["HTTP"],
        "awsJobExecutionsRolloutConfig": {
           "maximumPerMinute": 10
        },
        "otaUpdateFiles": [
            {
                "fileName": "firmware.bin",
                "fileLocation": {
                    "stream": {
                        "streamId": "004",
                        "fileId":123
                    }
                },
                "codeSigning": {
                    "awsSignerJobId": "48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"
                }
            }
        ],
        "roleArn": "arn:aws:iam:123456789012:role/service-role/my_ota_role"
        "otaUpdateStatus": "CREATE_COMPLETE",
        "awsIotJobId": "job54321",
        "awsIotJobArn": "arn:aws:iot:us-west-2:123456789012:job/job54321",
        "errorInfo": {
        }
```

```
    }
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 GetOTAUpdate。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetOtaUpdate。

get-percentiles

以下程式碼範例顯示如何使用 get-percentiles。

AWS CLI

將符合查詢的彙總值分組為百分位數分組

您可以使用下列設定指令碼來建立 10 個代表 10 個溫度感應器的物件。每個新物件都有 1 個屬性。

```
# Bash script. If in other shells, type `bash` before running
Temperatures=(70 71 72 73 74 75 47 97 98 99)
for ((i=0; i<10 ; i++))
do
    thing=$(aws iot create-thing --thing-name "TempSensor$i" --attribute-payload
 attributes="{temperature=${Temperatures[i]}}")
    aws iot describe-thing --thing-name "TempSensor$i"
done
```

設定指令碼的範例輸出：

```
{
    "version": 1,
    "thingName": "TempSensor0",
    "defaultClientId": "TempSensor0",
    "attributes": {
        "temperature": "70"
    },
    "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/TempSensor0",
    "thingId": "example1-90ab-cdef-fedc-ba987example"
}
```

下列get-percentiles範例會查詢設定指令碼建立的 10 個感應器，並傳回每個指定百分位數群組的值。百分位數群組「10」包含彙總欄位值，出現在大約 10% 符合查詢的值中。在下列輸出中，{"percent"： 10.0， "value"： 67.7} 表示大約 10.0% 的溫度值低於 67.7。

```
aws iot get-percentiles \
    --aggregation-field "attributes.temperature" \
    --query-string "thingName:TempSensor*" \
    --percents 10 25 50 75 90
```

輸出：

```
{
    "percentiles": [
        {
            "percent": 10.0,
            "value": 67.7
        },
        {
            "percent": 25.0,
            "value": 71.25
        },
        {
            "percent": 50.0,
            "value": 73.5
        },
        {
            "percent": 75.0,
            "value": 91.5
        },
        {
            "percent": 90.0,
            "value": 98.1
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的查詢彙總資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPercentiles。

get-policy-version

以下程式碼範例顯示如何使用 get-policy-version。

AWS CLI

取得特定政策版本的相關資訊

下列get-policy-version範例取得指定政策第一個版本的相關資訊。

```
aws iot get-policy \
    --policy-name UpdateDeviceCertPolicy
    --policy-version-id "1"
```

輸出：

```
{
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/UpdateDeviceCertPolicy",
    "policyName": "UpdateDeviceCertPolicy",
    "policyDocument": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\":
 \"Allow\", \"Action\":  \"iot:UpdateCertificate\", \"Resource\": \"*\" } ] }",
    "policyVersionId": "1",
    "isDefaultVersion": false,
    "creationDate": 1559925941.924,
    "lastModifiedDate": 1559926175.458,
    "generationId":
 "5066f1b6712ce9d2a1e56399771649a272d6a921762fead080e24fe52f24e042"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 IoT 政策。 AWS IoT

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetPolicyVersion。

get-policy

以下程式碼範例顯示如何使用 get-policy。

AWS CLI

取得政策預設版本的相關資訊

下列get-policy範例會擷取指定政策預設版本的相關資訊。

```
aws iot get-policy \
    --policy-name UpdateDeviceCertPolicy
```

輸出：

```
{
```

```
    "policyName": "UpdateDeviceCertPolicy",
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/UpdateDeviceCertPolicy",
    "policyDocument": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\":
 \"Allow\", \"Action\":  \"iot:UpdateCertificate\", \"Resource\": \"*\" } ] }",
    "defaultVersionId": "2",
    "creationDate": 1559925941.924,
    "lastModifiedDate": 1559925941.924,
    "generationId":
 "5066f1b6712ce9d2a1e56399771649a272d6a921762fead080e24fe52f24e042"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 IoT 政策。 AWS IoT

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetPolicy。

get-registration-code

以下程式碼範例顯示如何使用 get-registration-code。

AWS CLI

取得 AWS 您的帳戶特定註冊碼

下列get-registration-code範例會擷取 AWS 您的帳戶特定註冊碼。

```
aws iot get-registration-code
```

輸出：

```
{
    "registrationCode":
 "15c51ae5e36ba59ba77042df1115862076bea4bd15841c838fcb68d5010a614c"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的使用您自己的憑證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRegistrationCode。

get-statistics

以下程式碼範例顯示如何使用 get-statistics。

AWS CLI

搜尋裝置索引以取得彙總資料

下列get-statistics範例會在其裝置影子中傳回名為 的 屬性connectivity.connected設為的物件數量 false（即未連線的裝置數量）。

```
aws iot get-statistics \
    --index-name AWS_Things \
    --query-string "connectivity.connected:false"
```

輸出：

```
{
    "statistics": {
        "count": 6
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的取得裝置機群的統計資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetStatistics。

get-topic-rule-destination

以下程式碼範例顯示如何使用 get-topic-rule-destination。

AWS CLI

若要取得主題規則目的地

下列get-topic-rule-destination範例取得主題規則目的地的相關資訊。

```
aws iot get-topic-rule-destination \
    --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

輸出：

```
{
    "topicRuleDestination": {
```

```
        "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "status": "DISABLED",
        "httpUrlProperties": {
            "confirmationUrl": "https://example.com"
        }
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的使用主題規則目的地。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTopicRuleDestination。

get-topic-rule

以下程式碼範例顯示如何使用 get-topic-rule。

AWS CLI

取得規則的相關資訊

下列get-topic-rule範例會取得指定規則的相關資訊。

```
aws iot get-topic-rule \
    --rule-name MyRPiLowMoistureAlertRule
```

輸出：

```
{
    "ruleArn": "arn:aws:iot:us-west-2:123456789012:rule/MyRPiLowMoistureAlertRule",
    "rule": {
        "ruleName": "MyRPiLowMoistureAlertRule",
        "sql": "SELECT * FROM '$aws/things/MyRPi/shadow/update/accepted' WHERE
 state.reported.moisture = 'low'\n                    ",
        "description": "Sends an alert whenever soil moisture level readings are too
 low.",
        "createdAt": 1558624363.0,
        "actions": [
            {
                "sns": {
                    "targetArn": "arn:aws:sns:us-
west-2:123456789012:MyRPiLowMoistureTopic",
```

```
                    "roleArn": "arn:aws:iam::123456789012:role/service-role/
MyRPiLowMoistureTopicRole",
                    "messageFormat": "RAW"
                }
            }
        ],
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23"
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的檢視您的規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTopicRule。

get-v2-logging-options

以下程式碼範例顯示如何使用 get-v2-logging-options。

AWS CLI

列出目前的記錄選項

下列get-v2-logging-options範例列出目前的 AWS IoT 記錄選項。

```
aws iot get-v2-logging-options
```

輸出：

```
{
    "roleArn": "arn:aws:iam::094249569039:role/service-role/iotLoggingRole",
    "defaultLogLevel": "WARN",
    "disableAllLogs": false
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的標題。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetV2LoggingOptions。

list-active-violations

以下程式碼範例顯示如何使用 list-active-violations。

AWS CLI

列出作用中違規

下列list-active-violations範例列出指定安全性描述檔的所有違規。

```
aws iot list-active-violations \
    --security-profile-name Testprofile
```

輸出：

```
{
    "activeViolations": [
        {
            "violationId": "174db59167fa474c80a652ad1583fd44",
            "thingName": "iotconsole-1560269126751-1",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "Authorization",
                "metric": "aws:num-authorization-failures",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 10
                    },
                    "durationSeconds": 300,
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
            "lastViolationValue": {
                "count": 0
            },
            "lastViolationTime": 1560293700.0,
            "violationStartTime": 1560279000.0
        },
        {
            "violationId": "c8a9466a093d3b7b35cd44ca58bdbeab",
            "thingName": "TvnQoEoU",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "CellularBandwidth",
                "metric": "aws:message-byte-size",
```

```
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 128
                    },
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
            "lastViolationValue": {
                "count": 110
            },
            "lastViolationTime": 1560369000.0,
            "violationStartTime": 1560276600.0
        },
        {
            "violationId": "74aa393adea02e6648f3ac362beed55e",
            "thingName": "iotconsole-1560269232412-2",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "Authorization",
                "metric": "aws:num-authorization-failures",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 10
                    },
                    "durationSeconds": 300,
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
            "lastViolationValue": {
                "count": 0
            },
            "lastViolationTime": 1560276600.0,
            "violationStartTime": 1560276600.0
        },
        {
            "violationId": "1e6ab5f7cf39a1466fcd154e1377e406",
            "thingName": "TvnQoEoU",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "Authorization",
```

```
                "metric": "aws:num-authorization-failures",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 10
                    },
                    "durationSeconds": 300,
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
            "lastViolationValue": {
                "count": 0
            },
            "lastViolationTime": 1560369000.0,
            "violationStartTime": 1560276600.0
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListActiveViolations。

list-attached-policies

以下程式碼範例顯示如何使用 list-attached-policies。

AWS CLI

範例 1：列出連接到群組的政策

下列list-attached-policies範例列出連接到指定群組的政策。

```
aws iot list-attached-policies \
    --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
```

輸出：

```
{
    "policies": [
        {
            "policyName": "UpdateDeviceCertPolicy",
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
UpdateDeviceCertPolicy"
```

```
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件群組。

範例 2：列出連接到裝置憑證的政策

下列list-attached-policies範例列出連接至裝置憑證的 AWS IoT 政策。憑證由其 ARN 識別。

```
aws iot list-attached-policies \
    --target arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

輸出：

```
{
    "policies": [
        {
            "policyName": "TemperatureSensorPolicy",
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TemperatureSensorPolicy"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAttachedPolicies。

list-audit-findings

以下程式碼範例顯示如何使用 list-audit-findings。

AWS CLI

範例 1：列出稽核中的所有調查結果

下列list-audit-findings範例列出具有指定任務 ID 的 AWS IoT Device Defender 稽核的所有調查結果。

```
aws iot list-audit-findings \
```

```
    --task-id a3aea009955e501a31b764abe1bebd3d
```

輸出：

```
{
    "findings": []
}
```

範例 2：列出稽核檢查類型的調查結果

下列`list-audit-findings`範例顯示 2019 年 6 月 5 日至 2019 年 6 月 19 日期間執行的 AWS IoT Device Defender 稽核調查結果，其中裝置共用裝置憑證。當您指定檢查名稱時，必須提供開始和結束時間。

```
aws iot list-audit-findings \
    --check-name DEVICE_CERTIFICATE_SHARED_CHECK \
    --start-time 1559747125 \
    --end-time 1560962028
```

輸出：

```
{
    "findings": [
        {
            "taskId": "eeef61068b0eb03c456d746c5a26ee04",
            "checkName": "DEVICE_CERTIFICATE_SHARED_CHECK",
            "taskStartTime": 1560161017.172,
            "findingTime": 1560161017.592,
            "severity": "CRITICAL",
            "nonCompliantResource": {
                "resourceType": "DEVICE_CERTIFICATE",
                "resourceIdentifier": {
                    "deviceCertificateId":
 "b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b"
                }
            },
            "relatedResources": [
                {
                    "resourceType": "CLIENT_ID",
                    "resourceIdentifier": {
                        "clientId": "ZipxgAIl"
                    },
```

```
                    "additionalInfo": {
                        "CONNECTION_TIME": "1560086374068"
                    }
                },
                {
                    "resourceType": "CLIENT_ID",
                    "resourceIdentifier": {
                        "clientId": "ZipxgAIl"
                    },
                    "additionalInfo": {
                        "CONNECTION_TIME": "1560081552187",
                        "DISCONNECTION_TIME": "1560086371552"
                    }
                },
                {
                    "resourceType": "CLIENT_ID",
                    "resourceIdentifier": {
                        "clientId": "ZipxgAIl"
                    },
                    "additionalInfo": {
                        "CONNECTION_TIME": "1559289863631",
                        "DISCONNECTION_TIME": "1560081532716"
                    }
                }
            ],
            "reasonForNonCompliance": "Certificate shared by one or more devices.",
            "reasonForNonComplianceCode": "CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES"
        },
        {
            "taskId": "bade6b5efd2e1b1569822f6021b39cf5",
            "checkName": "DEVICE_CERTIFICATE_SHARED_CHECK",
            "taskStartTime": 1559988217.27,
            "findingTime": 1559988217.655,
            "severity": "CRITICAL",
            "nonCompliantResource": {
                "resourceType": "DEVICE_CERTIFICATE",
                "resourceIdentifier": {
                    "deviceCertificateId":
"b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b"
                }
            },
            "relatedResources": [
                {
                    "resourceType": "CLIENT_ID",
```

```
                    "resourceIdentifier": {
                        "clientId": "xShGENLW"
                    },
                    "additionalInfo": {
                        "CONNECTION_TIME": "1559972350825"
                    }
                },
                {
                    "resourceType": "CLIENT_ID",
                    "resourceIdentifier": {
                        "clientId": "xShGENLW"
                    },
                    "additionalInfo": {
                        "CONNECTION_TIME": "1559255062002",
                        "DISCONNECTION_TIME": "1559972350616"
                    }
                }
            ],
            "reasonForNonCompliance": "Certificate shared by one or more devices.",
            "reasonForNonComplianceCode": "CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES"
        },
        {
            "taskId": "c23f6233ba2d35879c4bb2810fb5ffd6",
            "checkName": "DEVICE_CERTIFICATE_SHARED_CHECK",
            "taskStartTime": 1559901817.31,
            "findingTime": 1559901817.767,
            "severity": "CRITICAL",
            "nonCompliantResource": {
                "resourceType": "DEVICE_CERTIFICATE",
                "resourceIdentifier": {
                    "deviceCertificateId":
"b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b"
                }
            },
            "relatedResources": [
                {
                    "resourceType": "CLIENT_ID",
                    "resourceIdentifier": {
                        "clientId": "TvnQoEoU"
                    },
                    "additionalInfo": {
                        "CONNECTION_TIME": "1559826729768"
                    }
                },
```

```
                {
                    "resourceType": "CLIENT_ID",
                    "resourceIdentifier": {
                        "clientId": "TvnQoEoU"
                    },
                    "additionalInfo": {
                        "CONNECTION_TIME": "1559345920964",
                        "DISCONNECTION_TIME": "1559826728402"
                    }
                }
            ],
            "reasonForNonCompliance": "Certificate shared by one or more devices.",
            "reasonForNonComplianceCode": "CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAuditFindings。

`list-audit-mitigation-actions-executions`

以下程式碼範例顯示如何使用 `list-audit-mitigation-actions-executions`。

AWS CLI

列出稽核緩解動作執行的詳細資訊

稽核緩解動作任務會將緩解動作套用至來自 AWS IoT Device Defender 稽核的一或多個調查結果。下列`list-audit-mitigation-actions-executions`範例列出具有指定 之緩解動作任務的詳細資訊，`taskId`以及指定調查結果的詳細資訊。

```
aws iot list-audit-mitigation-actions-executions \
    --task-id myActionsTaskId \
    --finding-id 0edbaaec-2fe1-4cf5-abc9-d4c3e51f7464
```

輸出：

```
{
    "actionsExecutions": [
```

```
        {
            "taskId": "myActionsTaskId",
            "findingId": "0edbaaec-2fe1-4cf5-abc9-d4c3e51f7464",
            "actionName": "ResetPolicyVersionAction",
            "actionId": "1ea0b415-bef1-4a01-bd13-72fb63c59afb",
            "status": "COMPLETED",
            "startTime": "2019-12-10T15:19:13.279000-08:00",
            "endTime": "2019-12-10T15:19:13.337000-08:00"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 ListAuditMitigationActionsExecutions（緩解動作命令）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAuditMitigationActionsExecutions。

list-audit-mitigation-actions-tasks

以下程式碼範例顯示如何使用 list-audit-mitigation-actions-tasks。

AWS CLI

列出稽核緩解動作任務

下列list-audit-mitigation-actions-tasks範例列出在指定期間內套用至調查結果的緩解動作。

```
aws iot list-audit-mitigation-actions-tasks \
    --start-time 1594157400 \
    --end-time 1594157430
```

輸出：

```
{
    "tasks": [
        {
                "taskId": "0062f2d6-3999-488f-88c7-bef005414103",
                "startTime": "2020-07-07T14:30:15.172000-07:00",
            "taskStatus": "COMPLETED"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 ListAuditMitigationActionsTasks（緩解動作命令）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAuditMitigationActionsTasks。

list-audit-suppressions

以下程式碼範例顯示如何使用 list-audit-suppressions。

AWS CLI

列出所有稽核問題清單禁止

下列list-audit-suppressions範例列出所有作用中的稽核調查結果抑制。

```
aws iot list-audit-suppressions
```

輸出：

```
{
    "suppressions": [
        {
        "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",
            "resourceIdentifier": {
                "deviceCertificateId": "c7691e<shortened>"
            },
        "expirationDate": 1597881600.0,
        "suppressIndefinitely": false
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核調查結果抑制。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAuditSuppressions。

list-audit-tasks

以下程式碼範例顯示如何使用 list-audit-tasks。

AWS CLI

從稽核列出所有調查結果

下列list-audit-tasks範例列出 2019 年 6 月 5 日至 2019 年 6 月 12 日期間執行的稽核任務。

```
aws iot list-audit-tasks \
    --start-time 1559747125 \
    --end-time 1560357228
```

輸出：

```
{
    "tasks": [
        {
            "taskId": "a3aea009955e501a31b764abe1bebd3d",
            "taskStatus": "COMPLETED",
            "taskType": "ON_DEMAND_AUDIT_TASK"
        },
        {
            "taskId": "f76b4b5102b632cd9ae38a279c266da1",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "51d9967d9f9ff4d26529505f6d2c444a",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "eeef61068b0eb03c456d746c5a26ee04",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "041c49557b7c7b04c079a49514b55589",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "82c7f2afac1562d18a4560be73998acc",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
```

```
        },
        {
            "taskId": "bade6b5efd2e1b1569822f6021b39cf5",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "c23f6233ba2d35879c4bb2810fb5ffd6",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "ac9086b7222a2f5e2e17bb6fd30b3aeb",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAuditTasks。

list-authorizers

以下程式碼範例顯示如何使用 `list-authorizers`。

AWS CLI

列出您的自訂授權方

下列`list-authorizers`範例列出您 AWS 帳戶中的自訂授權方。

```
aws iot list-authorizers
```

輸出：

```
{
    "authorizers": [
        {
            "authorizerName": "CustomAuthorizer",
            "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer"
```

```
        },
        {
            "authorizerName": "CustomAuthorizer2",
            "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer2"
        },
        {
            "authorizerName": "CustomAuthorizer3",
            "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer3"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 ListAuthorizers。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAuthorizers。

`list-billing-groups`

以下程式碼範例顯示如何使用 `list-billing-groups`。

AWS CLI

列出您 AWS 帳戶和區域的帳單群組

下列`list-billing-groups`範例列出為 AWS 您的帳戶和 AWS 區域定義的所有帳單群組。

```
aws iot list-billing-groups
```

輸出：

```
{
    "billingGroups": [
        {
            "groupName": "GroupOne",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的帳單群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBillingGroups。

list-ca-certificates

以下程式碼範例顯示如何使用 list-ca-certificates。

AWS CLI

列出您 AWS 帳戶中註冊的 CA 憑證

下列list-ca-certificates範例列出在您 AWS 帳戶中註冊的 CA 憑證。

```
aws iot list-ca-certificates
```

輸出：

```
{
    "certificates": [
        {
            "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467",
            "certificateId":
 "f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467",
            "status": "INACTIVE",
            "creationDate": 1569365372.053
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的使用您自己的憑證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCaCertificates。

list-certificates-by-ca

以下程式碼範例顯示如何使用 list-certificates-by-ca。

AWS CLI

列出使用 CA 憑證簽署的所有裝置憑證

下列list-certificates-by-ca範例列出您 AWS 帳戶中使用指定 CA 憑證簽署的所有裝置憑證。

```
aws iot list-certificates-by-ca \
    --ca-certificate-
id f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467
```

輸出：

```
{
    "certificates": [
        {
            "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
            "certificateId":
 "488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
            "status": "ACTIVE",
            "creationDate": 1569363250.557
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 ListCertificatesByCA。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCertificatesByCa。

list-certificates

以下程式碼範例顯示如何使用 list-certificates。

AWS CLI

範例 1：列出在您 AWS 帳戶中註冊的憑證

下列list-certificates範例列出您帳戶中註冊的所有憑證。如果您有超過預設分頁限制 25，您可以使用此命令的nextMarker回應值，並將其提供給下一個命令以取得下一批結果。重複此步驟，直到 nextMarker 傳回時沒有值。

```
aws iot list-certificates
```

輸出：

```
{
    "certificates": [
        {
            "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/604c48437a57b7d5fc5d137c5be75011c6ee67c9a6943683a1acb4b1626bac36",
            "certificateId":
 "604c48437a57b7d5fc5d137c5be75011c6ee67c9a6943683a1acb4b1626bac36",
            "status": "ACTIVE",
            "creationDate": 1556810537.617
        },
        {
            "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/262a1ac8a7d8aa72f6e96e365480f7313aa9db74b8339ec65d34dc3074e1c31e",
            "certificateId":
 "262a1ac8a7d8aa72f6e96e365480f7313aa9db74b8339ec65d34dc3074e1c31e",
            "status": "ACTIVE",
            "creationDate": 1546447050.885
        },
        {
            "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/
b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b",
            "certificateId":
 "b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b",
            "status": "ACTIVE",
            "creationDate": 1546292258.322
        },
        {
            "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/7aebeea3845d14a44ec80b06b8b78a89f3f8a706974b8b34d18f5adf0741db42",
            "certificateId":
 "7aebeea3845d14a44ec80b06b8b78a89f3f8a706974b8b34d18f5adf0741db42",
            "status": "ACTIVE",
            "creationDate": 1541457693.453
        },
        {
            "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/54458aa39ebb3eb39c91ffbbdcc3a6ca1c7c094d1644b889f735a6fc2cd9a7e3",
            "certificateId":
 "54458aa39ebb3eb39c91ffbbdcc3a6ca1c7c094d1644b889f735a6fc2cd9a7e3",
            "status": "ACTIVE",
            "creationDate": 1541113568.611
        },
        {
```

```
            "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
            "certificateId":
 "4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
            "status": "ACTIVE",
            "creationDate": 1541022751.983
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCertificates。

list-custom-metrics

以下程式碼範例顯示如何使用 list-custom-metrics。

AWS CLI

列出您的自訂指標

下列list-custom-metrics範例列出所有自訂指標。

```
aws iot list-custom-metrics \
    --region us-east-1
```

輸出：

```
{
    "metricNames": [
        "batteryPercentage"
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的自訂指標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCustomMetrics。

list-dimensions

以下程式碼範例顯示如何使用 list-dimensions。

AWS CLI

列出您 AWS 帳戶的維度

下列`list-dimensions`範例列出您 AWS 帳戶中定義的所有 AWS IoT Device Defender 維度。

```
aws iot list-dimensions
```

輸出：

```
{
    "dimensionNames": [
        "TopicFilterForAuthMessages",
        "TopicFilterForActivityMessages"
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Detect 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDimensions。

`list-domain-configurations`

以下程式碼範例顯示如何使用 `list-domain-configurations`。

AWS CLI

列出網域組態

下列`list-domain-configurations`範例列出您 AWS 帳戶中具有指定服務類型的網域組態。

```
aws iot list-domain-configurations \
    --service-type "DATA"
```

輸出：

```
{
    "domainConfigurations":
    [
        {
            "domainConfigurationName": "additionalDataDomain",
```

```
            "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh",
            "serviceType": "DATA"
        },

        {
            "domainConfigurationName": "iot:Jobs",
            "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/iot:Jobs",
            "serviceType": "JOBS"
        },
        {
            "domainConfigurationName": "iot:Data-ATS",
            "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/iot:Data-ATS",
            "serviceType": "DATA"
        },
        {
            "domainConfigurationName": "iot:CredentialProvider",
            "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/iot:CredentialProvider",
            "serviceType": "CREDENTIAL_PROVIDER"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的可設定端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDomainConfigurations。

list-indices

以下程式碼範例顯示如何使用 `list-indices`。

AWS CLI

列出設定的搜尋索引

下列`list-indices`範例列出您 AWS 帳戶中所有設定的搜尋索引。如果您尚未啟用物件索引，則可能沒有任何索引。

```
aws iot list-indices
```

輸出：

```
{
    "indexNames": [
        "AWS_Things"
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的管理物件索引。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListIndices。

list-job-executions-for-job

以下程式碼範例顯示如何使用 list-job-executions-for-job。

AWS CLI

列出您 AWS 帳戶中的任務

下列list-job-executions-for-job範例列出您 AWS 帳戶中任務的所有任務執行，由 jobId 指定。

```
aws iot list-job-executions-for-job \
    --job-id my-ota-job
```

輸出：

```
{
     "executionSummaries": [
         {
             "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/my_thing",
             "jobExecutionSummary": {
                 "status": "QUEUED",
                 "queuedAt": "2022-03-07T15:58:42.195000-08:00",
                 "lastUpdatedAt": "2022-03-07T15:58:42.195000-08:00",
                 "executionNumber": 1,
                 "retryAttempt": 0
             }
         }
     ]
 }
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的建立和管理任務 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListJobExecutionsForJob。

list-job-executions-for-thing

以下程式碼範例顯示如何使用 list-job-executions-for-thing。

AWS CLI

列出為物件執行的任務

下列list-job-executions-for-thing範例列出針對名為 的物件執行的所有任務MyRaspberryPi。

```
aws iot list-job-executions-for-thing \
    --thing-name "MyRaspberryPi"
```

輸出：

```
{
    "executionSummaries": [
        {
            "jobId": "example-job-01",
            "jobExecutionSummary": {
                "status": "QUEUED",
                "queuedAt": 1560787023.636,
                "lastUpdatedAt": 1560787023.636,
                "executionNumber": 1
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的建立和管理任務 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListJobExecutionsForThing。

list-jobs

以下程式碼範例顯示如何使用 list-jobs。

AWS CLI

列出您 AWS 帳戶中的任務

下列`list-jobs`範例會列出您 AWS 帳戶中依任務狀態排序的所有任務。

```
aws iot list-jobs
```

輸出：

```
{
    "jobs": [
        {
            "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",
            "jobId": "example-job-01",
            "targetSelection": "SNAPSHOT",
            "status": "IN_PROGRESS",
            "createdAt": 1560787022.733,
            "lastUpdatedAt": 1560787026.294
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的建立和管理任務 (CLI)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListJobs。

`list-mitigation-actions`

以下程式碼範例顯示如何使用 `list-mitigation-actions`。

AWS CLI

列出所有定義的緩解動作

下列`list-mitigation-actions`範例列出您 AWS 帳戶和區域的所有已定義緩解動作。對於每個動作，會列出名稱、ARN 和建立日期。

```
aws iot list-mitigation-actions
```

輸出：

```
{
    "actionIdentifiers": [
        {
            "actionName": "DeactivateCACertAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
DeactivateCACertAction",
            "creationDate": "2019-12-10T11:12:47.574000-08:00"
        },
        {
            "actionName": "ResetPolicyVersionAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
ResetPolicyVersionAction",
            "creationDate": "2019-12-10T11:11:48.920000-08:00"
        },
        {
            "actionName": "PublishFindingToSNSAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
PublishFindingToSNSAction",
            "creationDate": "2019-12-10T11:10:49.546000-08:00"
        },
        {
            "actionName": "AddThingsToQuarantineGroupAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
AddThingsToQuarantineGroupAction",
            "creationDate": "2019-12-10T11:09:35.999000-08:00"
        },
        {
            "actionName": "UpdateDeviceCertAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
UpdateDeviceCertAction",
            "creationDate": "2019-12-10T11:08:44.263000-08:00"
        },
        {
            "actionName": "SampleMitigationAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
SampleMitigationAction",
            "creationDate": "2019-12-10T11:03:41.840000-08:00"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 ListMitigationActions（緩解動作命令）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMitigationActions。

list-mitigations-actions

以下程式碼範例顯示如何使用 list-mitigations-actions。

AWS CLI

列出所有定義的緩解動作

下列list-mitigations-actions範例列出您 AWS 帳戶和區域的所有已定義緩解動作。對於每個動作，會列出名稱、ARN 和建立日期。

```
aws iot list-mitigation-actions
```

輸出：

```
{
    "actionIdentifiers": [
        {
            "actionName": "DeactivateCACertAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
DeactivateCACertAction",
            "creationDate": "2019-12-10T11:12:47.574000-08:00"
        },
        {
            "actionName": "ResetPolicyVersionAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
ResetPolicyVersionAction",
            "creationDate": "2019-12-10T11:11:48.920000-08:00"
        },
        {
            "actionName": "PublishFindingToSNSAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
PublishFindingToSNSAction",
            "creationDate": "2019-12-10T11:10:49.546000-08:00"
        },
        {
            "actionName": "AddThingsToQuarantineGroupAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
AddThingsToQuarantineGroupAction",
            "creationDate": "2019-12-10T11:09:35.999000-08:00"
```

```
        },
        {
            "actionName": "UpdateDeviceCertAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
UpdateDeviceCertAction",
            "creationDate": "2019-12-10T11:08:44.263000-08:00"
        },
        {
            "actionName": "SampleMitigationAction",
            "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
SampleMitigationAction",
            "creationDate": "2019-12-10T11:03:41.840000-08:00"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 ListMitigationActions （緩解動作命令）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMitigationsActions。

`list-ota-updates`

以下程式碼範例顯示如何使用 `list-ota-updates`。

AWS CLI

列出帳戶的 OTA 更新

下列`list-ota-updates`範例列出可用的 OTA 更新。

```
aws iot list-ota-updates
```

輸出：

```
{
    "otaUpdates": [
        {
            "otaUpdateId": "itsaupdate",
            "otaUpdateArn": "arn:aws:iot:us-west-2:123456789012:otaupdate/
itsaupdate",
            "creationDate": 1557863215.995
        }
    ]
```

```
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 ListOTAUpdates。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListOtaUpdates。

list-outgoing-certificates

以下程式碼範例顯示如何使用 list-outgoing-certificates。

AWS CLI

列出要轉移到不同 AWS 帳戶的憑證

下列list-outgoing-certificates範例列出使用 transfer-certificate命令傳輸到不同 AWS 帳戶的所有裝置憑證。

```
aws iot list-outgoing-certificates
```

輸出：

```
{
    "outgoingCertificates": [
        {
            "certificateArn": "arn:aws:iot:us-
west-2:030714055129:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
            "certificateId":
 "488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
            "transferredTo": "030714055129",
            "transferDate": 1569427780.441,
            "creationDate": 1569363250.557
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 ListOutgoingCertificates。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListOutgoingCertificates。

list-policies

以下程式碼範例顯示如何使用 list-policies。

AWS CLI

列出您 AWS 帳戶中定義的政策

下列list-policies範例列出您 AWS 帳戶中定義的所有政策。

```
aws iot list-policies
```

輸出：

```
{
    "policies": [
        {
            "policyName": "UpdateDeviceCertPolicy",
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
UpdateDeviceCertPolicy"
        },
        {
            "policyName": "PlantIoTPolicy",
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/PlantIoTPolicy"
        },
        {
            "policyName": "MyPiGroup_Core-policy",
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/MyPiGroup_Core-
policy"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 IoT 政策。 AWS IoT

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListPolicies。

list-policy-versions

以下程式碼範例顯示如何使用 list-policy-versions。

AWS CLI

範例 1：查看政策的所有版本

下列list-policy-versions範例列出指定政策的所有版本及其建立日期。

```
aws iot list-policy-versions \
    --policy-name LightBulbPolicy
```

輸出：

```
{
    "policyVersions": [
        {
            "versionId": "2",
            "isDefaultVersion": true,
            "createDate": 1559925941.924
        },
        {
            "versionId": "1",
            "isDefaultVersion": false,
            "createDate": 1559925941.924
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 IoT 政策。 AWS IoT

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 ListPolicyVersions。

list-principal-things

以下程式碼範例顯示如何使用 `list-principal-things`。

AWS CLI

列出與委託人連接的物件

下列`list-principal-things`範例列出附加至 ARN 所指定委託人的物件。

```
aws iot list-principal-things \
    --principal arn:aws:iot:us-
west-2:123456789012:cert/2e1eb273792174ec2b9bf4e9b37e6c6c692345499506002a351597670552 78e8
```

輸出：

```
{
```

```
    "things": [
        "DeskLamp",
        "TableLamp"
    ]
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 ListPrincipalThings。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPrincipalThings。

list-provisioning-template-versions

以下程式碼範例顯示如何使用 list-provisioning-template-versions。

AWS CLI

列出佈建範本版本

下列list-provisioning-template-versions範例列出指定佈建範本的可用版本。

```
aws iot list-provisioning-template-versions \
    --template-name "widget-template"
```

輸出：

```
{
    "versions": [
        {
            "versionId": 1,
            "creationDate": 1574800471.339,
            "isDefaultVersion": true
        },
        {
            "versionId": 2,
            "creationDate": 1574801192.317,
            "isDefaultVersion": false
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的 IoT 安全通道。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListProvisioningTemplateVersions。

`list-provisioning-templates`

以下程式碼範例顯示如何使用 `list-provisioning-templates`。

AWS CLI

列出佈建範本

下列`list-provisioning-templates`範例列出您 AWS 帳戶中的所有佈建範本。

```
aws iot list-provisioning-templates
```

輸出：

```
{
    "templates": [
        {
            "templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/
widget-template",
            "templateName": "widget-template",
            "description": "A provisioning template for widgets",
            "creationDate": 1574800471.367,
            "lastModifiedDate": 1574801192.324,
            "enabled": false
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的 IoT 安全通道。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListProvisioningTemplates。

`list-role-aliases`

以下程式碼範例顯示如何使用 `list-role-aliases`。

AWS CLI

列出您 AWS 帳戶中的 AWS IoT 角色別名

下列list-role-aliases範例列出您 AWS 帳戶中的 AWS IoT 角色別名。

```
aws iot list-role-aliases
```

輸出：

```
{
    "roleAliases": [
        "ResidentAlias",
        "ElectricianAlias"
    ]
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 ListRoleAliases。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRoleAliases。

list-scheduled-audits

以下程式碼範例顯示如何使用 list-scheduled-audits。

AWS CLI

列出您 AWS 帳戶的排程稽核

下列list-scheduled-audits範例列出為 AWS 您的帳戶排程的任何稽核。

```
aws iot list-scheduled-audits
```

輸出：

```
{
    "scheduledAudits": [
        {
            "scheduledAuditName": "AWSIoTDeviceDefenderDailyAudit",
            "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
AWSIoTDeviceDefenderDailyAudit",
            "frequency": "DAILY"
        },
        {
            "scheduledAuditName": "AWSDeviceDefenderWeeklyAudit",
```

```
            "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
AWSDeviceDefenderWeeklyAudit",
            "frequency": "WEEKLY",
            "dayOfWeek": "SUN"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListScheduledAudits。

`list-security-profiles-for-target`

以下程式碼範例顯示如何使用 `list-security-profiles-for-target`。

AWS CLI

列出連接到目標的安全設定檔

下列`list-security-profiles-for-target`範例列出連接至未註冊裝置的 AWS IoT Device Defender 安全性描述檔。

```
aws iot list-security-profiles-for-target  \
    --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/
unregistered-things"
```

輸出：

```
{
    "securityProfileTargetMappings": [
        {
            "securityProfileIdentifier": {
                "name": "Testprofile",
                "arn": "arn:aws:iot:us-west-2:123456789012:securityprofile/
Testprofile"
            },
            "target": {
                "arn": "arn:aws:iot:us-west-2:123456789012:all/unregistered-things"
            }
        }
    ]
```

```
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Detect 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSecurityProfilesForTarget。

`list-security-profiles`

以下程式碼範例顯示如何使用 `list-security-profiles`。

AWS CLI

列出您 AWS 帳戶的安全設定檔

下列`list-security-profiles`範例列出您 AWS 帳戶中定義的所有 AWS IoT Device Defender 安全性描述檔。

```
aws iot list-security-profiles
```

輸出：

```
{
    "securityProfileIdentifiers": [
        {
            "name": "Testprofile",
            "arn": "arn:aws:iot:us-west-2:123456789012:securityprofile/Testprofile"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Detect 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSecurityProfiles。

`list-streams`

以下程式碼範例顯示如何使用 `list-streams`。

AWS CLI

列出帳戶中的串流

下列list-streams範例列出您 AWS 帳戶中的所有串流。

```
aws iot list-streams
```

輸出：

```
{
    "streams": [
        {
            "streamId": "stream12345",
            "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream12345",
            "streamVersion": 1,
            "description": "This stream is used for Amazon FreeRTOS OTA Update
 12345."
        },
        {
            "streamId": "stream54321",
            "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream54321",
            "streamVersion": 1,
            "description": "This stream is used for Amazon FreeRTOS OTA Update
 54321."
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 ListStreams。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListStreams。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

顯示與資源相關聯的標籤及其值

下列list-tags-for-resource範例顯示與物件群組 相關聯的標籤和值LightBulbs。

```
aws iot list-tags-for-resource \
    --resource-arn "arn:aws:iot:us-west-2:094249569039:thinggroup/LightBulbs"
```

輸出：

```
{
    "tags": [
        {
            "Key": "Assembly",
            "Value": "Fact1NW"
        },
        {
            "Key": "MyTag",
            "Value": "777"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的標記您的 IoT 資源。 AWS IoT

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

list-targets-for-policy

以下程式碼範例顯示如何使用 list-targets-for-policy。

AWS CLI

列出與 AWS IoT政策相關聯的委託人

下列list-targets-for-policy範例列出連接指定政策的裝置憑證。

```
aws iot list-targets-for-policy \
    --policy-name UpdateDeviceCertPolicy
```

輸出：

```
{
    "targets": [
        "arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
        "arn:aws:iot:us-west-2:123456789012:cert/
d1eb269fb55a628552143c8f96eb3c258fcd5331ea113e766ba0c82bf225f0be"
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTargetsForPolicy。

`list-targets-for-security-profile`

以下程式碼範例顯示如何使用 `list-targets-for-security-profile`。

AWS CLI

列出套用安全性設定檔的目標

下列`list-targets-for-security-profile`範例列出`PossibleIssue`套用名為 的 AWS IoT Device Defender 安全性描述檔的目標。

```
aws iot list-targets-for-security-profile \
    --security-profile-name Testprofile
```

輸出：

```
{
    "securityProfileTargets": [
        {
            "arn": "arn:aws:iot:us-west-2:123456789012:all/unregistered-things"
        },
        {
            "arn": "arn:aws:iot:us-west-2:123456789012:all/registered-things"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Detect 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTargetsForSecurityProfile。

`list-thing-groups-for-thing`

以下程式碼範例顯示如何使用 `list-thing-groups-for-thing`。

AWS CLI

列出物件所屬的群組

下列list-thing-groups-for-thing範例列出指定物件所屬的群組。

```
aws iot list-thing-groups-for-thing \
    --thing-name MyLightBulb
```

輸出：

```
{
    "thingGroups": [
        {
            "groupName": "DeadBulbs",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/DeadBulbs"
        },
        {
            "groupName": "LightBulbs",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListThingGroupsForThing。

list-thing-groups

以下程式碼範例顯示如何使用 list-thing-groups。

AWS CLI

列出您 AWS 帳戶中定義的物件群組

下列describe-thing-group範例列出您 AWS 帳戶中定義的所有物件群組。

```
aws iot list-thing-groups
```

輸出：

```
{
    "thingGroups": [
```

```
        {
            "groupName": "HalogenBulbs",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/HalogenBulbs"
        },
        {
            "groupName": "LightBulbs",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListThingGroups。

`list-thing-principals`

以下程式碼範例顯示如何使用 `list-thing-principals`。

AWS CLI

列出與物件相關聯的委託人

下列`list-thing-principals`範例列出與指定物件相關聯的委託人 (X.509 憑證、IAM 使用者、群組、角色、Amazon Cognito 身分或聯合身分）。

```
aws iot list-thing-principals \
    --thing-name MyRaspberryPi
```

輸出：

```
{
    "principals": [
        "arn:aws:iot:us-
west-2:123456789012:cert/33475ac865079a5ffd5ecd44240640349293facc760642d7d8d5dbb6b4c86893"
    ]
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 ListThingPrincipals。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListThingPrincipals。

list-thing-types

以下程式碼範例顯示如何使用 list-thing-types。

AWS CLI

列出定義的物件類型

下列list-thing-types範例顯示您 AWS 帳戶中定義的物件類型清單。

```
aws iot list-thing-types
```

輸出：

```
{
    "thingTypes": [
        {
            "thingTypeName": "LightBulb",
            "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/
LightBulb",
            "thingTypeProperties": {
                "thingTypeDescription": "light bulb type",
                "searchableAttributes": [
                    "model",
                    "wattage"
                ]
            },
            "thingTypeMetadata": {
            "deprecated": false,
            "creationDate": 1559772562.498
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件類型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListThingTypes。

list-things-in-billing-group

以下程式碼範例顯示如何使用 list-things-in-billing-group。

AWS CLI

列出帳單群組中的物件

下列list-things-in-billing-group範例列出指定帳單群組中的物件。

```
aws iot list-things-in-billing-group \
    --billing-group-name GroupOne
```

輸出：

```
{
    "things": [
        "MyOtherLightBulb",
        "MyLightBulb"
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的帳單群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListThingsInBillingGroup。

list-things-in-thing-group

以下程式碼範例顯示如何使用 list-things-in-thing-group。

AWS CLI

列出屬於群組的物件

下列list-things-in-thing-group範例列出屬於指定物件群組的物件。

```
aws iot list-things-in-thing-group \
    --thing-group-name LightBulbs
```

輸出：

```
{
    "things": [
        "MyLightBulb"
    ]
```

```
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListThingsInThingGroup。

`list-things`

以下程式碼範例顯示如何使用 `list-things`。

AWS CLI

範例 1：列出登錄檔中的所有物件

下列`list-things`範例列出在您 AWS 帳戶的 AWS IoT 登錄檔中定義的物件 （裝置）。

```
aws iot list-things
```

輸出：

```
{
    "things": [
        {
            "thingName": "ThirdBulb",
            "thingTypeName": "LightBulb",
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/ThirdBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 2
        },
        {
            "thingName": "MyOtherLightBulb",
            "thingTypeName": "LightBulb",
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyOtherLightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 3
        },
```

```
        {
            "thingName": "MyLightBulb",
            "thingTypeName": "LightBulb",
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 1
        },
        {
        "thingName": "SampleIoTThing",
        "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/SampleIoTThing",
        "attributes": {},
        "version": 1
        }
    ]
}
```

範例 2：列出具有特定屬性的定義物件

下列list-things範例顯示屬性名為 的物件清單wattage。

```
aws iot list-things \
    --attribute-name wattage
```

輸出：

```
{
    "things": [
        {
            "thingName": "MyLightBulb",
            "thingTypeName": "LightBulb",
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 1
        },
        {
            "thingName": "MyOtherLightBulb",
            "thingTypeName": "LightBulb",
```

```
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyOtherLightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 3
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的如何使用登錄管理物件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListThings。

list-topic-rule-destinations

以下程式碼範例顯示如何使用 list-topic-rule-destinations。

AWS CLI

列出您的主題規則目的地

下列list-topic-rule-destinations範例列出您在目前區域中定義的所有主題規則目的地 AWS 。

```
aws iot list-topic-rule-destinations
```

輸出：

```
{
    "destinationSummaries": [
        {
            "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "status": "ENABLED",
            "httpUrlSummary": {
                "confirmationUrl": "https://example.com"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的使用主題規則目的地。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTopicRuleDestinations。

`list-topic-rules`

以下程式碼範例顯示如何使用 `list-topic-rules`。

AWS CLI

列出您的規則

下列`list-topic-rules`範例列出您已定義的所有規則。

```
aws iot list-topic-rules
```

輸出：

```
{
    "rules": [
        {
            "ruleArn": "arn:aws:iot:us-west-2:123456789012:rule/
MyRPiLowMoistureAlertRule",
            "ruleName": "MyRPiLowMoistureAlertRule",
            "topicPattern": "$aws/things/MyRPi/shadow/update/accepted",
            "createdAt": 1558624363.0,
            "ruleDisabled": false
        },
        {
            "ruleArn": "arn:aws:iot:us-west-2:123456789012:rule/
MyPlantPiMoistureAlertRule",
            "ruleName": "MyPlantPiMoistureAlertRule",
            "topicPattern": "$aws/things/MyPlantPi/shadow/update/accepted",
            "createdAt": 1541458459.0,
            "ruleDisabled": false
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的檢視您的規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTopicRules。

list-v2-logging-levels

以下程式碼範例顯示如何使用 list-v2-logging-levels。

AWS CLI

列出記錄層級

下列list-v2-logging-levels範例列出設定的日誌層級。如果未設定記錄層級，則當您執行此命令時NotConfiguredException，會發生 。

```
aws iot list-v2-logging-levels
```

輸出：

```
{
    "logTargetConfigurations": [
        {
            "logTarget": {
                "targetType": "DEFAULT"
            },
            "logLevel": "ERROR"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListV2LoggingLevels。

list-violation-events

以下程式碼範例顯示如何使用 list-violation-events。

AWS CLI

列出一段時間內的安全性描述檔違規

下列list-violation-events範例列出在 2019 年 6 月 5 日至 2019 年 6 月 12 日期間，針對目前 AWS 帳戶和 AWS 區域的 all AWS IoT Device Defender 安全性描述檔所發生的違規。

```
aws iot list-violation-events \
    --start-time 1559747125 \
    --end-time 1560351925
```

輸出：

```
{
    "violationEvents": [
        {
            "violationId": "174db59167fa474c80a652ad1583fd44",
            "thingName": "iotconsole-1560269126751-1",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "Authorization",
                "metric": "aws:num-authorization-failures",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 10
                    },
                    "durationSeconds": 300,
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
            "metricValue": {
                "count": 0
            },
            "violationEventType": "in-alarm",
            "violationEventTime": 1560279000.0
        },
        {
            "violationId": "c8a9466a093d3b7b35cd44ca58bdbeab",
            "thingName": "TvnQoEoU",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "CellularBandwidth",
                "metric": "aws:message-byte-size",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 128
                    },
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
            "metricValue": {
```

```
                "count": 110
            },
            "violationEventType": "in-alarm",
            "violationEventTime": 1560276600.0
        },
        {
            "violationId": "74aa393adea02e6648f3ac362beed55e",
            "thingName": "iotconsole-1560269232412-2",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "Authorization",
                "metric": "aws:num-authorization-failures",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 10
                    },
                    "durationSeconds": 300,
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
            "metricValue": {
                "count": 0
            },
            "violationEventType": "in-alarm",
            "violationEventTime": 1560276600.0
        },
        {
            "violationId": "1e6ab5f7cf39a1466fcd154e1377e406",
            "thingName": "TvnQoEoU",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "Authorization",
                "metric": "aws:num-authorization-failures",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 10
                    },
                    "durationSeconds": 300,
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
```

```
            },
            "metricValue": {
                "count": 0
            },
            "violationEventType": "in-alarm",
            "violationEventTime": 1560276600.0
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Detect 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListViolationEvents。

register-ca-certificate

以下程式碼範例顯示如何使用 register-ca-certificate。

AWS CLI

註冊憑證授權單位 (CA) 憑證

下列register-ca-certificate範例會註冊 CA 憑證。命令提供 CA 憑證和金鑰驗證憑證，證明您擁有與 CA 憑證相關聯的私有金鑰。

```
aws iot register-ca-certificate \
    --ca-certificate file://rootCA.pem \
    --verification-cert file://verificationCert.pem
```

輸出：

```
{
    "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f05362952074264600 58cd1467",
    "certificateId":
 "f4efed62c0142f16af278166f61962501165c4f05362952074264600 58cd1467"
 }
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 RegisterCACertificate。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterCaCertificate。

register-certificate

以下程式碼範例顯示如何使用 register-certificate。

AWS CLI

註冊自我簽署的裝置憑證

下列register-certificate範例會註冊由 rootCA.pem CA 憑證簽署deviceCert.pem的裝置憑證。您必須先註冊 CA 憑證，才能使用它來註冊自我簽署的裝置憑證。自我簽署憑證必須由您傳遞至此命令的相同 CA 憑證簽署。

```
aws iot register-certificate \
    --certificate-pem file://deviceCert.pem \
    --ca-certificate-pem file://rootCA.pem
```

輸出：

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
    "certificateId":
 "488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142"
 }
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 RegisterCertificate。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterCertificate。

register-thing

以下程式碼範例顯示如何使用 register-thing。

AWS CLI

註冊物件

下列register-thing範例使用佈建範本註冊物件。

```
aws iot register-thing \
    --template-body '{"Parameters":{"ThingName":
{"Type":"String"},"AWS::IoT::Certificate::Id":{"Type":"String"}},"Resources":
 {"certificate":{"Properties":{"CertificateId":
```

```
{"Ref":"AWS::IoT::Certificate::Id"},"Status":"Active"},"Type":"AWS::IoT::Certificate"},"poli
{"Properties":{"PolicyName":"MyIotPolicy"},"Type":"AWS::IoT::Policy"},"thing":
{"OverrideSettings":
{"AttributePayload":"MERGE","ThingGroups":"DO_NOTHING","ThingTypeName":"REPLACE"},"Propertie
{"AttributePayload":{},"ThingGroups":[],"ThingName":
{"Ref":"ThingName"},"ThingTypeName":"VirtualThings"},"Type":"AWS::IoT::Thing"}}}' \
    --parameters '{"ThingName":"Register-thing-
trial-1","AWS::IoT::Certificate::Id":"799a9ea048a1e6aea42b55EXAMPLEf8697b4bafcd77a318a3068e3
```

輸出：

```
{
    "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCCAkGgAwIBAgIUYLk81I35cIppobpw
HiOJ2jNjboIwDQYJKoZIhvcNAQEL
\nBQAwTTFLMEkGA1UECwxCQW1hem9uIFdlYiBTZXJ2aWNlcyBPPUFtYXpvbi
5jb20g\nSW5jLiBMPVNlYXR0bGUgU1Q9V2FzaGluZ3RvbiBDPVVTMB4XDTIwMDcyMzE2NDUw
\nOVoXDTQ5MTIzMT
IzNTk1OVowHjEcMBoGA1UEAwwTQVdTIElvVCBDZXJ0aWZpY2F0\nZTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCC
AQoCggEBAO71uADhdBajqTmgrMV5\nmCFfBZQRMo1MdtVoZr2X+M4MzL
+RARrtUzH9a2SMAckeX8KeblIOTKzORI
RDXnyE
\n6lVOwjgAsd0ku22rFxex4eG2ikha7pYYkvuToqA7L3TxItRvfKrxRI4ZfJoFPip4\nKqiuBJVNOGKTcQ
Hd1RNOrddwwu6kFJLeKDmEXAMPLEdUF0N+qfR9yKnZQkm
+g6Q2\nGXu7u0W3hn6nlRN8qVoka0uW12p53xM7oHVz
Gf+cxKBxlbOhGkp6yCfTSkUBm3Sp\n9zLw35kiHXVm4EVpwgNlnk6XcIGIkw8a/iy4pzmvuGAANY1/uU/
zgCjymw
ZT5S30\nBV0CAwEAAaNgMF4wHwYDVR0jBBgwFoAUGx0tCcU3q2n1WXAuUCv6hugXjKswHQYD
\nVR0OBBYEFOVtvZ
9Aj2RYFnkX7Iu01XTRUdxgMAwGA1UdEwEB/wQCMAAwDgYDVR0P\nAQH/
BAQDAgeAMA0GCSqGSIb3DQEBCwUAA4IB
AQCXCQcpOtubS5ftOsDMTcpP/jNX
\nDHyArxmjpSc2aCdmm7WX59lTKWyAdxGAvqaDVWqTo0oXI7tZ8w7aINlGi5
pXnifx\n3SBebMUoBbTktrC97yUaeL025mCFv8emDnTR/fE7PTsBKjW0g/rrfpwBxZLXDFwN
\nnqkQjy3EDfifj2
6j0xYIqqWMPogyn4srOCKynS5wMJuQZlHQOnabVwnwK4Y0Mflp
\np9+4susFUR9aT3BT1AcIwqSpzhlKhh4Iz7ND
kRn4amsUT210jg/zOO1Ow+BTHcVQ\nJly8XDu0CWSu04q6SnaBzHmlySIajxuRTP/AdfRouP1OXe
+qlbPOBcvVvF
8o\n-----END CERTIFICATE-----\n",
    "resourceArns": {
        "certificate": "arn:aws:iot:us-
west-2:571032923833:cert/799a9ea048a1e6aea42b55EXAMPLEf8697b4bafcd77a318a3068e30404b9233c",
```

```
        "thing": "arn:aws:iot:us-west-2:571032923833:thing/Register-thing-trial-1"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的由信任的使用者佈建。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterThing。

reject-certificate-transfer

以下程式碼範例顯示如何使用 reject-certificate-transfer。

AWS CLI

拒絕憑證轉移

下列reject-certificate-transfer範例會拒絕從另一個 AWS 帳戶轉移指定的裝置憑證。

```
aws iot reject-certificate-transfer \
    --certificate-
id f0f33678c7c9a046e5cc87b2b1a58dfa0beec26db78addd5e605d630e05c7fc8
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的將憑證轉移到另一個帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RejectCertificateTransfer。

remove-thing-from-billing-group

以下程式碼範例顯示如何使用 remove-thing-from-billing-group。

AWS CLI

從帳單群組移除物件

下列remove-thing-from-billing-group範例會從帳單群組移除指定的物件。

```
aws iot remove-thing-from-billing-group \
    --billing-group-name GroupOne \
    --thing-name MyOtherLightBulb
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的帳單群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveThingFromBillingGroup。

remove-thing-from-thing-group

以下程式碼範例顯示如何使用 remove-thing-from-thing-group。

AWS CLI

從物件群組中移除物件

下列remove-thing-from-thing-group範例會從物件群組中移除指定的物件。

```
aws iot remove-thing-from-thing-group \
    --thing-name bulb7 \
    --thing-group-name DeadBulbs
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件群組 <https://docs.aws.amazon.com/iot/latest/developerguide/thing-groups.html >。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveThingFromThingGroup。

replace-topic-rule

以下程式碼範例顯示如何使用 replace-topic-rule。

AWS CLI

更新主題的規則定義

下列replace-topic-rule範例會更新指定的規則，以在土壤濕度等級讀數過低時傳送 SNS 提醒。

```
aws iot replace-topic-rule \
    --rule-name MyRPiLowMoistureAlertRule \
    --topic-rule-payload "{\"sql\": \"SELECT * FROM '$aws/things/MyRPi/shadow/
update/accepted' WHERE state.reported.moisture = 'low'\", \"description\": \"Sends
 an alert when soil moisture level readings are too low.\",\"actions\": [{\"sns
```

```
\":{\"targetArn\":\"arn:aws:sns:us-west-2:123456789012:MyRPiLowMoistureTopic\",
\"roleArn\":\"arn:aws:iam::123456789012:role/service-role/MyRPiLowMoistureTopicRole
\",\"messageFormat\": \"RAW\"}}],\"ruleDisabled\": false,\"awsIotSqlVersion\":
\"2016-03-23\"}"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的建立 IoT 規則。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReplaceTopicRule。

search-index

以下程式碼範例顯示如何使用 search-index。

AWS CLI

查詢物件索引

下列search-index範例會查詢AWS_Things索引中是否有類型為 的物件LightBulb。

```
aws iot search-index \
    --index-name "AWS_Things" \
    --query-string "thingTypeName:LightBulb"
```

輸出：

```
{
    "things": [
        {
            "thingName": "MyLightBulb",
            "thingId": "40da2e73-c6af-406e-b415-15acae538797",
            "thingTypeName": "LightBulb",
            "thingGroupNames": [
                "LightBulbs",
                "DeadBulbs"
            ],
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "connectivity": {
                "connected": false
```

```
            }
        },
        {
            "thingName": "ThirdBulb",
            "thingId": "615c8455-33d5-40e8-95fd-3ee8b24490af",
            "thingTypeName": "LightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "connectivity": {
                "connected": false
            }
        },
        {
            "thingName": "MyOtherLightBulb",
            "thingId": "6dae0d3f-40c1-476a-80c4-1ed24ba6aa11",
            "thingTypeName": "LightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "connectivity": {
                "connected": false
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的管理物件索引。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchIndex。

set-default-authorizer

以下程式碼範例顯示如何使用 set-default-authorizer。

AWS CLI

設定預設授權方

下列set-default-authorizer範例會將名為 CustomAuthorizer的自訂授權方設定為預設授權方。

```
aws iot set-default-authorizer \
    --authorizer-name CustomAuthorizer
```

輸出：

```
{
    "authorizerName": "CustomAuthorizer",
    "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer"
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 CreateDefaultAuthorizer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetDefaultAuthorizer。

set-default-policy-version

以下程式碼範例顯示如何使用 set-default-policy-version。

AWS CLI

設定政策的預設版本

下列set-default-policy-version範例會將名為 2的政策的預設版本設定為 UpdateDeviceCertPolicy。

```
aws iot set-default-policy-version \
    --policy-name UpdateDeviceCertPolicy \
    --policy-version-id 2
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 SetDefaultPolicyVersion。

set-v2-logging-level

以下程式碼範例顯示如何使用 set-v2-logging-level。

AWS CLI

設定物件群組的記錄層級

下列set-v2-logging-level範例會將記錄層級設定為記錄指定物件群組的警告。

```
aws iot set-v2-logging-level \
    --log-target "{\"targetType\":\"THING_GROUP\",\"targetName\":\"LightBulbs\"}" \
    --log-level WARN
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetV2LoggingLevel。

set-v2-logging-options

以下程式碼範例顯示如何使用 set-v2-logging-options。

AWS CLI

設定記錄選項

下列set-v2-logging-options範例會將預設記錄詳細程度設定為 ERROR，並指定用於記錄的 ARN。

```
aws iot set-v2-logging-options \
    --default-log-level ERROR \
    --role-arn "arn:aws:iam::094249569039:role/service-role/iotLoggingRole"
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetV2LoggingOptions。

start-audit-mitigation-actions-task

以下程式碼範例顯示如何使用 start-audit-mitigation-actions-task。

AWS CLI

將緩解動作套用至稽核的調查結果

下列start-audit-mitigation-actions-task範例會將 ResetPolicyVersionAction動作（清除政策）套用至指定的單一調查結果。

```
aws iot start-audit-mitigation-actions-task \
```

```
    --task-id "myActionsTaskId" \
    --target "findingIds=[\"0edbaaec-2fe1-4cf5-abc9-d4c3e51f7464\"]" \
    --audit-check-to-actions-mapping
 "IOT_POLICY_OVERLY_PERMISSIVE_CHECK=[\"ResetPolicyVersionAction\"]" \
    --client-request-token "adhadhahda"
```

輸出：

```
{
    "taskId": "myActionsTaskId"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 StartAuditMitigationActionsTask （緩解動作命令）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartAuditMitigationActionsTask。

`start-on-demand-audit-task`

以下程式碼範例顯示如何使用 `start-on-demand-audit-task`。

AWS CLI

立即開始稽核

下列`start-on-demand-audit-task`範例會啟動 AWS IoT Device Defender 稽核，並執行三次憑證檢查。

```
aws iot start-on-demand-audit-task \
    --target-check-
names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK REVOKED_CA_CERTIFICATE
```

輸出：

```
{
    "taskId": "a3aea009955e501a31b764abe1bebd3d"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartOnDemandAuditTask。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

指定資源的標籤索引鍵和值

下列tag-resource範例會將具有索引鍵 Assembly和 值的標籤Fact1NW套用至物件群組 LightBulbs。

```
aws iot tag-resource \
    --tags Key=Assembly,Value="Fact1NW" \
    --resource-arn "arn:aws:iot:us-west-2:094249569039:thinggroup/LightBulbs"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的標記您的 IoT 資源。 AWS IoT

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

test-authorization

以下程式碼範例顯示如何使用 test-authorization。

AWS CLI

測試您的 AWS IoT 政策

下列test-authorization範例會測試與指定委託人相關聯的 AWS IoT 政策。

```
aws iot test-authorization \
    --auth-infos actionType=CONNECT,resources=arn:aws:iot:us-
east-1:123456789012:client/client1 \
    --principal arn:aws:iot:us-west-2:123456789012:cert/
aab1068f7f43ac3e3cae4b3a8aa3f308d2a750e6350507962e32c1eb465d9775
```

輸出：

```
{
    "authResults": [
        {
```

```
            "authInfo": {
                "actionType": "CONNECT",
                "resources": [
                    "arn:aws:iot:us-east-1:123456789012:client/client1"
                ]
            },
            "allowed": {
                "policies": [
                    {
                        "policyName": "TestPolicyAllowed",
                        "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TestPolicyAllowed"
                    }
                ]
            },
            "denied": {
                "implicitDeny": {
                    "policies": [
                        {
                            "policyName": "TestPolicyDenied",
                            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TestPolicyDenied"
                        }
                    ]
                },
                "explicitDeny": {
                    "policies": [
                        {
                            "policyName": "TestPolicyExplicitDenied",
                            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TestPolicyExplicitDenied"
                        }
                    ]
                }
            },
            "authDecision": "IMPLICIT_DENY",
            "missingContextValues": []
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 TestAuthorization。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TestAuthorization。

test-invoke-authorizer

以下程式碼範例顯示如何使用 test-invoke-authorizer。

AWS CLI

測試您的自訂授權方

下列test-invoke-authorizer範例 testS 您的自訂授權方。

```
aws iot test-invoke-authorizer \
    --authorizer-name IoTAuthorizer \
    --token allow \
    --token-signature "mE0GvaHqy9nER/
FdgtJX5lXYEJ3b3vE7t1gEszc0TKGgLKWXTnPkb2AbKnOAZ8lGyoN5dVtWDWVmr25m7+
+zjbYIMk2TBvyGXhOmvKFBPkdgyA43KL6SiZy0cTqlPMcQDsP7VX2rXr7CTowCxSNKphGXdQe0/
I5dQ+JO6KUaHwCmupt0/MejKtaNwiia064j6wprOAUwG5S1IYFuRd0X
+wfo8pb0DubAIX1Ua705kuhRUcTx4SxUShEYKmN4IDEvLB6FsIr0B2wvB7y4iPmcajxzGl02ExvyCUNctCV9dYlRRGJj
```

輸出：

```
{
    "isAuthenticated": true,
    "principalId": "principalId",
    "policyDocuments": [
        "{"Version":"2012-10-17","Statement":
[{"Action":"iot:Publish","Effect":"Allow","Resource":"arn:aws:iot:us-
west-2:123456789012:topic/customauthtesting"}]}"
    ],
    "refreshAfterInSeconds": 600,
    "disconnectAfterInSeconds": 3600
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 TestInvokeAuthorizer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TestInvokeAuthorizer。

transfer-certificate

以下程式碼範例顯示如何使用 transfer-certificate。

AWS CLI

將裝置憑證轉移到不同的 AWS 帳戶

下列transfer-certificate範例會將裝置憑證轉移到另一個 AWS 帳戶。憑證和 AWS 帳戶由 ID 識別。

```
aws iot transfer-certificate \
    --certificate-
id 488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142 \
    --target-aws-account 030714055129
```

輸出：

```
{
    "transferredCertificateArn": "arn:aws:iot:us-
west-2:030714055129:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142"
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的將憑證轉移到另一個帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TransferCertificate。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源移除標籤金鑰

下列untag-resource範例會從物件群組 移除標籤MyTag及其值LightBulbs。

```
command
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的標記您的 IoT 資源。 AWS IoT

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-account-audit-configuration

以下程式碼範例顯示如何使用 update-account-audit-configuration。

AWS CLI

範例 1：為稽核通知啟用 Amazon SNS 通知

下列update-account-audit-configuration範例會啟用 AWS IoT Device Defender 稽核通知的 Amazon SNS 通知，指定用於寫入該目標的目標和角色。

```
aws iot update-account-audit-configuration \
    --audit-notification-target-configurations "SNS={targetArn=\"arn:aws:sns:us-west-2:123456789012:ddaudits\",roleArn=\"arn:aws:iam::123456789012:role/service-role/AWSIoTDeviceDefenderAudit\",enabled=true}"
```

此命令不會產生輸出。

範例 2：啟用稽核檢查

下列update-account-audit-configuration範例會啟用名為 的 AWS IoT Device Defender 稽核檢查AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK。如果稽核檢查是 AWS 帳戶targetCheckNames一或多個排程稽核的一部分，則無法停用稽核檢查。

```
aws iot update-account-audit-configuration \
    --audit-check-configurations
 "{\"AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK\":{\"enabled\":true}}"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAccountAuditConfiguration。

update-audit-suppression

以下程式碼範例顯示如何使用 update-audit-suppression。

AWS CLI

更新稽核問題清單抑制

下列update-audit-suppression範例會將稽核發現結果禁止的過期日期更新為 2020-09-21。

```
aws iot update-audit-suppression \
    --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
    --resource-identifier deviceCertificateId=c7691e<shortened> \
    --no-suppress-indefinitely \
    --expiration-date 2020-09-21
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核調查結果抑制。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAuditSuppression。

update-authorizer

以下程式碼範例顯示如何使用 update-authorizer。

AWS CLI

更新自訂授權方

下列update-authorizer範例為 CustomAuthorizer2到 的狀態INACTIVE。

```
aws iot update-authorizer \
    --authorizer-name CustomAuthorizer2 \
    --status INACTIVE
```

輸出：

```
{
    "authorizerName": "CustomAuthorizer2",
    "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer2"
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 UpdateAuthorizer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAuthorizer。

update-billing-group

以下程式碼範例顯示如何使用 update-billing-group。

AWS CLI

更新帳單群組的相關資訊

下列update-billing-group範例會更新指定帳單群組的描述。

```
aws iot update-billing-group \
    --billing-group-name GroupOne \
    --billing-group-properties "billingGroupDescription=\"Primary bulb billing group
\""
```

輸出：

```
{
    "version": 2
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的帳單群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateBillingGroup。

update-ca-certificate

以下程式碼範例顯示如何使用 update-ca-certificate。

AWS CLI

更新憑證授權單位 (CA) 憑證

下列update-ca-certificate範例會將指定的 CA 憑證設定為 ACTIVE 狀態。

```
aws iot update-ca-certificate \
    --certificate-
id f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467 \
    --new-status ACTIVE
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT API 參考中的 UpdateCACertificate。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateCaCertificate。

update-certificate

以下程式碼範例顯示如何使用 update-certificate。

AWS CLI

更新裝置憑證

下列update-certificate範例會將指定的裝置憑證設定為 INACTIVE 狀態。

```
aws iot update-certificate \
    --certificate-
id d1eb269fb55a628552143c8f96eb3c258fcd5331ea113e766ba0c82bf225f0be \
    --new-status INACTIVE
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT API 參考中的 UpdateCertificate。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateCertificate。

update-custom-metric

以下程式碼範例顯示如何使用 update-custom-metric。

AWS CLI

更新自訂指標

下列update-custom-metric範例會將自訂指標更新為具有新的 display-name。

```
aws iot update-custom-metric \
    --metric-name batteryPercentage \
    --display-name 'remaining battery percentage on device' \
    --region us-east-1
```

輸出：

```
{
    "metricName": "batteryPercentage",
    "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/
batteryPercentage",
```

```
    "metricType": "number",
    "displayName": "remaining battery percentage on device",
    "creationDate": "2020-11-17T23:01:35.110000-08:00",
    "lastModifiedDate": "2020-11-17T23:02:12.879000-08:00"
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的自訂指標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateCustomMetric。

update-dimension

以下程式碼範例顯示如何使用 update-dimension。

AWS CLI

更新維度

下列update-dimension範例會更新維度。

```
aws iot update-dimension \
    --name TopicFilterForAuthMessages \
    --string-values device/${iot:ClientId}/auth
```

輸出：

```
{
    "name": "TopicFilterForAuthMessages",
    "lastModifiedDate": 1585866222.317,
    "stringValues": [
        "device/${iot:ClientId}/auth"
    ],
    "creationDate": 1585854500.474,
    "type": "TOPIC_FILTER",
    "arn": "arn:aws:iot:us-west-2:1234564789012:dimension/
TopicFilterForAuthMessages"
}
```

如需詳細資訊，請參閱《AWS IoT Core 開發人員指南》中的使用維度調整安全性設定檔中的指標範圍。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDimension。

update-domain-configuration

以下程式碼範例顯示如何使用 update-domain-configuration。

AWS CLI

更新網域組態

下列update-domain-configuration範例會停用指定的網域組態。

```
aws iot update-domain-configuration \
    --domain-configuration-name "additionalDataDomain" \
    --domain-configuration-status "DISABLED"
```

輸出：

```
{
    "domainConfigurationName": "additionalDataDomain",
    "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的可設定端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDomainConfiguration。

update-dynamic-thing-group

以下程式碼範例顯示如何使用 update-dynamic-thing-group。

AWS CLI

更新動態物件群組

下列update-dynamic-thing-group範例會更新指定的動態物件群組。它提供描述並更新查詢字串，以變更群組成員資格條件。

```
aws iot update-dynamic-thing-group \
    --thing-group-name "RoomTooWarm"
    --thing-group-properties "thingGroupDescription=\"This thing group contains
 rooms warmer than 65F.\"" \
    --query-string "attributes.temperature>65"
```

輸出：

```
{
    "version": 2
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的動態物件群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDynamicThingGroup。

update-event-configurations

以下程式碼範例顯示如何使用 update-event-configurations。

AWS CLI

顯示要發佈的事件類型

下列update-event-configurations範例會更新組態，以在新增、更新或刪除 CA 憑證時啟用訊息。

```
aws iot update-event-configurations \
    --event-configurations "{\"CA_CERTIFICATE\":{\"Enabled\":true}}"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的事件訊息。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateEventConfigurations。

update-indexing-configuration

以下程式碼範例顯示如何使用 update-indexing-configuration。

AWS CLI

啟用物件索引

下列update-indexing-configuration範例可讓物件索引支援使用 AWS_Things 索引搜尋登錄檔資料、影子資料和物件連線狀態。

```
aws iot update-indexing-configuration
```

```
    --thing-indexing-
configuration thingIndexingMode=REGISTRY_AND_SHADOW,thingConnectivityIndexingMode=STATUS
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的管理物件索引。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateIndexingConfiguration。

update-job

以下程式碼範例顯示如何使用 update-job。

AWS CLI

取得任務的詳細狀態

下列update-job範例會取得 ID 為 之任務的詳細狀態example-job-01。

```
aws iot describe-job \
    --job-id "example-job-01"
```

輸出：

```
{
    "job": {
        "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",
        "jobId": "example-job-01",
        "targetSelection": "SNAPSHOT",
        "status": "IN_PROGRESS",
        "targets": [
            "arn:aws:iot:us-west-2:123456789012:thing/MyRaspberryPi"
        ],
        "description": "example job test",
        "presignedUrlConfig": {},
        "jobExecutionsRolloutConfig": {},
        "createdAt": 1560787022.733,
        "lastUpdatedAt": 1560787026.294,
        "jobProcessDetails": {
            "numberOfCanceledThings": 0,
            "numberOfSucceededThings": 0,
            "numberOfFailedThings": 0,
            "numberOfRejectedThings": 0,
```

```
            "numberOfQueuedThings": 1,
            "numberOfInProgressThings": 0,
            "numberOfRemovedThings": 0,
            "numberOfTimedOutThings": 0
        },
        "timeoutConfig": {}
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的建立和管理任務 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateJob。

update-mitigation-action

以下程式碼範例顯示如何使用 update-mitigation-action。

AWS CLI

更新緩解動作

下列update-mitigation-action範例會更新名為 的指定緩解動作AddThingsToQuarantineGroupAction、變更物件群組名稱，並將overrideDynamicGroups設定為 false。您可以使用 describe-mitigation-action命令來驗證您的變更。

```
aws iot update-mitigation-action \
    --cli-input-json "{ \"actionName\": \"AddThingsToQuarantineGroupAction\",
 \"actionParams\": { \"addThingsToThingGroupParams\": {\"thingGroupNames\":
[\"QuarantineGroup2\"],\"overrideDynamicGroups\": false}}}"
```

輸出：

```
{
    "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
AddThingsToQuarantineGroupAction",
    "actionId": "2fd2726d-98e1-4abf-b10f-09465ccd6bfa"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 UpdateMitigationAction （緩解動作命令）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateMitigationAction。

update-provisioning-template

以下程式碼範例顯示如何使用 update-provisioning-template。

AWS CLI

更新佈建範本

下列update-provisioning-template範例會修改指定佈建範本的描述和角色 ARN，並啟用範本。

```
aws iot update-provisioning-template \
    --template-name widget-template \
    --enabled \
    --description "An updated provisioning template for widgets" \
    --provisioning-role-arn arn:aws:iam::504350838278:role/Provision_role
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的 IoT 安全通道。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateProvisioningTemplate。

update-role-alias

以下程式碼範例顯示如何使用 update-role-alias。

AWS CLI

更新角色別名

下列update-role-alias範例會更新LightBulbRole角色別名。

```
aws iot update-role-alias \
    --role-alias LightBulbRole \
    --role-arn arn:aws:iam::123456789012:role/lightbulbrole-001
```

輸出：

```
{
    "roleAlias": "LightBulbRole",
```

```
    "roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/LightBulbRole"
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 UpdateRoleAlias。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRoleAlias。

update-scheduled-audit

以下程式碼範例顯示如何使用 update-scheduled-audit。

AWS CLI

更新排定的稽核定義

下列update-scheduled-audit範例會變更 an AWS IoT Device Defender 排程稽核的目標檢查名稱。

```
aws iot update-scheduled-audit \
    --scheduled-audit-name WednesdayCertCheck \
    --target-check-
names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK REVOKED_CA_CERTIFICATE
```

輸出：

```
{
    "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
WednesdayCertCheck"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的稽核命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateScheduledAudit。

update-security-profile

以下程式碼範例顯示如何使用 update-security-profile。

AWS CLI

變更安全性描述檔

下列update-security-profile範例會更新 an AWS IoT Device Defender 安全性描述檔的描述和行為。

```
aws iot update-security-profile \
    --security-profile-name PossibleIssue \
    --security-profile-description "Check to see if authorization fails 12 times in
 5 minutes or if cellular bandwidth exceeds 128"  \
    --behaviors "[{\"name\":\"CellularBandwidth\",\"metric\":\"aws:message-byte-size
\",\"criteria\":{\"comparisonOperator\":\"greater-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}},{\"name
\":\"Authorization\",\"metric\":\"aws:num-authorization-failures\",\"criteria\":
{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":12},\"durationSeconds
\":300,\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]"
```

輸出：

```
{
    "securityProfileName": "PossibleIssue",
    "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/
PossibleIssue",
    "securityProfileDescription": "check to see if authorization fails 12 times in 5
 minutes or if cellular bandwidth exceeds 128",
    "behaviors": [
        {
            "name": "CellularBandwidth",
            "metric": "aws:message-byte-size",
            "criteria": {
                "comparisonOperator": "greater-than",
                "value": {
                    "count": 128
                },
                "consecutiveDatapointsToAlarm": 1,
                "consecutiveDatapointsToClear": 1
            }
        },
        {
            "name": "Authorization",
            "metric": "aws:num-authorization-failures",
            "criteria": {
                "comparisonOperator": "less-than",
                "value": {
                    "count": 12
                },
```

```
                "durationSeconds": 300,
                "consecutiveDatapointsToAlarm": 1,
                "consecutiveDatapointsToClear": 1
            }
        }
    ],
    "version": 2,
    "creationDate": 1560278102.528,
    "lastModifiedDate": 1560352711.207
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Detect 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSecurityProfile。

update-stream

以下程式碼範例顯示如何使用 update-stream。

AWS CLI

更新串流

下列update-stream範例會更新現有的串流。串流版本會遞增 1。

```
aws iot update-stream \
    --cli-input-json file://update-stream.json
```

update-stream.json 的內容：

```
{
    "streamId": "stream12345",
    "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",
    "files": [
        {
            "fileId": 123,
            "s3Location": {
                "bucket":"codesign-ota-bucket",
                "key":"48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"
            }
        }
    ]
```

```
    "roleArn": "arn:aws:iam:us-west-2:123456789012:role/service-role/
my_ota_stream_role"
}
```

輸出：

```
{
    "streamId": "stream12345",
    "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream12345",
    "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",
    "streamVersion": 2
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 UpdateStream。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateStream。

update-thing-group

以下程式碼範例顯示如何使用 update-thing-group。

AWS CLI

更新物件群組的定義

下列update-thing-group範例會更新指定物件群組的定義，變更描述和兩個屬性。

```
aws iot update-thing-group \
    --thing-group-name HalogenBulbs \
    --thing-group-properties "thingGroupDescription=\"Halogen bulb group\",
 attributePayload={attributes={Manufacturer=AnyCompany,wattage=60}}"
```

輸出：

```
{
    "version": 2
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateThingGroup。

update-thing-groups-for-thing

以下程式碼範例顯示如何使用 update-thing-groups-for-thing。

AWS CLI

變更物件所屬的群組

下列update-thing-groups-for-thing範例MyLightBulb會從名為 的群組中移除名為 的物件，DeadBulbs並replaceableItems同時將其新增至名為 的群組。

```
aws iot update-thing-groups-for-thing \
    --thing-name MyLightBulb \
    --thing-groups-to-add "replaceableItems" \
    --thing-groups-to-remove "DeadBulbs"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateThingGroupsForThing。

update-thing

以下程式碼範例顯示如何使用 update-thing。

AWS CLI

將物件與物件類型建立關聯

下列update-thing範例會將 AWS IoT 登錄檔中的物件與物件類型建立關聯。當您建立關聯時，您可以為物件類型定義的屬性提供值。

```
aws iot update-thing \
    --thing-name "MyOtherLightBulb" \
    --thing-type-name "LightBulb" \
    --attribute-payload "{"attributes": {"wattage":"75", "model":"123"}}"
```

此命令不會產生輸出。使用 describe-thing命令查看結果。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的物件類型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateThing。

update-topic-rule-destination

以下程式碼範例顯示如何使用 update-topic-rule-destination。

AWS CLI

範例 1：啟用主題規則目的地

下列update-topic-rule-destination範例可讓流量流向主題規則目的地。

```
aws iot update-topic-rule-destination \
    --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \
    --status ENABLED
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的啟用主題規則目的地。

範例 2：停用主題規則目的地

下列update-topic-rule-destination範例會停用通往主題規則目的地的流量。

```
aws iot update-topic-rule-destination \
    --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \
    --status DISABLED
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的停用主題規則目的地。

範例 3：傳送新的確認訊息

下列update-topic-rule-destination範例會傳送主題規則目的地的新確認訊息。

```
aws iot update-topic-rule-destination \
    --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \
```

```
    --status IN_PROGRESS
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的傳送新的確認訊息。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateTopicRuleDestination。

validate-security-profile-behaviors

以下程式碼範例顯示如何使用 validate-security-profile-behaviors。

AWS CLI

範例 1：驗證安全性設定檔的行為參數

下列validate-security-profile-behaviors範例會驗證 an AWS IoT Device Defender 安全性描述檔的一組格式正確行為。

```
aws iot validate-security-profile-behaviors \
    --behaviors "[{\"name\":\"CellularBandwidth\",\"metric\":\"aws:message-byte-size
\",\"criteria\":{\"comparisonOperator\":\"greater-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}},{\"name
\":\"Authorization\",\"metric\":\"aws:num-authorization-failures\",\"criteria\":
{\"comparisonOperator\":\"greater-than\",\"value\":{\"count\":12},\"durationSeconds
\":300,\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]"
```

輸出：

```
{
    "valid": true,
    "validationErrors": []
}
```

範例 2：驗證安全性描述檔的不正確行為參數

下列validate-security-profile-behaviors範例會驗證一組行為，其中包含 an AWS IoT Device Defender 安全性描述檔的錯誤。

```
aws iot validate-security-profile-behaviors \
    --behaviors "[{\"name\":\"CellularBandwidth\",\"metric\":\"aws:message-byte-size
\",\"criteria\":{\"comparisonOperator\":\"greater-than\",\"value\":{\"count\":128},
```

```
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}},{\"name
\":\"Authorization\",\"metric\":\"aws:num-authorization-failures\",\"criteria\":
{\"comparisonOperator\":\"greater-than\",\"value\":{\"count\":12},\"durationSeconds
\":300,\"consecutiveDatapointsToAlarm\":100000,\"consecutiveDatapointsToClear
\":1}}]"
```

輸出：

```
{
    "valid": false,
    "validationErrors": [
        {
            "errorMessage": "Behavior Authorization is malformed.
 consecutiveDatapointsToAlarm 100000 should be in range[1,10]"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Detect 命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ValidateSecurityProfileBehaviors。

AWS IoT Analytics 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS IoT Analytics。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`batch-put-message`

以下程式碼範例顯示如何使用 `batch-put-message`。

AWS CLI

將訊息傳送至頻道

下列batch-put-message範例會傳送訊息至指定的頻道。

```
aws iotanalytics batch-put-message \
    --cli-binary-format raw-in-base64-out \
    --cli-input-json file://batch-put-message.json
```

batch-put-message.json 的內容：

```
{
    "channelName": "mychannel",
    "messages": [
        {
            "messageId": "0001",
            "payload": "eyAidGVtcGVyYXR1cmUiOiAyMCB9"
        }
    ]
}
```

輸出：

```
{
    "batchPutMessageErrorEntries": []
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 BatchPutMessage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchPutMessage。

cancel-pipeline-reprocessing

以下程式碼範例顯示如何使用 cancel-pipeline-reprocessing。

AWS CLI

取消透過管道重新處理資料

下列cancel-pipeline-reprocessing範例會取消透過指定管道重新處理資料。

```
aws iotanalytics cancel-pipeline-reprocessing \
    --pipeline-name mypipeline \
    --reprocessing-id "6ad2764f-fb13-4de3-b101-4e74af03b043"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 CancelPipelineReprocessing。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelPipelineReprocessing。

create-channel

以下程式碼範例顯示如何使用 create-channel。

AWS CLI

建立頻道

下列create-channel範例會建立具有指定組態的頻道。頻道會從 MQTT 主題收集資料，並會在將資料發佈到管道前，先將未處理的原始訊息封存。

```
aws iotanalytics create-channel \
    --cli-input-json file://create-channel.json
```

create-channel.json 的內容：

```
{
    "channelName": "mychannel",
    "retentionPeriod": {
        "unlimited": true
    },
    "tags": [
        {
            "key": "Environment",
            "value": "Production"
        }
    ]
}
```

輸出：

```
{
```

```
    "channelArn": "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel",
    "channelName": "mychannel",
    "retentionPeriod": {
        "unlimited": true
    }
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 CreateChannel。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateChannel。

create-dataset-content

以下程式碼範例顯示如何使用 create-dataset-content。

AWS CLI

建立資料集的內容

下列create-dataset-content範例會透過套用 queryAction(SQL 查詢） 或 containerAction（執行容器化應用程式） 來建立指定資料集的內容。

```
aws iotanalytics create-dataset-content \
    --dataset-name mydataset
```

輸出：

```
{
    "versionId": "d494b416-9850-4670-b885-ca22f1e89d62"
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 CreateDatasetContent。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDatasetContent。

create-dataset

以下程式碼範例顯示如何使用 create-dataset。

AWS CLI

建立資料集

下列create-dataset範例會建立資料集。資料集會套用 queryAction(SQL 查詢） 或 containerAction（執行容器化應用程式） 來存放從資料存放區擷取的資料。此操作會建立資料集的骨架。您可以呼叫 CreateDatasetContent或根據您trigger指定的 自動填入資料集。

```
aws iotanalytics create-dataset \
    --cli-input-json file://create-dataset.json
```

create-dataset.json 的內容：

```
{
    "datasetName": "mydataset",
    "actions": [
        {
            "actionName": "myDatasetAction",
            "queryAction": {
                "sqlQuery": "SELECT * FROM mydatastore"
            }
        }
    ],
    "retentionPeriod": {
        "unlimited": true
    },
    "tags": [
        {
            "key": "Environment",
            "value": "Production"
        }
    ]
}
```

輸出：

```
{
    "datasetName": "mydataset",
    "retentionPeriod": {
        "unlimited": true
    },
    "datasetArn": "arn:aws:iotanalytics:us-west-2:123456789012:dataset/mydataset"
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 CreateDataset。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDataset。

create-datastore

以下程式碼範例顯示如何使用 create-datastore。

AWS CLI

建立資料存放區

下列create-datastore範例會建立資料存放區，這是訊息的儲存庫。

```
aws iotanalytics create-datastore \
    --cli-input-json file://create-datastore.json
```

create-datastore.json 的內容：

```
{
    "datastoreName": "mydatastore",
    "retentionPeriod": {
        "numberOfDays": 90
    },
    "tags": [
        {
            "key": "Environment",
            "value": "Production"
        }
    ]
}
```

輸出：

```
{
    "datastoreName": "mydatastore",
    "datastoreArn": "arn:aws:iotanalytics:us-west-2:123456789012:datastore/
mydatastore",
    "retentionPeriod": {
        "numberOfDays": 90,
        "unlimited": false
    }
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 CreateDatastore。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateDatastore。

create-pipeline

以下程式碼範例顯示如何使用 create-pipeline。

AWS CLI

建立 IoT Analytics 管道

下列create-pipeline範例會建立管道。管道會從頻道取用訊息，並能在將訊息存放於資料存放區之前，先處理該訊息。您必須同時指定頻道和資料存放區活動，以及選擇性地在pipelineActivities陣列中指定多達 23 個其他活動。

```
aws iotanalytics create-pipeline \
    --cli-input-json file://create-pipeline.json
```

create-pipeline.json 的內容：

```
{
    "pipelineName": "mypipeline",
    "pipelineActivities": [
        {
            "channel": {
                "name": "myChannelActivity",
                "channelName": "mychannel",
                "next": "myMathActivity"
            }
        },
        {
            "datastore": {
                "name": "myDatastoreActivity",
                "datastoreName": "mydatastore"
            }
        },
        {
            "math": {
                "name": "myMathActivity",
                "math": "((temp - 32) * 5.0) / 9.0",
                "attribute": "tempC",
```

```
                "next": "myDatastoreActivity"
            }
        }
    ],
    "tags": [
        {
            "key": "Environment",
            "value": "Beta"
        }
    ]
}
```

輸出：

```
{
    "pipelineArn": "arn:aws:iotanalytics:us-west-2:123456789012:pipeline/
mypipeline",
    "pipelineName": "mypipeline"
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 CreatePipeline。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePipeline。

`delete-channel`

以下程式碼範例顯示如何使用 `delete-channel`。

AWS CLI

刪除 IoT Analytics 頻道

下列`delete-channel`範例會刪除指定的頻道。

```
aws iotanalytics delete-channel \
    --channel-name mychannel
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 DeleteChannel。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteChannel。

delete-dataset-content

以下程式碼範例顯示如何使用 delete-dataset-content。

AWS CLI

刪除資料集內容

下列delete-dataset-content範例會刪除指定資料集的內容。

```
aws iotanalytics delete-dataset-content \
    --dataset-name mydataset
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 DeleteDatasetContent。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDatasetContent。

delete-dataset

以下程式碼範例顯示如何使用 delete-dataset。

AWS CLI

刪除資料集

下列delete-dataset範例會刪除指定的資料集。在執行此操作之前，您不需要刪除資料集的內容。

```
aws iotanalytics delete-dataset \
    --dataset-name mydataset
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 DeleteDataset。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDataset。

delete-datastore

以下程式碼範例顯示如何使用 delete-datastore。

AWS CLI

刪除資料存放區

下列delete-datastore範例會刪除指定的資料存放區。

```
aws iotanalytics delete-datastore \
    --datastore-name mydatastore
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 DeleteDatastore。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteDatastore。

delete-pipeline

以下程式碼範例顯示如何使用 delete-pipeline。

AWS CLI

刪除管道

下列delete-pipeline範例會刪除指定的管道。

```
aws iotanalytics delete-pipeline \
    --pipeline-name mypipeline
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 DeletePipeline。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePipeline。

describe-channel

以下程式碼範例顯示如何使用 describe-channel。

AWS CLI

擷取頻道的相關資訊

下列describe-channel範例顯示指定頻道的詳細資訊，包括統計資料。

```
aws iotanalytics describe-channel \
    --channel-name mychannel \
    --include-statistics
```

輸出：

```
{
    "statistics": {
        "size": {
            "estimatedSizeInBytes": 402.0,
            "estimatedOn": 1561504380.0
        }
    },
    "channel": {
        "status": "ACTIVE",
        "name": "mychannel",
        "lastUpdateTime": 1557860351.001,
        "creationTime": 1557860351.001,
        "retentionPeriod": {
            "unlimited": true
        },
        "arn": "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 DescribeChannel。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeChannel。

describe-dataset

以下程式碼範例顯示如何使用 describe-dataset。

AWS CLI

擷取資料集的相關資訊

下列describe-dataset範例顯示指定資料集的詳細資訊。

```
aws iotanalytics describe-dataset \
    --dataset-name mydataset
```

輸出：

```
{
    "dataset": {
        "status": "ACTIVE",
        "contentDeliveryRules": [],
        "name": "mydataset",
        "lastUpdateTime": 1557859240.658,
        "triggers": [],
        "creationTime": 1557859240.658,
        "actions": [
            {
                "actionName": "query_32",
                "queryAction": {
                    "sqlQuery": "SELECT * FROM mydatastore",
                    "filters": []
                }
            }
        ],
        "retentionPeriod": {
            "numberOfDays": 90,
            "unlimited": false
        },
        "arn": "arn:aws:iotanalytics:us-west-2:123456789012:dataset/mydataset"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 DescribeDataset。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDataset。

describe-datastore

以下程式碼範例顯示如何使用 describe-datastore。

AWS CLI

擷取資料存放區的相關資訊

下列describe-datastore範例顯示指定資料存放區的詳細資訊，包括統計資料。

```
aws iotanalytics describe-datastore \
    --datastore-name mydatastore \
```

```
    --include-statistics
```

輸出：

```
{
    "datastore": {
        "status": "ACTIVE",
        "name": "mydatastore",
        "lastUpdateTime": 1557858971.02,
        "creationTime": 1557858971.02,
        "retentionPeriod": {
            "unlimited": true
        },
        "arn": "arn:aws:iotanalytics:us-west-2:123456789012:datastore/mydatastore"
    },
    "statistics": {
        "size": {
            "estimatedSizeInBytes": 397.0,
            "estimatedOn": 1561592040.0
        }
    }
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 DescribeDatastore。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDatastore。

describe-logging-options

以下程式碼範例顯示如何使用 describe-logging-options。

AWS CLI

擷取目前的記錄選項

下列describe-logging-options範例顯示目前的 AWS IoT Analytics 記錄選項。

```
aws iotanalytics describe-logging-options
```

此命令不會產生輸出。輸出：

```
{
```

```
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/service-role/myIoTAnalyticsRole",
        "enabled": true,
        "level": "ERROR"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 DescribeLoggingOptions。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLoggingOptions。

describe-pipeline

以下程式碼範例顯示如何使用 describe-pipeline。

AWS CLI

擷取管道的相關資訊

下列describe-pipeline範例顯示指定管道的詳細資訊。

```
aws iotanalytics describe-pipeline \
    --pipeline-name mypipeline
```

輸出：

```
{
    "pipeline": {
        "activities": [
            {
                "channel": {
                    "channelName": "mychannel",
                    "name": "mychannel_28",
                    "next": "mydatastore_29"
                }
            },
            {
                "datastore": {
                    "datastoreName": "mydatastore",
                    "name": "mydatastore_29"
                }
            }
        ],
```

```
        "name": "mypipeline",
        "lastUpdateTime": 1561676362.515,
        "creationTime": 1557859124.432,
        "reprocessingSummaries": [
            {
                "status": "SUCCEEDED",
                "creationTime": 1561676362.189,
                "id": "6ad2764f-fb13-4de3-b101-4e74af03b043"
            }
        ],
        "arn": "arn:aws:iotanalytics:us-west-2:123456789012:pipeline/mypipeline"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 DescribePipeline。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePipeline。

get-dataset-content

以下程式碼範例顯示如何使用 get-dataset-content。

AWS CLI

擷取資料集的內容

下列get-dataset-content範例會將資料集的內容擷取為預先簽章URIs。

```
aws iotanalytics get-dataset-content --dataset-name mydataset
```

輸出：

```
{
    "status": {
        "state": "SUCCEEDED"
    },
    "timestamp": 1557863215.995,
    "entries": [
        {
            "dataURI": "https://aws-radiant-
dataset-12345678-1234-1234-1234-123456789012.s3.us-west-2.amazonaws.com/
results/12345678-e8b3-46ba-b2dd-efe8d86cf385.csv?X-Amz-Security-Token=...-Amz-
Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20190628T173437Z&X-Amz-SignedHeaders=host&X-
```

```
Amz-Expires=7200&X-Amz-Credential=...F20190628%2Fus-west-2%2Fs3%2Faws4_request&X-
Amz-Signature=..."
        }
    ]
}
```

如需詳細資訊，請參閱 指南中的 GetDatasetContent。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDatasetContent。

`list-channels`

以下程式碼範例顯示如何使用 `list-channels`。

AWS CLI

擷取頻道清單

下列`list-channels`範例顯示可用頻道的摘要資訊。

```
aws iotanalytics list-channels
```

輸出：

```
{
    "channelSummaries": [
        {
            "status": "ACTIVE",
            "channelName": "mychannel",
            "creationTime": 1557860351.001,
            "lastUpdateTime": 1557860351.001
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 ListChannels。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListChannels。

`list-dataset-contents`

以下程式碼範例顯示如何使用 `list-dataset-contents`。

AWS CLI

列出資料集內容的相關資訊

下列list-dataset-contents範例列出已建立之資料集內容的相關資訊。

```
aws iotanalytics list-dataset-contents \
    --dataset-name mydataset
```

輸出：

```
{
    "datasetContentSummaries": [
        {
            "status": {
                "state": "SUCCEEDED"
            },
            "scheduleTime": 1557863215.995,
            "version": "b10ea2a9-66c1-4d99-8d1f-518113b738d0",
            "creationTime": 1557863215.995
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 ListDatasetContents。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDatasetContents。

list-datasets

以下程式碼範例顯示如何使用 list-datasets。

AWS CLI

擷取資料集的相關資訊

下列list-datasets範例列出可用資料集的摘要資訊。

```
aws iotanalytics list-datasets
```

輸出：

```
{
    "datasetSummaries": [
        {
            "status": "ACTIVE",
            "datasetName": "mydataset",
            "lastUpdateTime": 1557859240.658,
            "triggers": [],
            "creationTime": 1557859240.658,
            "actions": [
                {
                    "actionName": "query_32",
                    "actionType": "QUERY"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 ListDatasets。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDatasets。

`list-datastores`

以下程式碼範例顯示如何使用 `list-datastores`。

AWS CLI

擷取資料存放區的清單

下列`list-datastores`範例顯示可用資料存放區的摘要資訊。

```
aws iotanalytics list-datastores
```

輸出：

```
{
    "datastoreSummaries": [
        {
            "status": "ACTIVE",
            "datastoreName": "mydatastore",
            "creationTime": 1557858971.02,
```

```
            "lastUpdateTime": 1557858971.02
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 ListDatastores。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListDatastores。

list-pipelines

以下程式碼範例顯示如何使用 list-pipelines。

AWS CLI

擷取管道清單

下列list-pipelines範例顯示可用管道的清單。

```
aws iotanalytics list-pipelines
```

輸出：

```
{
    "pipelineSummaries": [
        {
            "pipelineName": "mypipeline",
            "creationTime": 1557859124.432,
            "lastUpdateTime": 1557859124.432,
            "reprocessingSummaries": []
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 ListPipelines。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPipelines。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出資源的標籤

下列list-tags-for-resource範例列出您已連接至指定資源的標籤。

```
aws iotanalytics list-tags-for-resource \
    --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel"
```

輸出：

```
{
    "tags": [
        {
            "value": "bar",
            "key": "foo"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 ListTagsForResource。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

put-logging-options

以下程式碼範例顯示如何使用 put-logging-options。

AWS CLI

設定或更新記錄選項

下列put-logging-options範例會設定或更新 AWS IoT Analytics 記錄選項。如果您更新任何loggingOptions欄位的值，變更最多可能需要一分鐘才會生效。此外，如果您變更附加到您在 "roleArn" 欄位中指定的角色的政策（例如，更正無效的政策），該變更最多可能需要五分鐘才會生效。

```
aws iotanalytics put-logging-options \
    --cli-input-json file://put-logging-options.json
```

put-logging-options.json 的內容：

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/service-role/myIoTAnalyticsRole",
        "level": "ERROR",
        "enabled": true
    }
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 PutLoggingOptions。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutLoggingOptions。

run-pipeline-activity

以下程式碼範例顯示如何使用 run-pipeline-activity。

AWS CLI

模擬管道活動

下列run-pipeline-activity範例模擬在訊息承載上執行管道活動的結果。

```
aws iotanalytics run-pipeline-activity \
    --pipeline-activity file://maths.json \
    --payloads file://payloads.json
```

maths.json 的內容：

```
{
    "math": {
        "name": "MyMathActivity",
        "math": "((temp - 32) * 5.0) / 9.0",
        "attribute": "tempC"
    }
}
```

payloads.json 的內容：

```
[
```

```
    "{\"humidity\": 52, \"temp\": 68 }",
    "{\"humidity\": 52, \"temp\": 32 }"
]
```

輸出：

```
{
    "logResult": "",
    "payloads": [
        "eyJodW1pZGl0eSI6NTIsInRlbXAiOjY4LCJ0ZW1wQyI6MjB9",
        "eyJodW1pZGl0eSI6NTIsInRlbXAiOjMyLCJ0ZW1wQyI6MH0="
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 RunPipelineActivity。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RunPipelineActivity。

sample-channel-data

以下程式碼範例顯示如何使用 sample-channel-data。

AWS CLI

從頻道擷取範例訊息

下列sample-channel-data範例會從指定時間範圍內擷取的指定頻道擷取訊息範例。您最多可以擷取 10 則訊息。

```
aws iotanalytics sample-channel-data \
    --channel-name mychannel
```

輸出：

```
{
    "payloads": [
        "eyAidGVtcGVyYXR1cmUiOiAyMCB9",
        "eyAiZm9vIjogImJhciIgfQ=="
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 SampleChannelData。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SampleChannelData。

start-pipeline-reprocessing

以下程式碼範例顯示如何使用 start-pipeline-reprocessing。

AWS CLI

開始管道重新處理

下列start-pipeline-reprocessing範例會透過指定的管道開始重新處理原始訊息資料。

```
aws iotanalytics start-pipeline-reprocessing \
    --pipeline-name mypipeline
```

輸出：

```
{
    "reprocessingId": "6ad2764f-fb13-4de3-b101-4e74af03b043"
}
```

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 StartPipelineReprocessing。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartPipelineReprocessing。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

新增或修改資源的標籤

下列tag-resource範例會新增或修改連接至指定資源的標籤。

```
aws iotanalytics tag-resource \
    --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel" \
    --tags "[{\"key\": \"Environment\", \"value\": \"Production\"}]"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 TagResource。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源移除標籤

下列untag-resource範例會從指定的資源移除具有指定金鑰名稱的標籤。

```
aws iotanalytics untag-resource \
    --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel" \
    --tag-keys "[\"Environment\"]"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 UntagResource <https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API_UntagResource.html >。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-channel

以下程式碼範例顯示如何使用 update-channel。

AWS CLI

修改頻道

下列update-channel範例會修改指定頻道的設定。

```
aws iotanalytics update-channel \
    --cli-input-json file://update-channel.json
```

update-channel.json 的內容：

```
{
    "channelName": "mychannel",
    "retentionPeriod": {
        "numberOfDays": 92
    }
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 UpdateChannel。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateChannel。

update-dataset

以下程式碼範例顯示如何使用 update-dataset。

AWS CLI

更新資料集

下列update-dataset範例會修改指定資料集的設定。

```
aws iotanalytics update-dataset \
    --cli-input-json file://update-dataset.json
```

update-dataset.json 的內容：

```
{
    "datasetName": "mydataset",
    "actions": [
        {
            "actionName": "myDatasetUpdateAction",
            "queryAction": {
                "sqlQuery": "SELECT * FROM mydatastore"
            }
        }
    ],
    "retentionPeriod": {
        "numberOfDays": 92
    }
```

```
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 UpdateDataset <https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API_UpdateDataset.html >。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDataset。

update-datastore

以下程式碼範例顯示如何使用 update-datastore。

AWS CLI

更新資料存放區

下列update-datastore範例會修改指定資料存放區的設定。

```
aws iotanalytics update-datastore \
    --cli-input-json file://update-datastore.json
```

update-datastore.json 的內容：

```
{
    "datastoreName": "mydatastore",
    "retentionPeriod": {
        "numberOfDays": 93
    }
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 UpdateDatastore。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDatastore。

update-pipeline

以下程式碼範例顯示如何使用 update-pipeline。

AWS CLI

更新管道

下列update-pipeline範例會修改指定管道的設定。您必須同時指定頻道和資料存放區活動，並選擇性地在pipelineActivities陣列中指定多達 23 個額外的活動。

```
aws iotanalytics update-pipeline \
    --cli-input-json file://update-pipeline.json
```

update-pipeline.json 的內容：

```
{
    "pipelineName": "mypipeline",
    "pipelineActivities": [
        {
            "channel": {
                "name": "myChannelActivity",
                "channelName": "mychannel",
                "next": "myMathActivity"
            }
        },
        {
            "datastore": {
                "name": "myDatastoreActivity",
                "datastoreName": "mydatastore"
            }
        },
        {
            "math": {
                "name": "myMathActivity",
                "math": "(((temp - 32) * 5.0) / 9.0) + 273.15",
                "attribute": "tempK",
                "next": "myDatastoreActivity"
            }
        }
    ]
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Analytics API 參考中的 UpdatePipeline。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePipeline。

使用 的 Device Advisor 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Device Advisor 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-suite-definition`

以下程式碼範例顯示如何使用 `create-suite-definition`。

AWS CLI

範例 1：建立 IoT Device Advisor 測試套件

下列`create-suite-definition`範例會使用指定的套件定義組態，在 AWS IoT 中建立裝置建議程式測試套件。

```
aws iotdeviceadvisor create-suite-definition \
    --suite-definition-configuration '{ \
        "suiteDefinitionName": "TestSuiteName", \
        "devices": [{"thingArn":"arn:aws:iot:us-east-1:123456789012:thing/
MyIotThing"}], \
        "intendedForQualification": false, \
        "rootGroup": "{\"configuration\":{},\"tests\":[{\"name\":\"MQTT Connect\",
\"configuration\":{\"EXECUTION_TIMEOUT\":120},\"tests\":[{\"name\":\"MQTT_Connect\",
\"configuration\":{},\"test\":{\"id\":\"MQTT_Connect\",\"testCase\":null,\"version
\":\"0.0.0\"}}]}]}", \
        "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole"}'
```

輸出：

```
{
    "suiteDefinitionId": "0jtsgio7yenu",
    "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-
east-1:123456789012:suitedefinition/0jtsgio7yenu",
    "suiteDefinitionName": "TestSuiteName",
    "createdAt": "2022-12-02T11:38:13.263000-05:00"
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的建立測試套件定義。

範例 2：建立 IoT Device Advisor 最新資格測試套件

下列create-suite-definition範例會使用指定的套件定義組態，在 AWS IoT 中建立具有最新版本的裝置顧問資格測試套件。

```
aws iotdeviceadvisor create-suite-definition \
    --suite-definition-configuration '{ \
        "suiteDefinitionName": "TestSuiteName", \
        "devices": [{"thingArn":"arn:aws:iot:us-east-1:123456789012:thing/
MyIotThing"}], \
        "intendedForQualification": true, \
        "rootGroup": "", \
        "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole"}'
```

輸出：

```
{
    "suiteDefinitionId": "txgsuolk2myj",
    "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-
east-1:123456789012:suitedefinition/txgsuolk2myj",
    "suiteDefinitionName": "TestSuiteName",
    "createdAt": "2022-12-02T11:38:13.263000-05:00"
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的建立測試套件定義。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSuiteDefinition。

delete-suite-definition

以下程式碼範例顯示如何使用 delete-suite-definition。

AWS CLI

刪除 IoT Device Advisor 測試套件

下列delete-suite-definition範例會刪除具有指定套件定義 ID 的裝置建議程式測試套件。

```
aws iotdeviceadvisor delete-suite-definition \
    --suite-definition-id 0jtsgio7yenu
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT API 參考中的 DeleteSuiteDefinition。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSuiteDefinition。

get-endpoint

以下程式碼範例顯示如何使用 get-endpoint。

AWS CLI

範例 1：取得 IoT Device Advisor 帳戶層級端點的相關資訊

下列get-endpoint範例會取得裝置顧問帳戶層級測試端點的相關資訊。

```
aws iotdeviceadvisor get-endpoint
```

輸出：

```
{
    "endpoint": "t6y4c143x9sfo.deviceadvisor.iot.us-east-1.amazonaws.com"
}
```

範例 2：取得 IoT Device Advisor 裝置層級端點的相關資訊

下列get-endpoint範例會使用指定的 物件arn 或 certificate-arn，取得裝置建議程式裝置層級測試端點的相關資訊。

```
aws iotdeviceadvisor get-endpoint \
    --thing-arn arn:aws:iot:us-east-1:123456789012:thing/MyIotThing
```

輸出：

```
{
    "endpoint": "tdb7719be5t6y4c143x9sfo.deviceadvisor.iot.us-east-1.amazonaws.com"
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的取得測試端點。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetEndpoint。

get-suite-definition

以下程式碼範例顯示如何使用 get-suite-definition。

AWS CLI

取得 IoT Device Advisor 測試套件的相關資訊

下列get-suite-definition範例取得具有指定套件定義 ID 之 Aevice Advisor 測試套件的相關資訊。

```
aws iotdeviceadvisor get-suite-definition \
    --suite-definition-id qqcsmtyyjabl
```

輸出：

```
{
    "suiteDefinitionId": "qqcsmtyyjabl",
    "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-
east-1:123456789012:suitedefinition/qqcsmtyyjabl",
    "suiteDefinitionVersion": "v1",
    "latestVersion": "v1",
    "suiteDefinitionConfiguration": {
        "suiteDefinitionName": "MQTT connection",
        "devices": [],
        "intendedForQualification": false,
        "isLongDurationTest": false,
        "rootGroup": "{\"configuration\":{},\"tests\":[{\"id\":\"uta5d9j1kvwc\",
\"name\":\"Test group 1\",\"configuration\":{},\"tests\":[{\"id\":\"awr8pq5vc9yp\",
\"name\":\"MQTT Connect\",\"configuration\":{},\"test\":{\"id\":\"MQTT_Connect\",
\"testCase\":null,\"version\":\"0.0.0\"}}]}]}",
```

```
        "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole",
        "protocol": "MqttV3_1_1"
    },
    "createdAt": "2022-11-11T22:28:52.389000-05:00",
    "lastModifiedAt": "2022-11-11T22:28:52.389000-05:00",
    "tags": {}
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的取得測試套件定義。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSuiteDefinition。

get-suite-run-report

以下程式碼範例顯示如何使用 get-suite-run-report。

AWS CLI

若要取得 IoT Device Advisor 合格測試套件的相關資訊，請執行報告

下列get-suite-run-report範例取得成功裝置顧問合格測試套件執行的報告下載連結，其中包含指定的套件定義 ID 和套件執行 ID。

```
aws iotdeviceadvisor get-suite-run-report \
    --suite-definition-id ztvb5aek4w4x \
    --suite-run-id p6awv83nre6v
```

輸出：

```
{
    "qualificationReportDownloadUrl": "https://senate-apn-reports-us-east-1-
prod.s3.amazonaws.com/report.downloadlink"
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的取得成功執行資格測試套件的資格報告。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSuiteRunReport。

get-suite-run

以下程式碼範例顯示如何使用 get-suite-run。

AWS CLI

取得 IoT Device Advisor 測試套件執行狀態的相關資訊

下列get-suite-run範例會使用指定的套件定義 ID 和套件執行 ID，取得裝置建議程式測試套件執行狀態的相關資訊。

```
aws iotdeviceadvisor get-suite-run \
    --suite-definition-id qqcsmtyyjabl \
    --suite-run-id nzlfyhaa18oa
```

輸出：

```
{
    "suiteDefinitionId": "qqcsmtyyjabl",
    "suiteDefinitionVersion": "v1",
    "suiteRunId": "nzlfyhaa18oa",
    "suiteRunArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suiterun/
qqcsmtyyjabl/nzlfyhaa18oa",
    "suiteRunConfiguration": {
        "primaryDevice": {
            "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyIotThing",
            "certificateArn": "arn:aws:iot:us-east-1:123456789012:cert/certFile"
        },
        "parallelRun": false
    },
    "testResult": {
        "groups": [
            {
                "groupId": "uta5d9j1kvwc",
                "groupName": "Test group 1",
                "tests": [
                    {
                        "testCaseRunId": "2ve2twrqyr0s",
                        "testCaseDefinitionId": "awr8pq5vc9yp",
                        "testCaseDefinitionName": "MQTT Connect",
                        "status": "PASS",
                        "startTime": "2022-11-12T00:01:53.693000-05:00",
                        "endTime": "2022-11-12T00:02:15.443000-05:00",
                        "logUrl": "https://console.aws.amazon.com/
cloudwatch/home?region=us-east-1#logEventViewer:group=/aws/iot/deviceadvisor/
qqcsmtyyjabl;stream=nzlfyhaa18oa_2ve2twrqyr0s",
                        "warnings": "null",
```

```
                        "failure": "null"
                    }
                ]
            }
        ]
    },
    "startTime": "2022-11-12T00:01:52.673000-05:00",
    "endTime": "2022-11-12T00:02:16.496000-05:00",
    "status": "PASS",
    "tags": {}
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的取得測試套件執行。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSuiteRun。

`list-suite-definitions`

以下程式碼範例顯示如何使用 `list-suite-definitions`。

AWS CLI

範例 1：列出您建立的 IoT Device Advisor 測試套件

下列`list-suite-definitions`範例列出您在 AWS IoT 中建立的最多 25 個裝置建議程式測試套件。如果您有超過 25 個測試套件，則輸出中會顯示「nextToken」。您可以使用此 "nextToken" 來顯示您建立的其餘測試套件。

```
aws iotdeviceadvisor list-suite-definitions
```

輸出：

```
{
    "suiteDefinitionInformationList": [
        {
            "suiteDefinitionId": "3hsn88h4p2g5",
            "suiteDefinitionName": "TestSuite1",
            "defaultDevices": [
                {
                    "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/
MyIotThing"
                }
```

```
            ],
            "intendedForQualification": false,
            "isLongDurationTest": false,
            "protocol": "MqttV3_1_1",
            "createdAt": "2022-11-17T14:15:56.830000-05:00"
        },
        {
            ......
        }
    ],
    "nextToken": "nextTokenValue"
}
```

範例 2：列出您使用指定設定建立的 IoT Device Advisor 測試套件

下列`list-suite-definitions`範例列出您在 AWS IoT 中以指定最大結果編號建立的裝置建議程式測試套件。如果您有比最大數量更多的測試套件，則輸出中會顯示「nextToken」。如果您有 "nextToken"，您可以使用 "nextToken" 來顯示您建立的先前未顯示的測試套件。

```
aws iotdeviceadvisor list-suite-definitions \
    --max-result 1 \
    --next-token "nextTokenValue"
```

輸出：

```
{
    "suiteDefinitionInformationList": [
        {
            "suiteDefinitionId": "ztvb5aew4w4x",
            "suiteDefinitionName": "TestSuite2",
            "defaultDevices": [],
            "intendedForQualification": true,
            "isLongDurationTest": false,
            "protocol": "MqttV3_1_1",
            "createdAt": "2022-11-17T14:15:56.830000-05:00"
        }
    ],
    "nextToken": "nextTokenValue"
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 ListSuiteDefinitions。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSuiteDefinitions。

list-suite-runs

以下程式碼範例顯示如何使用 list-suite-runs。

AWS CLI

範例 1：列出指定 IoT Device Advisor 測試套件執行狀態的所有資訊

下列list-suite-runs範例列出裝置建議程式測試套件執行狀態的所有相關資訊，以及指定的套件定義 ID。如果您有超過 25 個測試套件執行，則輸出中會顯示「nextToken」。您可以使用此「nextToken」來顯示其餘的測試套件執行。

```
aws iotdeviceadvisor list-suite-runs \
    --suite-definition-id ztvb5aew4w4x
```

輸出：

```
{
    "suiteRunsList": [
        {
            "suiteDefinitionId": "ztvb5aew4w4x",
            "suiteDefinitionVersion": "v1",
            "suiteDefinitionName": "TestSuite",
            "suiteRunId": "p6awv89nre6v",
            "createdAt": "2022-12-01T16:33:14.212000-05:00",
            "startedAt": "2022-12-01T16:33:15.710000-05:00",
            "endAt": "2022-12-01T16:42:03.323000-05:00",
            "status": "PASS",
            "passed": 6,
            "failed": 0
        }
    ]
}
```

範例 2：列出具有指定設定之指定 IoT Device Advisor 測試套件執行狀態的相關資訊

下列list-suite-runs範例列出裝置建議程式測試套件執行狀態的相關資訊，其中包含指定的套件定義 ID 和指定的最大結果編號。如果您的測試套件執行次數超過上限，則輸出中會顯示「nextToken」。如果您有 "nextToken"，您可以使用 "nextToken" 來顯示之前未顯示的測試套件執行。

```
aws iotdeviceadvisor list-suite-runs \
```

```
    --suite-definition-id qqcsmtyyjaml \
    --max-result 1 \
    --next-token "nextTokenValue"
```

輸出：

```
{
    "suiteRunsList": [
        {
            "suiteDefinitionId": "qqcsmtyyjaml",
            "suiteDefinitionVersion": "v1",
            "suiteDefinitionName": "MQTT connection",
            "suiteRunId": "gz9vm2s6d2jy",
            "createdAt": "2022-12-01T20:10:27.079000-05:00",
            "startedAt": "2022-12-01T20:10:28.003000-05:00",
            "endAt": "2022-12-01T20:10:45.084000-05:00",
            "status": "STOPPED",
            "passed": 0,
            "failed": 0
        }
    ],
    "nextToken": "nextTokenValue"
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 ListSuiteRuns。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSuiteRuns。

list-tags-for-resource

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出連接至 IoT Device Advisor 資源的標籤

下列`list-tags-for-resource`範例列出連接至裝置建議程式資源的標籤。裝置建議程式資源可以是 Suitedefinition-Arn 或 Suiterun-Arn。

```
aws iotdeviceadvisor list-tags-for-resource \
    --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/ba0uyjpg38ny
```

輸出：

```
{
    "tags": {
        "TestTagKey": "TestTagValue"
    }
}
```

如需詳細資訊，請參閱AWS IoT API 參考》中的 ListTagsForResource，以及《服務授權參考》中的 AWS IoT Core Device Advisor 定義的資源類型。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`start-suite-run`

以下程式碼範例顯示如何使用 `start-suite-run`。

AWS CLI

啟動 IoT Device Advisor 測試套件執行

下列`start-suite-run`範例列出您 AWS 帳戶中可用的小工具。

```
aws iotdeviceadvisor start-suite-run \
    --suite-definition-id qqcsmtyyjabl \
    --suite-definition-version v1 \
    --suite-run-configuration '{"primaryDevice":{"thingArn": "arn:aws:iot:us-
east-1:123456789012:thing/MyIotThing","certificateArn":"arn:aws:iot:us-
east-1:123456789012:cert/certFile"}}'
```

輸出：

```
{
    "suiteRunId": "pwmucgw7lt9s",
    "suiteRunArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suiterun/
qqcsmtyyjabl/pwmucgw7lk9s",
    "createdAt": "2022-12-02T15:43:05.581000-05:00"
}
```

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的啟動測試套件執行。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartSuiteRun。

stop-suite-run

以下程式碼範例顯示如何使用 stop-suite-run。

AWS CLI

停止目前正在執行的 IoT Device Advisor 測試套件

下列stop-suite-run範例會停止目前使用指定套件定義 ID 和套件執行 ID 執行的裝置建議程式測試套件。

```
aws iotdeviceadvisor stop-suite-run \
    --suite-definition-id qqcsmtyyjabl \
    --suite-run-id nzlfyhaa18oa
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Core 開發人員指南中的停止測試套件執行。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopSuiteRun。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

新增和修改 IoT Device Advisor 資源的現有標籤

下列tag-resource範例會新增和修改具有指定資源 arn 和標籤之裝置建議程式資源的現有標籤。裝置建議程式資源可以是 Suitedefinition-Arn 或 Suiterun-Arn。

```
aws iotdeviceadvisor tag-resource \
    --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/
ba0uyjpg38ny \
    --tags '{"TagKey": "TagValue"}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT API 參考中的 TagResource，以及服務授權參考中的 AWS IoT Core Device Advisor 定義的資源類型。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

`untag-resource`

以下程式碼範例顯示如何使用 `untag-resource`。

AWS CLI

從 IoT Device Advisor 資源移除現有標籤

下列`untag-resource`範例會從具有指定資源 arn 和標籤索引鍵的裝置建議程式資源中移除現有標籤。裝置建議程式資源可以是 Suitedefinition-Arn 或 Suiterun-Arn。

```
aws iotdeviceadvisor untag-resource \
    --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/
ba0uyjpg38ny \
    --tag-keys "TagKey"
```

此命令不會產生輸出。

如需詳細資訊，請參閱AWS IoT API 參考》中的 UntagResource，以及《服務授權參考》中的 AWS IoT Core Device Advisor 定義的資源類型。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

`update-suite-definition`

以下程式碼範例顯示如何使用 `update-suite-definition`。

AWS CLI

範例 1：更新 IoT Device Advisor 測試套件

下列`update-suite-definition`範例會使用指定的套件定義 ID 和套件定義組態，更新 AWS IoT 中的裝置建議程式測試套件。

```
aws iotdeviceadvisor update-suite-definition \
    --suite-definition-id 3hsn88h4p2g5 \
    --suite-definition-configuration '{ \
        "suiteDefinitionName": "TestSuiteName", \
        "devices": [{"thingArn":"arn:aws:iot:us-east-1:123456789012:thing/
MyIotThing"}], \
```

```
        "intendedForQualification": false, \
        "rootGroup": "{\"configuration\":{},\"tests\":[{\"name\":\"MQTT Connect\",
\"configuration\":{\"EXECUTION_TIMEOUT\":120},\"tests\":[{\"name\":\"MQTT_Connect\",
\"configuration\":{},\"test\":{\"id\":\"MQTT_Connect\",\"testCase\":null,\"version
\":\"0.0.0\"}}]}]}", \
        "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole"}'
```

輸出：

```
{
    "suiteDefinitionId": "3hsn88h4p2g5",
    "suiteDefinitionName": "TestSuiteName",
    "suiteDefinitionVersion": "v3",
    "createdAt": "2022-11-17T14:15:56.830000-05:00",
    "lastUpdatedAt": "2022-12-02T16:02:45.857000-05:00"
}
```

範例 2：更新 IoT Device Advisor 資格測試套件

下列update-suite-definition範例會使用指定的套件定義 ID 和套件定義組態，更新 AWS IoT 中的裝置顧問資格測試套件。

```
aws iotdeviceadvisor update-suite-definition \
    --suite-definition-id txgsuolk2myj \
    --suite-definition-configuration '{
        "suiteDefinitionName": "TestSuiteName", \
        "devices": [{"thingArn":"arn:aws:iot:us-east-1:123456789012:thing/
MyIotThing"}], \
        "intendedForQualification": true, \
        "rootGroup": "", \
        "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole"}'
```

輸出：

```
{
    "suiteDefinitionId": "txgsuolk2myj",
    "suiteDefinitionName": "TestSuiteName",
    "suiteDefinitionVersion": "v3",
    "createdAt": "2022-11-17T14:15:56.830000-05:00",
    "lastUpdatedAt": "2022-12-02T16:02:45.857000-05:00"
}
```

如需詳細資訊，請參閱 AWS IoT API 參考中的 UpdateSuiteDefinition。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSuiteDefinition。

AWS IoT data 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS IoT data。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`delete-thing-shadow`

以下程式碼範例顯示如何使用 `delete-thing-shadow`。

AWS CLI

刪除裝置的影子文件

下列`delete-thing-shadow`範例會刪除名為 之裝置的整個影子文件`MyRPi`。

```
aws iot-data delete-thing-shadow \
    --thing-name MyRPi \
    "output.txt"
```

命令不會在顯示器上產生輸出，但`output.txt`包含確認您刪除之影子文件的版本和時間戳記的資訊。

```
{"version":2,"timestamp":1560270384}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的使用陰影。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteThingShadow。

get-thing-shadow

以下程式碼範例顯示如何使用 get-thing-shadow。

AWS CLI

取得物件影子文件

下列get-thing-shadow範例會取得指定 IoT 物件的物件影子文件。

```
aws iot-data get-thing-shadow \
    --thing-name MyRPi \
    output.txt
```

命令在顯示器上不會產生輸出，但以下顯示 的內容output.txt：

```
{
  "state":{
    "reported":{
    "moisture":"low"
    }
  },
  "metadata":{
    "reported":{
      "moisture":{
        "timestamp":1560269319
      }
    }
  },
  "version":1,"timestamp":1560269405
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Device Shadow Service Data Flow。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetThingShadow。

update-thing-shadow

以下程式碼範例顯示如何使用 update-thing-shadow。

AWS CLI

更新物件影子

下列update-thing-shadow範例會修改指定物件之裝置影子的目前狀態，並將其儲存至檔案output.txt。

```
aws iot-data update-thing-shadow \
    --thing-name MyRPi \
    --payload "{"state":{"reported":{"moisture":"okay"}}}" \
    "output.txt"
```

命令在顯示器上不會產生輸出，但以下顯示 的內容output.txt：

```
{
    "state": {
        "reported": {
            "moisture": "okay"
        }
    },
    "metadata": {
        "reported": {
            "moisture": {
                "timestamp": 1560270036
            }
        }
    },
    "version": 2,
    "timestamp": 1560270036
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Device Shadow Service Data Flow。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateThingShadow。

AWS IoT Events 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS IoT Events。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

batch-put-message

以下程式碼範例顯示如何使用 batch-put-message。

AWS CLI

傳送訊息（輸入）至 AWS IoT 事件

下列batch-put-message範例會將一組訊息傳送至 AWS IoT Events 系統。每個訊息承載都會轉換為您指定的輸入 inputName ()，並擷取到監控該輸入的任何偵測器。如果傳送多則訊息，無法保證訊息的處理順序。若要保證訂購，您必須一次傳送訊息一個，並等待成功回應。

```
aws iotevents-data batch-put-message \
    --cli-input-json file://highPressureMessage.json
```

highPressureMessage.json 的內容：

```
{
    "messages": [
        {
            "messageId": "00001",
            "inputName": "PressureInput",
            "payload": "{\"motorid\": \"Fulton-A32\", \"sensorData\": {\"pressure\":
 80, \"temperature\": 39} }"
        }
    ]
}
```

輸出：

```
{
    "BatchPutMessageErrorEntries": []
}
```

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 BatchPutMessage。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 BatchPutMessage。

batch-update-detector

以下程式碼範例顯示如何使用 batch-update-detector。

AWS CLI

更新偵測器（執行個體）

下列batch-update-detector範例會更新指定偵測器模型之一或多個偵測器（執行個體）的狀態、變數值和計時器設定。

```
aws iotevents-data batch-update-detector \
    --cli-input-json file://budFulton-A32.json
```

budFulton-A32.json 的內容：

```
{
    "detectors": [
        {
            "messageId": "00001",
            "detectorModelName": "motorDetectorModel",
            "keyValue": "Fulton-A32",
            "state": {
                "stateName": "Normal",
                "variables": [
                    {
                        "name": "pressureThresholdBreached",
                        "value": "0"
                    }
                ],
                "timers": [
                ]
            }
        }
    ]
}
```

輸出：

```
{
    "batchUpdateDetectorErrorEntries": []
}
```

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 BatchUpdateDetector。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchUpdateDetector。

create-detector-model

以下程式碼範例顯示如何使用 create-detector-model。

AWS CLI

建立偵測器模型

下列create-detector-model範例會使用參數檔案指定的組態來建立偵測器模型。

```
aws iotevents create-detector-model  \
    --cli-input-json file://motorDetectorModel.json
```

motorDetectorModel.json 的內容：

```
{
    "detectorModelName": "motorDetectorModel",
    "detectorModelDefinition": {
        "states": [
            {
                "stateName": "Normal",
                "onEnter": {
                    "events": [
                        {
                            "eventName": "init",
                            "condition": "true",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value": "0"
                                    }
                                }
                            ]
                        }
                    ]
                },
                "onInput": {
                    "transitionEvents": [
```

```
                        {
                            "eventName": "Overpressurized",
                            "condition": "$input.PressureInput.sensorData.pressure
 &gt; 70",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value":
 "$variable.pressureThresholdBreached + 3"
                                    }
                                }
                            ],
                            "nextState": "Dangerous"
                        }
                    ]
                }
            },
            {
                "stateName": "Dangerous",
                "onEnter": {
                    "events": [
                        {
                            "eventName": "Pressure Threshold Breached",
                            "condition": "$variable.pressureThresholdBreached &gt;
 1",
                            "actions": [
                                {
                                    "sns": {
                                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                                    }
                                }
                            ]
                        }
                    ]
                },
                "onInput": {
                    "events": [
                        {
                            "eventName": "Overpressurized",
                            "condition": "$input.PressureInput.sensorData.pressure
 &gt; 70",
                            "actions": [
```

```
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value": "3"
                                    }
                                }
                            ]
                        },
                        {
                            "eventName": "Pressure Okay",
                            "condition": "$input.PressureInput.sensorData.pressure
 &lt;= 70",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value":
 "$variable.pressureThresholdBreached - 1"
                                    }
                                }
                            ]
                        }
                    ],
                    "transitionEvents": [
                        {
                            "eventName": "BackToNormal",
                            "condition": "$input.PressureInput.sensorData.pressure
 &lt;= 70 &amp;&amp; $variable.pressureThresholdBreached &lt;= 1",
                            "nextState": "Normal"
                        }
                    ]
                },
                "onExit": {
                    "events": [
                        {
                            "eventName": "Normal Pressure Restored",
                            "condition": "true",
                            "actions": [
                                {
                                    "sns": {
                                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                                    }
                                }
```

```
                            ]
                        }
                    ]
                }
            }
        ],
        "initialStateName": "Normal"
    },
    "key": "motorid",
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
}
```

輸出：

```
{
    "detectorModelConfiguration": {
        "status": "ACTIVATING",
        "lastUpdateTime": 1560796816.077,
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "creationTime": 1560796816.077,
        "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/
motorDetectorModel",
        "key": "motorid",
        "detectorModelName": "motorDetectorModel",
        "detectorModelVersion": "1"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 CreateDetectorModel。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDetectorModel。

create-input

以下程式碼範例顯示如何使用 create-input。

AWS CLI

建立輸入

下列create-input範例會建立 輸入。

```
aws iotevents create-input  \
```

```
    --cli-input-json file://pressureInput.json
```

pressureInput.json 的內容：

```
{
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor",
    "inputDefinition": {
        "attributes": [
            { "jsonPath": "sensorData.pressure" },
            { "jsonPath": "motorid" }
        ]
    }
}
```

輸出：

```
{
    "inputConfiguration": {
        "status": "ACTIVE",
        "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
        "lastUpdateTime": 1560795312.542,
        "creationTime": 1560795312.542,
        "inputName": "PressureInput",
        "inputDescription": "Pressure readings from a motor"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 CreateInput。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateInput。

delete-detector-model

以下程式碼範例顯示如何使用 delete-detector-model。

AWS CLI

刪除偵測器模型

下列delete-detector-model範例會刪除指定的偵測器模型。偵測器模型的任何作用中執行個體也會遭到刪除。

```
aws iotevents delete-detector-model \
    --detector-model-name motorDetectorModel
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 DeleteDetectorModel。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDetectorModel。

delete-input

以下程式碼範例顯示如何使用 delete-input。

AWS CLI

刪除輸入

下列delete-input範例會刪除指定的輸入。

```
aws iotevents delete-input \
    --input-name PressureInput
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 DeleteInput。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteInput。

describe-detector-model

以下程式碼範例顯示如何使用 describe-detector-model。

AWS CLI

取得偵測器模型的相關資訊

下列describe-detector-model範例顯示指定偵測器模型的詳細資訊。由於未指定 version 參數，因此會傳回最新版本的相關資訊。

```
aws iotevents describe-detector-model \
    --detector-model-name motorDetectorModel
```

輸出：

```
{
    "detectorModel": {
        "detectorModelConfiguration": {
            "status": "ACTIVE",
            "lastUpdateTime": 1560796816.077,
            "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
            "creationTime": 1560796816.077,
            "detectorModelArn": "arn:aws:iotevents:us-
west-2:123456789012:detectorModel/motorDetectorModel",
            "key": "motorid",
            "detectorModelName": "motorDetectorModel",
            "detectorModelVersion": "1"
        },
        "detectorModelDefinition": {
            "states": [
                {
                    "onInput": {
                        "transitionEvents": [
                            {
                                "eventName": "Overpressurized",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
 "pressureThresholdBreached",
                                            "value":
 "$variable.pressureThresholdBreached + 3"
                                        }
                                    }
                                ],
                                "condition":
 "$input.PressureInput.sensorData.pressure > 70",
                                "nextState": "Dangerous"
                            }
                        ],
                        "events": []
                    },
                    "stateName": "Normal",
                    "onEnter": {
                        "events": [
                            {
                                "eventName": "init",
```

```
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
"pressureThresholdBreached",
                                            "value": "0"
                                        }
                                    }
                                ],
                                "condition": "true"
                            }
                        ]
                    },
                    "onExit": {
                        "events": []
                    }
                },
                {
                    "onInput": {
                        "transitionEvents": [
                            {
                                "eventName": "BackToNormal",
                                "actions": [],
                                "condition":
"$input.PressureInput.sensorData.pressure <= 70 &&
$variable.pressureThresholdBreached <= 1",
                                "nextState": "Normal"
                            }
                        ],
                        "events": [
                            {
                                "eventName": "Overpressurized",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
"pressureThresholdBreached",
                                            "value": "3"
                                        }
                                    }
                                ],
                                "condition":
"$input.PressureInput.sensorData.pressure > 70"
                            },
```

```
                            {
                                "eventName": "Pressure Okay",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
 "pressureThresholdBreached",
                                            "value":
 "$variable.pressureThresholdBreached - 1"
                                        }
                                    }
                                ],
                                "condition":
 "$input.PressureInput.sensorData.pressure <= 70"
                            }
                        ]
                    },
                    "stateName": "Dangerous",
                    "onEnter": {
                        "events": [
                            {
                                "eventName": "Pressure Threshold Breached",
                                "actions": [
                                    {
                                        "sns": {
                                            "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                                        }
                                    }
                                ],
                                "condition": "$variable.pressureThresholdBreached >
 1"
                            }
                        ]
                    },
                    "onExit": {
                        "events": [
                            {
                                "eventName": "Normal Pressure Restored",
                                "actions": [
                                    {
                                        "sns": {
                                            "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
```

```
                                    }
                                }
                            ],
                            "condition": "true"
                        }
                    ]
                }
            }
        ],
        "initialStateName": "Normal"
    }
  }
}
```

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 DescribeDetectorModel。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDetectorModel。

describe-detector

以下程式碼範例顯示如何使用 describe-detector。

AWS CLI

取得偵測器（執行個體）的相關資訊。

下列describe-detector範例顯示指定偵測器（執行個體）的詳細資訊。

```
aws iotevents-data describe-detector \
    --detector-model-name motorDetectorModel \
    --key-value "Fulton-A32"
```

輸出：

```
{
    "detector": {
        "lastUpdateTime": 1560797852.776,
        "creationTime": 1560797852.775,
        "state": {
            "variables": [
                {
                    "name": "pressureThresholdBreached",
```

```
                    "value": "3"
                }
            ],
            "stateName": "Dangerous",
            "timers": []
        },
        "keyValue": "Fulton-A32",
        "detectorModelName": "motorDetectorModel",
        "detectorModelVersion": "1"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 DescribeDetector。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDetector。

describe-input

以下程式碼範例顯示如何使用 describe-input。

AWS CLI

取得輸入的相關資訊

下列describe-input範例顯示指定輸入的詳細資訊。

```
aws iotevents describe-input \
    --input-name PressureInput
```

輸出：

```
{
    "input": {
        "inputConfiguration": {
            "status": "ACTIVE",
            "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/
PressureInput",
            "lastUpdateTime": 1560795312.542,
            "creationTime": 1560795312.542,
            "inputName": "PressureInput",
            "inputDescription": "Pressure readings from a motor"
        },
```

```
        "inputDefinition": {
            "attributes": [
                {
                    "jsonPath": "sensorData.pressure"
                },
                {
                    "jsonPath": "motorid"
                }
            ]
        }
    }
}
```

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 DescribeInput。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInput。

describe-logging-options

以下程式碼範例顯示如何使用 `describe-logging-options`。

AWS CLI

取得記錄設定的相關資訊

下列`describe-logging-options`範例會擷取 AWS IoT Events 記錄選項的目前設定。

```
aws iotevents describe-logging-options
```

輸出：

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "enabled": false,
        "level": "ERROR"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 DescribeLoggingOptions。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLoggingOptions。

list-detector-model-versions

以下程式碼範例顯示如何使用 list-detector-model-versions。

AWS CLI

取得偵測器模型版本的相關資訊

下列list-detector-model-versions範例列出偵測器模型的所有版本。只會傳回與每個偵測器模型版本相關聯的中繼資料。

```
aws iotevents list-detector-model-versions \
    --detector-model-name motorDetectorModel
```

輸出：

```
{
    "detectorModelVersionSummaries": [
        {
            "status": "ACTIVE",
            "lastUpdateTime": 1560796816.077,
            "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
            "creationTime": 1560796816.077,
            "detectorModelArn": "arn:aws:iotevents:us-
west-2:123456789012:detectorModel/motorDetectorModel",
            "detectorModelName": "motorDetectorModel",
            "detectorModelVersion": "1"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 ListDetectorModelVersions。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDetectorModelVersions。

list-detector-models

以下程式碼範例顯示如何使用 list-detector-models。

AWS CLI

若要取得偵測器模型的清單

下列list-detector-models範例列出您已建立的偵測器模型。只會傳回與每個偵測器模型相關聯的中繼資料。

```
aws iotevents list-detector-models
```

輸出：

```
{
    "detectorModelSummaries": [
        {
            "detectorModelName": "motorDetectorModel",
            "creationTime": 1552072424.212
            "detectorModelDescription": "Detect overpressure in a motor."
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 ListDetectorModels。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDetectorModels。

list-detectors

以下程式碼範例顯示如何使用 list-detectors。

AWS CLI

取得偵測器模型的偵測器清單

下列list-detectors範例列出您帳戶中的偵測器（偵測器模型的執行個體）。

```
aws iotevents-data list-detectors \
    --detector-model-name motorDetectorModel
```

輸出：

```
{
    "detectorSummaries": [
        {
            "lastUpdateTime": 1558129925.2,
            "creationTime": 1552073155.527,
```

```
            "state": {
                "stateName": "Normal"
            },
            "keyValue": "Fulton-A32",
            "detectorModelName": "motorDetectorModel",
            "detectorModelVersion": "1"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 ListDetectors。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDetectors。

`list-inputs`

以下程式碼範例顯示如何使用 `list-inputs`。

AWS CLI

列出輸入

下列`list-inputs`範例列出您在帳戶中建立的輸入。

```
aws iotevents list-inputs
```

此命令不會產生輸出。輸出：

```
{
    {
        "status": "ACTIVE",
        "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
        "lastUpdateTime": 1551742986.768,
        "creationTime": 1551742986.768,
        "inputName": "PressureInput",
        "inputDescription": "Pressure readings from a motor"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 ListInputs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListInputs。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出指派給資源的標籤。

下列list-tags-for-resource範例列出您指派給資源的標籤索引鍵名稱和值。

```
aws iotevents list-tags-for-resource \
    --resource-arn "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput"
```

輸出：

```
{
    "tags": [
        {
            "value": "motor",
            "key": "deviceType"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 ListTagsForResource。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

put-logging-options

以下程式碼範例顯示如何使用 put-logging-options。

AWS CLI

設定記錄選項

下列put-logging-options範例會設定或更新 AWS IoT Events 記錄選項。如果您更新任何loggingOptions` field, it can take up to one minute for the change to take effect. Also, if you change the policy attached to the role you specified in the ``roleArn欄位的值（例如，更正無效的政策），該變更最多可能需要五分鐘才會生效。

```
aws iotevents put-logging-options \
    --cli-input-json file://logging-options.json
```

logging-options.json 的內容：

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "level": "DEBUG",
        "enabled": true,
        "detectorDebugOptions": [
            {
                "detectorModelName": "motorDetectorModel",
                "keyValue": "Fulton-A32"
            }
        ]
    }
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 PutLoggingOptions。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutLoggingOptions。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤新增至資源

下列tag-resource範例會新增或修改（如果金鑰deviceType已存在）連接指定資源的標籤。

```
aws iotevents tag-resource \
    --cli-input-json file://pressureInput.tag.json
```

pressureInput.tag.json 的內容：

```
{
    "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
```

```
    "tags": [
        {
            "key": "deviceType",
            "value": "motor"
        }
    ]
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 TagResource。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源移除標籤

下列untag-resource範例會從指定的資源移除具有指定金鑰名稱的標籤。

```
aws iotevents untag-resource \
    --resource-arn arn:aws:iotevents:us-west-2:123456789012:input/PressureInput \
    --tagkeys deviceType
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 UntagResource。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-detector-model

以下程式碼範例顯示如何使用 update-detector-model。

AWS CLI

更新偵測器模型

下列update-detector-model範例會更新指定的偵測器模型。先前版本產生的偵測器（執行個體）會遭到刪除，然後在新輸入到達時重新建立。

```
aws iotevents update-detector-model \
    --cli-input-json file://motorDetectorModel.update.json
```

motorDetectorModel.update.json 的內容：

```
{
    "detectorModelName": "motorDetectorModel",
    "detectorModelDefinition": {
        "states": [
            {
                "stateName": "Normal",
                "onEnter": {
                    "events": [
                        {
                            "eventName": "init",
                            "condition": "true",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value": "0"
                                    }
                                }
                            ]
                        }
                    ]
                },
                "onInput": {
                    "transitionEvents": [
                        {
                            "eventName": "Overpressurized",
                            "condition": "$input.PressureInput.sensorData.pressure >
 70",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value":
 "$variable.pressureThresholdBreached + 3"
                                    }
                                }
                            ],
                            "nextState": "Dangerous"
```

```
                        }
                    ]
                }
            },
            {
                "stateName": "Dangerous",
                "onEnter": {
                    "events": [
                        {
                            "eventName": "Pressure Threshold Breached",
                            "condition": "$variable.pressureThresholdBreached > 1",
                            "actions": [
                                {
                                    "sns": {
                                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                                    }
                                }
                            ]
                        }
                    ]
                },
                "onInput": {
                    "events": [
                        {
                            "eventName": "Overpressurized",
                            "condition": "$input.PressureInput.sensorData.pressure >
 70",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value": "3"
                                    }
                                }
                            ]
                        },
                        {
                            "eventName": "Pressure Okay",
                            "condition": "$input.PressureInput.sensorData.pressure
 <= 70",
                            "actions": [
                                {
                                    "setVariable": {
```

```
                                        "variableName": "pressureThresholdBreached",
                                        "value":
 "$variable.pressureThresholdBreached - 1"
                                    }
                                }
                            ]
                        }
                    ],
                    "transitionEvents": [
                        {
                            "eventName": "BackToNormal",
                            "condition": "$input.PressureInput.sensorData.pressure
 <= 70 && $variable.pressureThresholdBreached <= 1",
                            "nextState": "Normal"
                        }
                    ]
                },
                "onExit": {
                    "events": [
                        {
                            "eventName": "Normal Pressure Restored",
                            "condition": "true",
                            "actions": [
                                {
                                    "sns": {
                                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                                    }
                                }
                            ]
                        }
                    ]
                }
            }
        ],
        "initialStateName": "Normal"
    },
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
}
```

輸出：

```
{
```

```
    "detectorModelConfiguration": {
        "status": "ACTIVATING",
        "lastUpdateTime": 1560799387.719,
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "creationTime": 1560799387.719,
        "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/
motorDetectorModel",
        "key": "motorid",
        "detectorModelName": "motorDetectorModel",
        "detectorModelVersion": "2"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 UpdateDetectorModel。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDetectorModel。

update-input

以下程式碼範例顯示如何使用 update-input。

AWS CLI

更新輸入

下列update-input範例會使用新的描述和定義更新指定的輸入。

```
aws iotevents update-input \
    --cli-input-json file://pressureInput.json
```

pressureInput.json 的內容：

```
{
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor",
    "inputDefinition": {
        "attributes": [
            { "jsonPath": "sensorData.pressure" },
            { "jsonPath": "motorid" }
        ]
    }
}
```

輸出：

```
{
    "inputConfiguration": {
        "status": "ACTIVE",
        "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
        "lastUpdateTime": 1560795976.458,
        "creationTime": 1560795312.542,
        "inputName": "PressureInput",
        "inputDescription": "Pressure readings from a motor"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Events API 參考中的 UpdateInput。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateInput。

AWS IoT Events-Data 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS IoT Events-Data。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`batch-put-message`

以下程式碼範例顯示如何使用 `batch-put-message`。

AWS CLI

傳送訊息（輸入）至 AWS IoT 事件

下列batch-put-message範例會將一組訊息傳送至 AWS IoT Events 系統。每個訊息承載都會轉換為您指定的輸入 inputName ()，並擷取到監控該輸入的任何偵測器。如果傳送多則訊息，無法保證訊息的處理順序。若要保證訂購，您必須一次傳送訊息一個，並等待成功回應。

```
aws iotevents-data batch-put-message \
    --cli-binary-format raw-in-base64-out \
    --cli-input-json file://highPressureMessage.json
```

highPressureMessage.json 的內容：

```
{
    "messages": [
        {
            "messageId": "00001",
            "inputName": "PressureInput",
            "payload": "{\"motorid\": \"Fulton-A32\", \"sensorData\": {\"pressure\":
 80, \"temperature\": 39} }"
        }
    ]
}
```

輸出：

```
{
    "BatchPutMessageErrorEntries": []
}
```

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 BatchPutMessage。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchPutMessage。

batch-update-detector

以下程式碼範例顯示如何使用 batch-update-detector。

AWS CLI

更新偵測器 （ 執行個體 ）

下列batch-update-detector範例會更新指定偵測器模型之一或多個偵測器 （ 執行個體 ） 的狀態、變數值和計時器設定。

```
aws iotevents-data batch-update-detector \
    --cli-input-json file://budFulton-A32.json
```

budFulton-A32.json 的內容：

```
{
    "detectors": [
        {
            "messageId": "00001",
            "detectorModelName": "motorDetectorModel",
            "keyValue": "Fulton-A32",
            "state": {
                "stateName": "Normal",
                "variables": [
                    {
                        "name": "pressureThresholdBreached",
                        "value": "0"
                    }
                ],
                "timers": [
                ]
            }
        }
    ]
}
```

輸出：

```
{
    "batchUpdateDetectorErrorEntries": []
}
```

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 BatchUpdateDetector。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchUpdateDetector。

create-detector-model

以下程式碼範例顯示如何使用 create-detector-model。

AWS CLI

建立偵測器模型

下列create-detector-model範例會建立偵測器模型。

```
aws iotevents create-detector-model \
    --cli-input-json file://motorDetectorModel.json
```

motorDetectorModel.json 的內容：

```
{
    "detectorModelName": "motorDetectorModel",
    "detectorModelDefinition": {
        "states": [
            {
                "stateName": "Normal",
                "onEnter": {
                    "events": [
                        {
                            "eventName": "init",
                            "condition": "true",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value": "0"
                                    }
                                }
                            ]
                        }
                    ]
                },
                "onInput": {
                    "transitionEvents": [
                        {
                            "eventName": "Overpressurized",
                            "condition": "$input.PressureInput.sensorData.pressure
 &gt; 70",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
```

```
                                        "value":
 "$variable.pressureThresholdBreached + 3"
                                    }
                                }
                            ],
                            "nextState": "Dangerous"
                        }
                    ]
                }
            },
            {
                "stateName": "Dangerous",
                "onEnter": {
                    "events": [
                        {
                            "eventName": "Pressure Threshold Breached",
                            "condition": "$variable.pressureThresholdBreached &gt;
 1",
                            "actions": [
                                {
                                    "sns": {
                                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                                    }
                                }
                            ]
                        }
                    ]
                },
                "onInput": {
                    "events": [
                        {
                            "eventName": "Overpressurized",
                            "condition": "$input.PressureInput.sensorData.pressure
 &gt; 70",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value": "3"
                                    }
                                }
                            ]
                        },
```

```
                        {
                            "eventName": "Pressure Okay",
                            "condition": "$input.PressureInput.sensorData.pressure
 &lt;= 70",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value":
 "$variable.pressureThresholdBreached - 1"
                                    }
                                }
                            ]
                        }
                    ],
                    "transitionEvents": [
                        {
                            "eventName": "BackToNormal",
                            "condition": "$input.PressureInput.sensorData.pressure
 &lt;= 70 &amp;&amp; $variable.pressureThresholdBreached &lt;= 1",
                            "nextState": "Normal"
                        }
                    ]
                },
                "onExit": {
                    "events": [
                        {
                            "eventName": "Normal Pressure Restored",
                            "condition": "true",
                            "actions": [
                                {
                                    "sns": {
                                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                                    }
                                }
                            ]
                        }
                    ]
                }
            }
        ],
        "initialStateName": "Normal"
    },
```

```
    "key": "motorid",
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
}
```

輸出：

```
{
    "detectorModelConfiguration": {
        "status": "ACTIVATING",
        "lastUpdateTime": 1560796816.077,
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "creationTime": 1560796816.077,
        "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/
motorDetectorModel",
        "key": "motorid",
        "detectorModelName": "motorDetectorModel",
        "detectorModelVersion": "1"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 CreateDetectorModel。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDetectorModel。

create-input

以下程式碼範例顯示如何使用 create-input。

AWS CLI

建立輸入

下列create-input範例會建立 輸入。

```
aws iotevents create-input \
    --cli-input-json file://pressureInput.json
```

pressureInput.json 的內容：

```
{
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor",
    "inputDefinition": {
```

```
        "attributes": [
                { "jsonPath": "sensorData.pressure" },
                { "jsonPath": "motorid" }
            ]
    }
}
```

輸出：

```
{
    "inputConfiguration": {
        "status": "ACTIVE",
        "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
        "lastUpdateTime": 1560795312.542,
        "creationTime": 1560795312.542,
        "inputName": "PressureInput",
        "inputDescription": "Pressure readings from a motor"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 CreateInput。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateInput。

delete-detector-model

以下程式碼範例顯示如何使用 delete-detector-model。

AWS CLI

刪除偵測器模型

下列delete-detector-model範例會刪除偵測器模型。偵測器模型的任何作用中執行個體也會遭到刪除。

```
aws iotevents delete-detector-model \
    --detector-model-name motorDetectorModel*
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 DeleteDetectorModel。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDetectorModel。

delete-input

以下程式碼範例顯示如何使用 delete-input。

AWS CLI

刪除輸入

下列delete-input範例會刪除輸入。

```
aws iotevents delete-input \
    --input-name PressureInput
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 DeleteInput。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteInput。

describe-detector-model

以下程式碼範例顯示如何使用 describe-detector-model。

AWS CLI

取得偵測器模型的相關資訊

下列describe-detector-model範例說明偵測器模型。如果未指定 version 參數，命令會傳回最新版本的相關資訊。

```
aws iotevents describe-detector-model \
    --detector-model-name motorDetectorModel
```

輸出：

```
{
    "detectorModel": {
        "detectorModelConfiguration": {
            "status": "ACTIVE",
            "lastUpdateTime": 1560796816.077,
            "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
            "creationTime": 1560796816.077,
```

```
            "detectorModelArn": "arn:aws:iotevents:us-
west-2:123456789012:detectorModel/motorDetectorModel",
            "key": "motorid",
            "detectorModelName": "motorDetectorModel",
            "detectorModelVersion": "1"
        },
        "detectorModelDefinition": {
            "states": [
                {
                    "onInput": {
                        "transitionEvents": [
                            {
                                "eventName": "Overpressurized",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
 "pressureThresholdBreached",
                                            "value":
 "$variable.pressureThresholdBreached + 3"
                                        }
                                    }
                                ],
                                "condition":
 "$input.PressureInput.sensorData.pressure > 70",
                                "nextState": "Dangerous"
                            }
                        ],
                        "events": []
                    },
                    "stateName": "Normal",
                    "onEnter": {
                        "events": [
                            {
                                "eventName": "init",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
 "pressureThresholdBreached",
                                            "value": "0"
                                        }
                                    }
                                ],
```

```
                                "condition": "true"
                            }
                        ]
                    },
                    "onExit": {
                        "events": []
                    }
                },
                {
                    "onInput": {
                        "transitionEvents": [
                            {
                                "eventName": "BackToNormal",
                                "actions": [],
                                "condition":
"$input.PressureInput.sensorData.pressure <= 70 &&
$variable.pressureThresholdBreached <= 1",
                                "nextState": "Normal"
                            }
                        ],
                        "events": [
                            {
                                "eventName": "Overpressurized",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
"pressureThresholdBreached",
                                            "value": "3"
                                        }
                                    }
                                ],
                                "condition":
"$input.PressureInput.sensorData.pressure > 70"
                            },
                            {
                                "eventName": "Pressure Okay",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
"pressureThresholdBreached",
                                            "value":
"$variable.pressureThresholdBreached - 1"
```

```
                                        }
                                    }
                                ],
                                "condition":
 "$input.PressureInput.sensorData.pressure <= 70"
                            }
                        ]
                    },
                    "stateName": "Dangerous",
                    "onEnter": {
                        "events": [
                            {
                                "eventName": "Pressure Threshold Breached",
                                "actions": [
                                    {
                                        "sns": {
                                            "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                                        }
                                    }
                                ],
                                "condition": "$variable.pressureThresholdBreached >
 1"
                            }
                        ]
                    },
                    "onExit": {
                        "events": [
                            {
                                "eventName": "Normal Pressure Restored",
                                "actions": [
                                    {
                                        "sns": {
                                            "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                                        }
                                    }
                                ],
                                "condition": "true"
                            }
                        ]
                    }
                }
            ],
```

```
            "initialStateName": "Normal"
        }
    }
}
```

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 DescribeDetectorModel。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDetectorModel。

describe-detector

以下程式碼範例顯示如何使用 describe-detector。

AWS CLI

取得偵測器的相關資訊 （ 執行個體 ）

下列describe-detector範例會傳回指定偵測器 （ 執行個體 ） 的相關資訊。

```
aws iotevents-data describe-detector \
    --detector-model-name motorDetectorModel \
    --key-value "Fulton-A32"
```

輸出：

```
{
    "detector": {
        "lastUpdateTime": 1560797852.776,
        "creationTime": 1560797852.775,
        "state": {
            "variables": [
                {
                    "name": "pressureThresholdBreached",
                    "value": "3"
                }
            ],
            "stateName": "Dangerous",
            "timers": []
        },
        "keyValue": "Fulton-A32",
        "detectorModelName": "motorDetectorModel",
        "detectorModelVersion": "1"
    }
```

```
}
```

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 DescribeDetector。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDetector。

describe-input

以下程式碼範例顯示如何使用 describe-input。

AWS CLI

取得輸入的相關資訊

下列describe-input範例會擷取輸入的詳細資訊。

```
aws iotevents describe-input \
    --input-name PressureInput
```

輸出：

```
{
    "input": {
        "inputConfiguration": {
            "status": "ACTIVE",
            "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/
PressureInput",
            "lastUpdateTime": 1560795312.542,
            "creationTime": 1560795312.542,
            "inputName": "PressureInput",
            "inputDescription": "Pressure readings from a motor"
        },
        "inputDefinition": {
            "attributes": [
                {
                    "jsonPath": "sensorData.pressure"
                },
                {
                    "jsonPath": "motorid"
                }
            ]
        }
    }
```

```
}
```

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 DescribeInput。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInput。

describe-logging-options

以下程式碼範例顯示如何使用 describe-logging-options。

AWS CLI

取得記錄設定的相關資訊

下列describe-logging-options範例會擷取目前的 AWS IoT Events 記錄選項。

```
aws iotevents describe-logging-options
```

輸出：

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "enabled": false,
        "level": "ERROR"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 DescribeLoggingOptions。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLoggingOptions。

list-detector-model-versions

以下程式碼範例顯示如何使用 list-detector-model-versions。

AWS CLI

取得偵測器模型版本的相關資訊

下列list-detector-model-versions範例列出偵測器模型的所有版本。只會傳回與每個偵測器模型版本相關聯的中繼資料。

```
aws iotevents list-detector-model-versions \
    --detector-model-name motorDetectorModel
```

輸出：

```
{
    "detectorModelVersionSummaries": [
        {
            "status": "ACTIVE",
            "lastUpdateTime": 1560796816.077,
            "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
            "creationTime": 1560796816.077,
            "detectorModelArn": "arn:aws:iotevents:us-
west-2:123456789012:detectorModel/motorDetectorModel",
            "detectorModelName": "motorDetectorModel",
            "detectorModelVersion": "1"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 ListDetectorModelVersions。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDetectorModelVersions。

list-detector-models

以下程式碼範例顯示如何使用 list-detector-models。

AWS CLI

取得偵測器模型的清單

下列list-detector-models範例列出您已建立的偵測器模型。只會傳回與每個偵測器模型相關聯的中繼資料。

```
aws iotevents list-detector-models
```

輸出：

```
{
    "detectorModelSummaries": [
        {
```

```
            "detectorModelName": "motorDetectorModel",
            "creationTime": 1552072424.212
            "detectorModelDescription": "Detect overpressure in a motor."
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 ListDetectorModels。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDetectorModels。

`list-detectors`

以下程式碼範例顯示如何使用 `list-detectors`。

AWS CLI

取得偵測器模型的偵測器清單

下列`list-detectors`範例列出偵測器 （ 偵測器模型的執行個體 ）。

```
aws iotevents-data list-detectors \
    --detector-model-name motorDetectorModel
```

輸出：

```
{
    "detectorSummaries": [
        {
            "lastUpdateTime": 1558129925.2,
            "creationTime": 1552073155.527,
            "state": {
                "stateName": "Normal"
            },
            "keyValue": "Fulton-A32",
            "detectorModelName": "motorDetectorModel",
            "detectorModelVersion": "1"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 ListDetectors。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDetectors。

list-inputs

以下程式碼範例顯示如何使用 list-inputs。

AWS CLI

列出輸入

下列list-inputs範例列出您已建立的輸入。

```
aws iotevents list-inputs
```

輸出：

```
{
    "status": "ACTIVE",
    "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
    "lastUpdateTime": 1551742986.768,
    "creationTime": 1551742986.768,
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor"
}
```

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 ListInputs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListInputs。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出指派給資源的標籤

下列list-tags-for-resource範例列出您已指派給資源的標籤（中繼資料）。

```
aws iotevents list-tags-for-resource \
    --resource-arn "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput"
```

輸出：

```
{
    "tags": [
        {
            "value": "motor",
            "key": "deviceType"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 ListTagsForResource。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

put-logging-options

以下程式碼範例顯示如何使用 put-logging-options。

AWS CLI

設定記錄選項

下列list-tags-for-resource範例會設定或更新 AWS IoT Events 記錄選項。如果您更新任何loggingOptions欄位的值，變更最多需要一分鐘才會生效。此外，如果您變更連接到您在 roleArn 欄位中指定之角色的政策（例如，更正無效的政策），該變更最多需要五分鐘才會生效。

```
aws iotevents put-logging-options \
    --cli-input-json file://logging-options.json
```

logging-options.json 的內容：

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "level": "DEBUG",
        "enabled": true,
        "detectorDebugOptions": [
            {
                "detectorModelName": "motorDetectorModel",
```

```
                "keyValue": "Fulton-A32"
            }
        ]
    }
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 PutLoggingOptions。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutLoggingOptions。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤新增至資源

下列tag-resource範例會新增或修改指定資源的標籤。標籤是可用於管理資源的中繼資料。

```
aws iotevents tag-resource \
    --cli-input-json file://pressureInput.tag.json
```

pressureInput.tag.json 的內容：

```
{
    "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
    "tags": [
        {
            "key": "deviceType",
            "value": "motor"
        }
    ]
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 TagResource。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源移除標籤

下列untag-resource範例會從 資源移除指定的標籤。

```
aws iotevents untag-resource \
    --cli-input-json file://pressureInput.untag.json
```

pressureInput.untag.json 的內容：

```
{
    "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
    "tagKeys": [
            "deviceType"
    ]
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 UntagResource。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-detector-model

以下程式碼範例顯示如何使用 update-detector-model。

AWS CLI

更新偵測器模型

下列update-detector-model範例會更新偵測器模型。先前版本產生的偵測器（執行個體）會遭到刪除，然後在新輸入到達時重新建立。

```
aws iotevents update-detector-model \
    --cli-input-json file://motorDetectorModel.update.json
```

motorDetectorModel.update.json:

```
{
  "detectorModelName": "motorDetectorModel",
  "detectorModelDefinition": {
    "states": [
      {
        "stateName": "Normal",
        "onEnter": {
          "events": [
            {
              "eventName": "init",
              "condition": "true",
              "actions": [
                {
                  "setVariable": {
                    "variableName": "pressureThresholdBreached",
                    "value": "0"
                  }
                }
              ]
            }
          ]
        },
        "onInput": {
          "transitionEvents": [
            {
              "eventName": "Overpressurized",
              "condition": "$input.PressureInput.sensorData.pressure > 70",
              "actions": [
                {
                  "setVariable": {
                    "variableName": "pressureThresholdBreached",
                    "value": "$variable.pressureThresholdBreached + 3"
                  }
                }
              ],
              "nextState": "Dangerous"
            }
          ]
        }
      },
      {
        "stateName": "Dangerous",
```

```
        "onEnter": {
          "events": [
            {
              "eventName": "Pressure Threshold Breached",
              "condition": "$variable.pressureThresholdBreached > 1",
              "actions": [
                {
                  "sns": {
                    "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                  }
                }
              ]
            }
          ]
        },
        "onInput": {
          "events": [
            {
              "eventName": "Overpressurized",
              "condition": "$input.PressureInput.sensorData.pressure > 70",
              "actions": [
                {
                  "setVariable": {
                    "variableName": "pressureThresholdBreached",
                    "value": "3"
                  }
                }
              ]
            },
            {
              "eventName": "Pressure Okay",
              "condition": "$input.PressureInput.sensorData.pressure <= 70",
              "actions": [
                {
                  "setVariable": {
                    "variableName": "pressureThresholdBreached",
                    "value": "$variable.pressureThresholdBreached - 1"
                  }
                }
              ]
            }
          ],
          "transitionEvents": [
```

```
            {
              "eventName": "BackToNormal",
              "condition": "$input.PressureInput.sensorData.pressure <= 70 &&
 $variable.pressureThresholdBreached <= 1",
              "nextState": "Normal"
            }
          ]
        },
        "onExit": {
          "events": [
            {
              "eventName": "Normal Pressure Restored",
              "condition": "true",
              "actions": [
                {
                  "sns": {
                    "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                  }
                }
              ]
            }
          ]
        }
      }
    ],
    "initialStateName": "Normal"
  },
  "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
}
```

輸出：

```
{
    "detectorModelConfiguration": {
        "status": "ACTIVATING",
        "lastUpdateTime": 1560799387.719,
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "creationTime": 1560799387.719,
        "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/
motorDetectorModel",
        "key": "motorid",
        "detectorModelName": "motorDetectorModel",
```

```
        "detectorModelVersion": "2"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 UpdateDetectorModel。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDetectorModel。

update-input

以下程式碼範例顯示如何使用 update-input。

AWS CLI

更新輸入

下列update-input範例會更新 輸入。

```
aws iotevents update-input \
    --cli-input-json file://pressureInput.json
```

pressureInput.json 的內容：

```
{
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor",
    "inputDefinition": {
        "attributes": [
            { "jsonPath": "sensorData.pressure" },
            { "jsonPath": "motorid" }
        ]
    }
}
```

輸出：

```
{
    "inputConfiguration": {
        "status": "ACTIVE",
        "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
        "lastUpdateTime": 1560795976.458,
```

```
        "creationTime": 1560795312.542,
        "inputName": "PressureInput",
        "inputDescription": "Pressure readings from a motor"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Events 開發人員指南*中的 UpdateInput。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateInput。

AWS IoT Greengrass 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS IoT Greengrass。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-role-to-group`

以下程式碼範例顯示如何使用 `associate-role-to-group`。

AWS CLI

將角色與 Greengrass 群組建立關聯

下列`associate-role-to-group`範例會將指定的 IAM 角色與 Greengrass 群組建立關聯。本機 Lambda 函數和連接器會使用群組角色來存取 AWS 服務。例如，您的群組角色可能會授予 CloudWatch Logs 整合所需的許可。

```
aws greengrass associate-role-to-group \
    --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b \
```

```
    --role-arn arn:aws:iam::123456789012:role/GG-Group-Role
```

輸出：

```
{
    "AssociatedAt": "2019-09-10T20:03:30Z"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的設定群組角色。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateRoleToGroup。

associate-service-role-to-account

以下程式碼範例顯示如何使用 associate-service-role-to-account。

AWS CLI

將服務角色與 AWS 您的帳戶建立關聯

下列associate-service-role-to-account範例會將 ARN 指定的 IAM 服務角色與 AWS 帳戶中的 AWS IoT Greengrass 建立關聯。您必須先前已在 IAM 中建立服務角色，而且必須將政策文件與允許 AWS IoT Greengrass 擔任此角色的政策文件建立關聯。

```
aws greengrass associate-service-role-to-account \
    --role-arn "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
```

輸出：

```
{
    "AssociatedAt": "2019-06-25T18:12:45Z"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的 Greengrass 服務角色。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateServiceRoleToAccount。

create-connector-definition-version

以下程式碼範例顯示如何使用 create-connector-definition-version。

AWS CLI

建立連接器定義版本

下列create-connector-definition-version範例會建立連接器定義版本，並將其與指定的連接器定義建立關聯。版本中的所有連接器都會為其參數定義值。

```
aws greengrass create-connector-definition-version \
    --connector-definition-id "55d0052b-0d7d-44d6-b56f-21867215e118" \
    --connectors "[{\"Id\": \"MyTwilioNotificationsConnector\",
 \"ConnectorArn\": \"arn:aws:greengrass:us-west-2::/connectors/
TwilioNotifications/versions/2\", \"Parameters\": {\"TWILIO_ACCOUNT_SID
\": \"AC1a8d4204890840d7fc482aab38090d57\", \"TwilioAuthTokenSecretArn\":
 \"arn:aws:secretsmanager:us-west-2:123456789012:secret:greengrass-TwilioAuthToken-
ntSlp6\", \"TwilioAuthTokenSecretArn-ResourceId\": \"TwilioAuthToken\",
 \"DefaultFromPhoneNumber\": \"4254492999\"}}]"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
connectors/55d0052b-0d7d-44d6-b56f-21867215e118/versions/33f709a0-c825-49cb-9eea-
dc8964fbd635",
    "CreationTimestamp": "2019-06-24T20:46:30.134Z",
    "Id": "55d0052b-0d7d-44d6-b56f-21867215e118",
    "Version": "33f709a0-c825-49cb-9eea-dc8964fbd635"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateConnectorDefinitionVersion。

create-connector-definition

以下程式碼範例顯示如何使用 create-connector-definition。

AWS CLI

建立連接器定義

下列create-connector-definition範例會建立連接器定義和初始連接器定義版本。初始版本包含一個連接器。版本中的所有連接器都會為其參數定義值。

```
aws greengrass create-connector-definition \
```

```
    --name MySNSConnector \
    --initial-version "{\"Connectors\": [{\"Id\":\"MySNSConnector\",\"ConnectorArn
\":\"arn:aws:greengrass:us-west-2::/connectors/SNS/versions/1\",\"Parameters\":
 {\"DefaultSNSArn\":\"arn:aws:sns:us-west-2:123456789012:GGConnectorTopic\"}}]}"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
    "CreationTimestamp": "2019-06-19T19:30:01.300Z",
    "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
    "LastUpdatedTimestamp": "2019-06-19T19:30:01.300Z",
    "LatestVersion": "63c57963-c7c2-4a26-a7e2-7bf478ea2623",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/versions/63c57963-
c7c2-4a26-a7e2-7bf478ea2623",
    "Name": "MySNSConnector"
}
```

如需詳細資訊，請參閱《AWS IoT Greengrass 開發人員指南》中的 Greengrass 連接器 (CLI) 入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateConnectorDefinition。

create-core-definition-version

以下程式碼範例顯示如何使用 create-core-definition-version。

AWS CLI

建立核心定義版本

下列create-core-definition-version範例會建立核心定義版本，並將其與指定的核心定義建立關聯。版本只能包含一個核心。您必須先建立和佈建對應的 AWS IoT 物件，才能建立核心。此程序包含下列iot命令，其會傳回create-core-definition-version命令CertificateArn所需的 ThingArn和 。

建立對應至核心裝置的 AWS IoT 物件：

```
aws iot create-thing \
    --thing-name "MyCoreDevice"
```

輸出：

```
{
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice",
    "thingName": "MyCoreDevice",
    "thingId": "cb419a19-9099-4515-9cec-e9b0e760608a"
}
```

為物件建立公有和私有金鑰和核心裝置憑證。此範例使用 create-keys-and-certificate命令，並需要目前目錄的寫入許可。或者，您可以使用 create-certificate-from-csr命令。

```
aws iot create-keys-and-certificate \
    --set-as-active \
    --certificate-pem-outfile "myCore.cert.pem" \
    --public-key-outfile "myCore.public.key" \
    --private-key-outfile "myCore.private.key"
```

輸出：

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz",
    "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCAkGgAwIBATgIUCgq6EGqou6zFqWgIZRndgQEFW+gwDQYJKoZIhvc...KdGewQS\n-----END
 CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqKpRgnn6yq26U3y...wIDAQAB\n-----END
 PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjbMXDcNOK4rMCxDR...fvY4+te\n-----END
 RSA PRIVATE KEY-----\n"
    },
    "certificateId":
 "123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
}
```

建立允許 iot和 greengrass動作的 AWS IoT 政策。為了簡化，以下政策允許對所有資源執行動作，但您的政策應該更嚴格。

```
aws iot create-policy \
```

```
    --policy-name "Core_Devices" \
    --policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}"
```

輸出：

```
{
    "policyName": "Core_Devices",
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/Core_Devices",
    "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}",
    "policyVersionId": "1"
}
```

將政策連接至憑證：

```
aws iot attach-policy \
    --policy-name "Core_Devices" \
    --target "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

此命令不會產生輸出。

將物件連接到憑證：

```
aws iot attach-thing-principal \
    --thing-name "MyCoreDevice" \
    --principal "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

此命令不會產生輸出。

建立核心定義版本：

```
aws greengrass create-core-definition-version \
    --core-definition-id "582efe12-b05a-409e-9a24-a2ba1bcc4a12" \
    --cores "[{\"Id\":\"MyCoreDevice\",\"ThingArn\":\"arn:aws:iot:us-
west-2:123456789012:thing/MyCoreDevice\",\"CertificateArn\":\"arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz
\",\"SyncShadow\":true}]"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
cores/582efe12-b05a-409e-9a24-a2ba1bcc4a12/versions/3fdc1190-2ce5-44de-b98b-
eec8f9571014",
    "Version": "3fdc1190-2ce5-44de-b98b-eec8f9571014",
    "CreationTimestamp": "2019-09-18T00:15:09.838Z",
    "Id": "582efe12-b05a-409e-9a24-a2ba1bcc4a12"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的設定 AWS IoT Greengrass 核心。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCoreDefinitionVersion。

create-core-definition

以下程式碼範例顯示如何使用 create-core-definition。

AWS CLI

範例 1：建立空的核心定義

下列create-core-definition範例會建立空白 （ 無初始版本 ） Greengrass 核心定義。在核心可供使用之前，您必須使用 create-core-definition-version命令來提供核心的其他參數。

```
aws greengrass create-core-definition \
    --name cliGroup_Core
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/
b5c08008-54cb-44bd-9eec-c121b04283b5",
    "CreationTimestamp": "2019-06-25T18:23:22.106Z",
```

```
    "Id": "b5c08008-54cb-44bd-9eec-c121b04283b5",
    "LastUpdatedTimestamp": "2019-06-25T18:23:22.106Z",
    "Name": "cliGroup_Core"
}
```

範例 2：使用初始版本建立核心定義

下列create-core-definition範例會建立包含初始核心定義版本的核心定義。版本只能包含一個核心。您必須先建立和佈建對應的 AWS IoT 物件，才能建立核心。此程序包含下列iot命令，其會傳回create-core-definition命令CertificateArn所需的 ThingArn和 。

建立對應至核心裝置的 AWS IoT 物件：

```
aws iot create-thing \
    --thing-name "MyCoreDevice"
```

輸出：

```
{
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice",
    "thingName": "MyCoreDevice",
    "thingId": "cb419a19-9099-4515-9cec-e9b0e760608a"
}
```

為物件建立公有和私有金鑰和核心裝置憑證。此範例使用 create-keys-and-certificate命令，並需要目前目錄的寫入許可。或者，您可以使用 create-certificate-from-csr命令。

```
aws iot create-keys-and-certificate \
    --set-as-active \
    --certificate-pem-outfile "myCore.cert.pem" \
    --public-key-outfile "myCore.public.key" \
    --private-key-outfile "myCore.private.key"
```

輸出：

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz",
    "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCAkGgAwIBATgIUCgq6EGqou6zFqWgIZRndgQEFW+gwDQYJKoZIhvc...KdGewQS\n-----END
 CERTIFICATE-----\n",
```

```
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqKpRgnn6yq26U3y...wIDAQAB\n-----END
 PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjbMXDcNOK4rMCxDR...fvY4+te\n-----END
 RSA PRIVATE KEY-----\n"
    },
    "certificateId":
 "123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
}
```

建立允許 iot和 greengrass動作的 AWS IoT 政策。為了簡化，以下政策允許對所有資源執行動作，但您的政策應該更嚴格。

```
aws iot create-policy \
    --policy-name "Core_Devices" \
    --policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}"
```

輸出：

```
{
    "policyName": "Core_Devices",
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/Core_Devices",
    "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}",
    "policyVersionId": "1"
}
```

將政策連接至憑證：

```
aws iot attach-policy \
    --policy-name "Core_Devices" \
```

```
    --target "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

此命令不會產生輸出。

將物件連接到憑證：

```
aws iot attach-thing-principal \
    --thing-name "MyCoreDevice" \
    --principal "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

此命令不會產生輸出。

建立核心定義：

```
aws greengrass create-core-definition \
    --name "MyCores" \
    --initial-version "{\"Cores\":[{\"Id\":\"MyCoreDevice\",\"ThingArn\":
\"arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice\",\"CertificateArn\":
\"arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz
\",\"SyncShadow\":true}]}"
```

輸出：

```
{
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/582efe12-b05a-409e-9a24-a2ba1bcc4a12/versions/
cc87b5b3-8f4b-465d-944c-1d6de5dbfcdb",
    "Name": "MyCores",
    "LastUpdatedTimestamp": "2019-09-18T00:11:06.197Z",
    "LatestVersion": "cc87b5b3-8f4b-465d-944c-1d6de5dbfcdb",
    "CreationTimestamp": "2019-09-18T00:11:06.197Z",
    "Id": "582efe12-b05a-409e-9a24-a2ba1bcc4a12",
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
cores/582efe12-b05a-409e-9a24-a2ba1bcc4a12"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的設定 AWS IoT Greengrass 核心。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCoreDefinition。

create-deployment

以下程式碼範例顯示如何使用 create-deployment。

AWS CLI

為 Greengrass 群組的版本建立部署

下列create-deployment範例會部署 Greengrass 群組的指定版本。

```
aws greengrass create-deployment \
    --deployment-type NewDeployment \
    --group-id "ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca" \
    --group-version-id "dc40c1e9-e8c8-4d28-a84d-a9cad5f599c9"
```

輸出：

```
{
    "DeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca/deployments/bfceb608-4e97-45bc-
af5c-460144270308",
    "DeploymentId": "bfceb608-4e97-45bc-af5c-460144270308"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的連接器 (CLI) 入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDeployment。

create-device-definition-version

以下程式碼範例顯示如何使用 create-device-definition-version。

AWS CLI

建立裝置定義版本

下列create-device-definition-version範例會建立裝置定義版本，並將其與指定的裝置定義建立關聯。版本定義兩個裝置。您必須先建立和佈建對應的 AWS IoT 物件，才能建立 Greengrass 裝置。此程序包含下列您必須執行的iot命令，以取得 Greengrass 命令的必要資訊：

建立對應至裝置的 AWS IoT 物件：

```
aws iot create-thing \
```

```
    --thing-name "InteriorTherm"
```

輸出：

```
{
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/InteriorTherm",
    "thingName": "InteriorTherm",
    "thingId": "01d4763c-78a6-46c6-92be-7add080394bf"
}
```

建立公有和私有金鑰以及物件的裝置憑證。此範例使用 create-keys-and-certificate命令，並需要目前目錄的寫入許可。或者，您可以使用 create-certificate-from-csr命令：

```
aws iot create-keys-and-certificate \
    --set-as-active \
    --certificate-pem-outfile "myDevice.cert.pem" \
    --public-key-outfile "myDevice.public.key" \
    --private-key-outfile "myDevice.private.key"
```

輸出：

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92",
    "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCAkGgAwIBATgIUCgq6EGqou6zFqWgIZRndgQEFW+gwDQYJKoZIhvc...KdGewQS\n-----END
 CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqKpRgnn6yq26U3y...wIDAQAB\n-----END
 PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjbMXDcNOK4rMCxDR...fvY4+te\n-----END
 RSA PRIVATE KEY-----\n"
    },
    "certificateId":
 "66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
}
```

建立允許 iot和 greengrass動作的 AWS IoT 政策。為了簡化，下列政策允許對所有資源執行動作，但您的政策可以更嚴格：

```
aws iot create-policy \
    --policy-name "GG_Devices" \
    --policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}"
```

輸出：

```
{
    "policyName": "GG_Devices",
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/GG_Devices",
    "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}",
    "policyVersionId": "1"
}
```

將政策連接至憑證：

```
aws iot attach-policy \
    --policy-name "GG_Devices" \
    --target "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
```

將物件連接至憑證

```
aws iot attach-thing-principal \
    --thing-name "InteriorTherm" \
    --principal "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
```

如上所示建立和設定 IoT 物件之後，請使用下列範例中前兩個命令中的 `CertificateArn` `ThingArn`和 。

```
aws greengrass create-device-definition-version \
```

```
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \
    --devices "[{\"Id\":\"InteriorTherm\",\"ThingArn\":\"arn:aws:iot:us-
west-2:123456789012:thing/InteriorTherm\",\"CertificateArn\":\"arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92\",
\"SyncShadow\":true},{\"Id\":\"ExteriorTherm\",\"ThingArn\":\"arn:aws:iot:us-
west-2:123456789012:thing/ExteriorTherm\",\"CertificateArn\":\"arn:aws:iot:us-
west-2:123456789012:cert/6c52ce1b47bde88a637e9ccdd45fe4e4c2c0a75a6866f8f63d980ee22fa51e02\",
\"SyncShadow\":true}]"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/83c13984-6fed-447e-84d5-5b8aa45d5f71",
    "Version": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
    "CreationTimestamp": "2019-09-11T00:15:09.838Z",
    "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDeviceDefinitionVersion。

create-device-definition

以下程式碼範例顯示如何使用 create-device-definition。

AWS CLI

建立裝置定義

下列create-device-definition範例會建立包含初始裝置定義版本的裝置定義。初始版本定義兩個裝置。您必須先建立和佈建對應的 AWS IoT 物件，才能建立 Greengrass 裝置。此程序包含下列您必須執行的iot命令，以取得 Greengrass 命令的必要資訊：

建立對應至裝置的 AWS IoT 物件：

```
aws iot create-thing \
    --thing-name "InteriorTherm"
```

輸出：

```
{
```

```
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/InteriorTherm",
    "thingName": "InteriorTherm",
    "thingId": "01d4763c-78a6-46c6-92be-7add080394bf"
}
```

建立公有和私有金鑰以及物件的裝置憑證。此範例使用 create-keys-and-certificate命令，並需要目前目錄的寫入許可。或者，您可以使用 create-certificate-from-csr命令：

```
aws iot create-keys-and-certificate \
    --set-as-active \
    --certificate-pem-outfile "myDevice.cert.pem" \
    --public-key-outfile "myDevice.public.key" \
    --private-key-outfile "myDevice.private.key"
```

輸出：

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92",
    "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCAkGgAwIBATgIUCgq6EGqou6zFqWgIZRndgQEFW+gwDQYJKoZIhvc...KdGewQS\n-----END
 CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqKpRgnn6yq26U3y...wIDAQAB\n-----END
 PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjbMXDcNOK4rMCxDR...fvY4+te\n-----END
 RSA PRIVATE KEY-----\n"
    },
    "certificateId":
 "66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
}
```

建立允許 iot和 greengrass動作的 AWS IoT 政策。為了簡化，下列政策允許對所有資源執行動作，但您的政策可以更嚴格：

```
aws iot create-policy \
    --policy-name "GG_Devices" \
    --policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
```

```
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}"
```

輸出：

```
{
    "policyName": "GG_Devices",
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/GG_Devices",
    "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}",
    "policyVersionId": "1"
}
```

將政策連接至憑證：

```
aws iot attach-policy \
    --policy-name "GG_Devices" \
    --target "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
```

將物件連接至憑證

```
aws iot attach-thing-principal \
    --thing-name "InteriorTherm" \
    --principal "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
```

如上所示建立和設定 IoT 物件之後，請使用下列範例中前兩個命令中的 `CertificateArn` `ThingArn`和 。

```
aws greengrass create-device-definition \
    --name "Sensors" \
    --initial-version "{\"Devices\":[{\"Id\":\"InteriorTherm
\",\"ThingArn\":\"arn:aws:iot:us-west-2:123456789012:thing/
InteriorTherm\",\"CertificateArn\":\"arn:aws:iot:us-
```

```
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92\",
\"SyncShadow\":true},{\"Id\":\"ExteriorTherm\",\"ThingArn\":\"arn:aws:iot:us-
west-2:123456789012:thing/ExteriorTherm\",\"CertificateArn\":\"arn:aws:iot:us-
west-2:123456789012:cert/6c52ce1b47bde88a637e9ccdd45fe4e4c2c0a75a6866f8f63d980ee22fa51e02\",
\"SyncShadow\":true}]}"
```

輸出：

```
{
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/3b5cc510-58c1-44b5-9d98-4ad858ffa795",
    "Name": "Sensors",
    "LastUpdatedTimestamp": "2019-09-11T00:11:06.197Z",
    "LatestVersion": "3b5cc510-58c1-44b5-9d98-4ad858ffa795",
    "CreationTimestamp": "2019-09-11T00:11:06.197Z",
    "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDeviceDefinition。

create-function-definition-version

以下程式碼範例顯示如何使用 create-function-definition-version。

AWS CLI

建立函數定義的版本

下列create-function-definition-version範例會建立新的指定函數定義版本。此版本指定 ID 為 的單一函數Hello-World-function，允許存取檔案系統，並指定記憶體大小上限和逾時期間。

```
aws greengrass create-function-definition-version \
    --cli-input-json "{\"FunctionDefinitionId\": \"e626e8c9-3b8f-4bf3-9cdc-
d26ecdeb9fa3\",\"Functions\": [{\"Id\": \"Hello-World-function\", \"FunctionArn\":
 \""arn:aws:lambda:us-
west-2:123456789012:function:Greengrass_HelloWorld_Counter:gghw-alias"\",
\"FunctionConfiguration\": {\"Environment\": {\"AccessSysfs\": true},\"Executable\":
```

```
 \"greengrassHelloWorldCounter.function_handler\",\"MemorySize\": 16000,\"Pinned\":
 false,\"Timeout\": 25}}]}"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/e626e8c9-3b8f-4bf3-9cdc-d26ecdeb9fa3/
versions/74abd1cc-637e-4abe-8684-9a67890f4043",
    "CreationTimestamp": "2019-06-25T22:03:43.376Z",
    "Id": "e626e8c9-3b8f-4bf3-9cdc-d26ecdeb9fa3",
    "Version": "74abd1cc-637e-4abe-8684-9a67890f4043"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFunctionDefinitionVersion。

create-function-definition

以下程式碼範例顯示如何使用 create-function-definition。

AWS CLI

建立 Lambda 函數定義

下列create-function-definition範例會透過提供 Lambda 函數清單 （在此情況下，僅列出一個名為 的函數TempMonitorFunction) 及其組態，來建立 Lambda 函數定義和初始版本。在建立函數定義之前，您需要 Lambda 函數 ARN。若要建立函數及其別名，請使用 Lambda 的create-function和 publish-version命令。Lambda 的create-function命令需要執行角色的 ARN，即使 AWS IoT Greengrass 不會使用該角色，因為許可是在 Greengrass 群組角色中指定。您可以使用 IAM create-role命令來建立空角色，以取得可與 Lambda 的 搭配使用的 ARN，create-function或者您可以使用現有的執行角色。

```
aws greengrass create-function-definition \
    --name MyGreengrassFunctions \
    --initial-version "{\"Functions\": [{\"Id\": \"TempMonitorFunction\",
 \"FunctionArn\": \"arn:aws:lambda:us-
west-2:123456789012:function:TempMonitor:GG_TempMonitor\", \"FunctionConfiguration
\": {\"Executable\": \"temp_monitor.function_handler\", \"MemorySize\": 16000,
\"Timeout\": 5}}]}"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
functions/3b0d0080-87e7-48c6-b182-503ec743a08b",
    "CreationTimestamp": "2019-06-19T22:24:44.585Z",
    "Id": "3b0d0080-87e7-48c6-b182-503ec743a08b",
    "LastUpdatedTimestamp": "2019-06-19T22:24:44.585Z",
    "LatestVersion": "67f918b9-efb4-40b0-b87c-de8c9faf085b",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/3b0d0080-87e7-48c6-b182-503ec743a08b/versions/67f918b9-
efb4-40b0-b87c-de8c9faf085b",
    "Name": "MyGreengrassFunctions"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的如何使用 AWS 命令列界面設定本機資源存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFunctionDefinition。

create-group-certificate-authority

以下程式碼範例顯示如何使用 create-group-certificate-authority。

AWS CLI

為群組建立憑證授權機構 (CA)

下列create-group-certificate-authority範例會為指定的群組建立或輪換 CA。

```
aws greengrass create-group-certificate-authority \
    --group-id "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1"
```

輸出：

```
{
    "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/8eaadd72-ce4b-4f15-892a-0cc4f3a343f1/certificateauthorities/
d31630d674c4437f6c5dbc0dca56312a902171ce2d086c38e509c8EXAMPLEcc5"
}
```

如需詳細資訊，請參閱AWS IoT Greengrass 開發人員指南AWS 》中的 IoT Greengrass 安全性。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateGroupCertificateAuthority。

create-group-version

以下程式碼範例顯示如何使用 create-group-version。

AWS CLI

建立 Greengrass 群組的版本

下列create-group-version範例會建立群組版本，並將其與指定的群組建立關聯。版本參考的核心、資源、連接器、函數和訂閱版本，其中包含要包含在此群組版本中的實體。您必須先建立這些實體，才能建立群組版本。

若要使用初始版本建立資源定義，請使用 create-resource-definition command.To 建立具有初始版本的連接器定義、使用 create-connector-definition command.To 建立具有初始版本的函數定義、使用 create-function-definition command.To 建立具有初始版本的訂閱定義、使用 create-subscription-definition command.To 擷取最新核心定義版本的 ARN、使用 get-group-version命令並指定最新群組版本的 ID。

```
aws greengrass create-group-version \
    --group-id "ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca" \
    --core-definition-version-arn "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/6a630442-8708-4838-ad36-eb98849d975e/
versions/6c87151b-1fb4-4cb2-8b31-6ee715d8f8ba" \
    --resource-definition-version-arn "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38/versions/
a5f94d0b-f6bc-40f4-bb78-7a1c5fe13ba1" \
    --connector-definition-version-arn "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/connectors/55d0052b-0d7d-44d6-b56f-21867215e118/
versions/78a3331b-895d-489b-8823-17b4f9f418a0" \
    --function-definition-version-arn "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/3b0d0080-87e7-48c6-b182-503ec743a08b/
versions/67f918b9-efb4-40b0-b87c-de8c9faf085b" \
    --subscription-definition-version-arn "arn:aws:greengrass:us-
west-2:123456789012:/greengrass/definition/subscriptions/9d611d57-5d5d-44bd-a3b4-
feccbdd69112/versions/aa645c47-ac90-420d-9091-8c7ffa4f103f"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/
ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca/versions/e10b0459-4345-4a09-88a4-1af1f5d34638",
    "CreationTimestamp": "2019-06-20T18:42:47.020Z",
```

```
    "Id": "ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca",
    "Version": "e10b0459-4345-4a09-88a4-1af1f5d34638"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的 IoT Greengrass 群組物件模型概觀。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateGroupVersion。

create-group

以下程式碼範例顯示如何使用 create-group。

AWS CLI

建立 Greeengrass 群組

下列create-group範例會建立名為 的群組cli-created-group。

```
aws greengrass create-group \
    --name cli-created-group
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/4e22bd92-898c-436b-ade5-434d883ff749",
    "CreationTimestamp": "2019-06-25T18:07:17.688Z",
    "Id": "4e22bd92-898c-436b-ade5-434d883ff749",
    "LastUpdatedTimestamp": "2019-06-25T18:07:17.688Z",
    "Name": "cli-created-group"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的 IoT Greengrass 群組物件模型概觀。 AWS IoT

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateGroup。

create-logger-definition-version

以下程式碼範例顯示如何使用 create-logger-definition-version。

AWS CLI

建立記錄器定義版本

下列create-logger-definition-version範例會建立記錄器定義版本，並將其與記錄器定義建立關聯。此版本定義四種記錄組態：1) 核心裝置檔案系統上的系統元件日誌、2) 核心裝置檔案系統上的使用者定義 Lambda 函數日誌、3) Amazon CloudWatch Logs 中的系統元件日誌，以及 4) Amazon CloudWatch Logs 中的使用者定義 Lambda 函數日誌。注意：對於 CloudWatch Logs 整合，您的群組角色必須授予適當的許可。

```
aws greengrass create-logger-definition-version \
    --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0" \
    --loggers "[{\"Id\":\"1\",\"Component\":\"GreengrassSystem\",\"Level\":\"ERROR
\",\"Space\":10240,\"Type\":\"FileSystem\"},{\"Id\":\"2\",\"Component\":\"Lambda
\",\"Level\":\"INFO\",\"Space\":10240,\"Type\":\"FileSystem\"},{\"Id\":\"3\",
\"Component\":\"GreengrassSystem\",\"Level\":\"WARN\",\"Type\":\"AWSCloudWatch\"},
{\"Id\":\"4\",\"Component\":\"Lambda\",\"Level\":\"INFO\",\"Type\":\"AWSCloudWatch
\"}]"
```

輸出：

```
{
 "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/loggers/
a454b62a-5d56-4ca9-bdc4-8254e1662cb0/versions/49aedb1e-01a3-4d39-9871-3a052573f1ea",
 "Version": "49aedb1e-01a3-4d39-9871-3a052573f1ea",
 "CreationTimestamp": "2019-07-24T00:04:48.523Z",
 "Id": "a454b62a-5d56-4ca9-bdc4-8254e1662cb0"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的使用 IoT Greengrass 日誌進行監控。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLoggerDefinitionVersion。

create-logger-definition

以下程式碼範例顯示如何使用 create-logger-definition。

AWS CLI

建立記錄器定義

下列create-logger-definition範例會建立記錄器定義，其中包含初始記錄器定義版本。初始版本定義三種記錄組態：1) 核心裝置檔案系統上的系統元件日誌、2) 核心裝置檔案系統上的使用者定義 Lambda 函數日誌，以及 3) Amazon CloudWatch Logs 中的使用者定義 Lambda 函數日誌。注意：對於 CloudWatch Logs 整合，您的群組角色必須授予適當的許可。

```
aws greengrass create-logger-definition \
    --name "LoggingConfigs" \
    --initial-version "{\"Loggers\":[{\"Id\":\"1\",\"Component\":\"GreengrassSystem
\",\"Level\":\"ERROR\",\"Space\":10240,\"Type\":\"FileSystem\"},{\"Id\":
\"2\",\"Component\":\"Lambda\",\"Level\":\"INFO\",\"Space\":10240,\"Type\":
\"FileSystem\"},{\"Id\":\"3\",\"Component\":\"Lambda\",\"Level\":\"INFO\",\"Type\":
\"AWSCloudWatch\"}]}"
```

輸出：

```
{
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/a454b62a-5d56-4ca9-bdc4-8254e1662cb0/versions/de1d9854-1588-4525-
b25e-b378f60f2322",
    "Name": "LoggingConfigs",
    "LastUpdatedTimestamp": "2019-07-23T23:52:17.165Z",
    "LatestVersion": "de1d9854-1588-4525-b25e-b378f60f2322",
    "CreationTimestamp": "2019-07-23T23:52:17.165Z",
    "Id": "a454b62a-5d56-4ca9-bdc4-8254e1662cb0",
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
loggers/a454b62a-5d56-4ca9-bdc4-8254e1662cb0"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的使用 IoT Greengrass 日誌進行監控。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLoggerDefinition。

create-resource-definition-version

以下程式碼範例顯示如何使用 create-resource-definition-version。

AWS CLI

建立資源定義的版本

下列create-resource-definition-version範例會建立新的 TwilioAuthToken 版本。

```
aws greengrass create-resource-definition-version \
    --resource-definition-id "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38" \
    --resources "[{\"Id\": \"TwilioAuthToken\",\"Name\": \"MyTwilioAuthToken
\",\"ResourceDataContainer\": {\"SecretsManagerSecretResourceData\": {\"ARN\":
 \"arn:aws:secretsmanager:us-west-2:123456789012:secret:greengrass-TwilioAuthToken-
ntSlp6\"}}}]"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38/versions/b3bcada0-5fb6-42df-
bf0b-1ee4f15e769e",
    "CreationTimestamp": "2019-06-24T21:17:25.623Z",
    "Id": "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",
    "Version": "b3bcada0-5fb6-42df-bf0b-1ee4f15e769e"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateResourceDefinitionVersion。

create-resource-definition

以下程式碼範例顯示如何使用 create-resource-definition。

AWS CLI

建立資源定義

下列create-resource-definition範例會建立資源定義，其中包含要在 Greengrass 群組中使用的資源清單。在此範例中，透過提供資源清單來包含資源定義的初始版本。此清單包含一個 Twilio 授權字符的資源，以及存放在 AWS Secrets Manager 中秘密的 ARN。您必須先建立秘密，才能建立資源定義。

```
aws greengrass create-resource-definition \
    --name MyGreengrassResources \
    --initial-version "{\"Resources\": [{\"Id\": \"TwilioAuthToken
\",\"Name\": \"MyTwilioAuthToken\",\"ResourceDataContainer\":
 {\"SecretsManagerSecretResourceData\": {\"ARN\": \"arn:aws:secretsmanager:us-
west-2:123456789012:secret:greengrass-TwilioAuthToken-ntSlp6\"}}}]}"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",
    "CreationTimestamp": "2019-06-19T21:51:28.212Z",
    "Id": "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",
    "LastUpdatedTimestamp": "2019-06-19T21:51:28.212Z",
    "LatestVersion": "a5f94d0b-f6bc-40f4-bb78-7a1c5fe13ba1",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38/versions/a5f94d0b-
f6bc-40f4-bb78-7a1c5fe13ba1",
    "Name": "MyGreengrassResources"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的如何使用 AWS 命令列界面設定本機資源存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateResourceDefinition。

create-software-update-job

以下程式碼範例顯示如何使用 create-software-update-job。

AWS CLI

建立核心的軟體更新任務

下列create-software-update-job範例會建立over-the-air (OTA) 更新任務，以更新名稱為 之核心上的 AWS IoT Greengrass Core 軟體MyFirstGroup_Core。此命令需要 IAM 角色，允許存取 Amazon S3 中的軟體更新套件，並包含 iot.amazonaws.com 作為信任的實體。

```
aws greengrass create-software-update-job \
    --update-targets-architecture armv7l \
    --update-targets [\"arn:aws:iot:us-west-2:123456789012:thing/MyFirstGroup_Core
\"] \
    --update-targets-operating-system raspbian \
    --software-to-update core \
    --s3-url-signer-role arn:aws:iam::123456789012:role/OTA_signer_role \
    --update-agent-log-level WARN
```

輸出：

```
{
```

```
    "IotJobId": "GreengrassUpdateJob_30b353e3-3af7-4786-be25-4c446663c09e",
    "IotJobArn": "arn:aws:iot:us-west-2:123456789012:job/
GreengrassUpdateJob_30b353e3-3af7-4786-be25-4c446663c09e",
    "PlatformSoftwareVersion": "1.9.3"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的 IoT Greengrass 核心軟體的 OTA 更新。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSoftwareUpdateJob。

create-subscription-definition-version

以下程式碼範例顯示如何使用 create-subscription-definition-version。

AWS CLI

建立新的訂閱定義版本

下列create-subscription-definition-version範例會建立新的訂閱定義版本，其中包含三個訂閱：觸發通知、溫度輸入和輸出狀態。

```
aws greengrass create-subscription-definition-version \
    --subscription-definition-id "9d611d57-5d5d-44bd-a3b4-feccbdd69112" \
    --subscriptions "[{\"Id\": \"TriggerNotification\", \"Source\":
 \"arn:aws:lambda:us-west-2:123456789012:function:TempMonitor:GG_TempMonitor
\", \"Subject\": \"twilio/txt\", \"Target\": \"arn:aws:greengrass:us-west-2::/
connectors/TwilioNotifications/versions/1\"},{\"Id\": \"TemperatureInput\", \"Source
\": \"cloud\", \"Subject\": \"temperature/input\", \"Target\": \"arn:aws:lambda:us-
west-2:123456789012:function:TempMonitor:GG_TempMonitor\"},{\"Id\": \"OutputStatus
\", \"Source\": \"arn:aws:greengrass:us-west-2::/connectors/TwilioNotifications/
versions/1\", \"Subject\": \"twilio/message/status\", \"Target\": \"cloud\"}]"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
subscriptions/9d611d57-5d5d-44bd-a3b4-feccbdd69112/versions/7b65dfae-50b6-4d0f-
b3e0-27728bfb0620",
    "CreationTimestamp": "2019-06-24T21:21:33.837Z",
    "Id": "9d611d57-5d5d-44bd-a3b4-feccbdd69112",
    "Version": "7b65dfae-50b6-4d0f-b3e0-27728bfb0620"
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSubscriptionDefinitionVersion。

create-subscription-definition

以下程式碼範例顯示如何使用 create-subscription-definition。

AWS CLI

建立訂閱定義

下列create-subscription-definition範例會建立訂閱定義，並指定其初始版本。初始版本包含三個訂閱：一個用於連接器訂閱的 MQTT 主題，一個用於允許函數接收來自 AWS IoT 的溫度讀數，另一個用於允許 AWS IoT 從連接器接收狀態資訊。此範例提供先前使用 Lambda create-alias命令建立之 Lambda 函數別名的 ARN。

```
aws greengrass create-subscription-definition \
    --initial-version "{\"Subscriptions\": [{\"Id\":
 \"TriggerNotification\", \"Source\": \"arn:aws:lambda:us-
west-2:123456789012:function:TempMonitor:GG_TempMonitor\", \"Subject\":
 \"twilio/txt\", \"Target\": \"arn:aws:greengrass:us-west-2::/connectors/
TwilioNotifications/versions/1\"},{\"Id\": \"TemperatureInput\", \"Source\":
 \"cloud\", \"Subject\": \"temperature/input\", \"Target\": \"arn:aws:lambda:us-
west-2:123456789012:function:TempMonitor:GG_TempMonitor\"},{\"Id\": \"OutputStatus
\", \"Source\": \"arn:aws:greengrass:us-west-2::/connectors/TwilioNotifications/
versions/1\", \"Subject\": \"twilio/message/status\", \"Target\": \"cloud\"}]}"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
subscriptions/9d611d57-5d5d-44bd-a3b4-feccbdd69112",
    "CreationTimestamp": "2019-06-19T22:34:26.677Z",
    "Id": "9d611d57-5d5d-44bd-a3b4-feccbdd69112",
    "LastUpdatedTimestamp": "2019-06-19T22:34:26.677Z",
    "LatestVersion": "aa645c47-ac90-420d-9091-8c7ffa4f103f",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/subscriptions/9d611d57-5d5d-44bd-a3b4-feccbdd69112/versions/aa645c47-
ac90-420d-9091-8c7ffa4f103f"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的連接器 (CLI) 入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSubscriptionDefinition。

delete-connector-definition

以下程式碼範例顯示如何使用 delete-connector-definition。

AWS CLI

刪除連接器定義

下列delete-connector-definition範例會刪除指定的 Greengrass 連接器定義。如果您刪除群組使用的連接器定義，則無法成功部署該群組。

```
aws greengrass delete-connector-definition \
    --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteConnectorDefinition。

delete-core-definition

以下程式碼範例顯示如何使用 delete-core-definition。

AWS CLI

刪除核心定義

下列delete-core-definition範例會刪除指定的 Greengrass 核心定義，包括所有版本。如果您刪除與 Greengrass 群組相關聯的核心，則無法成功部署該群組。

```
aws greengrass delete-core-definition \
    --core-definition-id "ff36cc5f-9f98-4994-b468-9d9b6dc52abd"
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCoreDefinition。

delete-device-definition

以下程式碼範例顯示如何使用 delete-device-definition。

AWS CLI

刪除裝置定義

下列delete-device-definition範例會刪除指定的裝置定義，包括其所有版本。如果您刪除群組版本所使用的裝置定義版本，則無法成功部署群組版本。

```
aws greengrass delete-device-definition \
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDeviceDefinition。

delete-function-definition

以下程式碼範例顯示如何使用 delete-function-definition。

AWS CLI

刪除函數定義

下列delete-function-definition範例會刪除指定的 Greengrass 函數定義。如果您刪除群組使用的函數定義，則無法成功部署該群組。

```
aws greengrass delete-function-definition \
    --function-definition-id "fd4b906a-dff3-4c1b-96eb-52ebfcfac06a"
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFunctionDefinition。

delete-group

以下程式碼範例顯示如何使用 delete-group。

AWS CLI

刪除群組

下列delete-group範例會刪除指定的 Greengrass 群組。

```
aws greengrass delete-group \
    --group-id "4e22bd92-898c-436b-ade5-434d883ff749"
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteGroup。

delete-logger-definition

以下程式碼範例顯示如何使用 delete-logger-definition。

AWS CLI

刪除記錄器定義

下列delete-logger-definition範例會刪除指定的記錄器定義，包括所有記錄器定義版本。如果您刪除群組版本所使用的記錄器定義版本，則無法成功部署群組版本。

```
aws greengrass delete-logger-definition \
    --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS IoT Greengrass 開發人員指南》中的使用 IoT Greengrass 日誌進行監控。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLoggerDefinition。

delete-resource-definition

以下程式碼範例顯示如何使用 delete-resource-definition。

AWS CLI

刪除資源定義

下列delete-resource-definition範例會刪除指定的資源定義，包括所有資源版本。如果您刪除群組使用的資源定義，則無法成功部署該群組。

```
aws greengrass delete-resource-definition \
    --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteResourceDefinition。

delete-subscription-definition

以下程式碼範例顯示如何使用 delete-subscription-definition。

AWS CLI

刪除訂閱定義

下列delete-subscription-definition範例會刪除指定的 Greengrass 訂閱定義。如果您刪除群組正在使用的訂閱，則無法成功部署該群組。

```
aws greengrass delete-subscription-definition \
    --subscription-definition-id "cd6f1c37-d9a4-4e90-be94-01a7404f5967"
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSubscriptionDefinition。

disassociate-role-from-group

以下程式碼範例顯示如何使用 disassociate-role-from-group。

AWS CLI

取消角色與 Greengrass 群組的關聯

下列disassociate-role-from-group範例會取消 IAM 角色與指定 Greengrass 群組的關聯。

```
aws greengrass disassociate-role-from-group \
    --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b
```

輸出：

```
{
    "DisassociatedAt": "2019-09-10T20:05:49Z"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的設定群組角色。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateRoleFromGroup。

disassociate-service-role-from-account

以下程式碼範例顯示如何使用 disassociate-service-role-from-account。

AWS CLI

取消服務角色與 AWS 帳戶的關聯

下列disassociate-service-role-from-account範例會移除與您 AWS 帳戶相關聯的服務角色。如果您不是在任何區域中使用 服務角色 AWS ，請使用 delete-role-policy命令從角色分離AWSGreengrassResourceAccessRolePolicy受管政策，然後使用 delete-role命令刪除角色。

```
aws greengrass disassociate-service-role-from-account
```

輸出：

```
{
    "DisassociatedAt": "2019-06-25T22:12:55Z"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的 Greengrass 服務角色。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateServiceRoleFromAccount。

get-associated-role

以下程式碼範例顯示如何使用 get-associated-role。

AWS CLI

取得與 Greengrass 群組相關聯的角色

下列get-associated-role範例會取得與指定 Greengrass 群組相關聯的 IAM 角色。本機 Lambda 函數和連接器會使用群組角色來存取 AWS 服務。

```
aws greengrass get-associated-role \
    --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b
```

輸出：

```
{
    "RoleArn": "arn:aws:iam::123456789012:role/GG-Group-Role",
    "AssociatedAt": "2019-09-10T20:03:30Z"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的設定群組角色。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAssociatedRole。

get-bulk-deployment-status

以下程式碼範例顯示如何使用 get-bulk-deployment-status。

AWS CLI

檢查大量部署的狀態

下列get-bulk-deployment-status範例會擷取指定大量部署操作的狀態資訊。在此範例中，指定要部署之群組的檔案具有無效的輸入記錄。

```
aws greengrass get-bulk-deployment-status \
    --bulk-deployment-id "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
```

輸出：

```
{
    "BulkDeploymentMetrics": {
        "InvalidInputRecords": 1,
        "RecordsProcessed": 1,
        "RetryAttempts": 0
    },
    "BulkDeploymentStatus": "Completed",
    "CreatedAt": "2019-06-25T16:11:33.265Z",
    "tags": {}
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的為群組建立大量部署。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBulkDeploymentStatus。

get-connectivity-info

以下程式碼範例顯示如何使用 get-connectivity-info。

AWS CLI

取得 Greengrass 核心的連線資訊

下列get-connectivity-info範例顯示裝置可用來連線到指定 Greengrass 核心的端點。連線資訊是 IP 地址或網域名稱的清單，其中包含對應的連接埠號碼和選用的客戶定義中繼資料。

```
aws greengrass get-connectivity-info \
    --thing-name "MyGroup_Core"
```

輸出：

```
{
    "ConnectivityInfo": [
        {
            "Metadata": "",
            "PortNumber": 8883,
            "HostAddress": "127.0.0.1",
            "Id": "AUTOIP_127.0.0.1_0"
        },
        {
            "Metadata": "",
            "PortNumber": 8883,
            "HostAddress": "192.168.1.3",
            "Id": "AUTOIP_192.168.1.3_1"
        },
        {
            "Metadata": "",
            "PortNumber": 8883,
            "HostAddress": "::1",
            "Id": "AUTOIP_::1_2"
        },
        {
            "Metadata": "",
            "PortNumber": 8883,
            "HostAddress": "fe80::1e69:ed93:f5b:f6d",
            "Id": "AUTOIP_fe80::1e69:ed93:f5b:f6d_3"
        }
    ]
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetConnectivityInfo。

get-connector-definition-version

以下程式碼範例顯示如何使用 get-connector-definition-version。

AWS CLI

擷取連接器定義特定版本的相關資訊

下列get-connector-definition-version範例會擷取指定連接器定義之指定版本的相關資訊。若要擷取連接器定義所有版本的 IDs，請使用 list-connector-definition-versions命令。若要擷取新增至連接器定義的最後一個版本的 ID，請使用 get-connector-definition命令並檢查 LatestVersion 屬性。

```
aws greengrass get-connector-definition-version \
    --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8" \
    --connector-definition-version-id "63c57963-c7c2-4a26-a7e2-7bf478ea2623"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/versions/63c57963-c7c2-4a26-
a7e2-7bf478ea2623",
    "CreationTimestamp": "2019-06-19T19:30:01.300Z",
    "Definition": {
        "Connectors": [
            {
                "ConnectorArn": "arn:aws:greengrass:us-west-2::/connectors/SNS/
versions/1",
                "Id": "MySNSConnector",
                "Parameters": {
                    "DefaultSNSArn": "arn:aws:sns:us-
west-2:123456789012:GGConnectorTopic"
                }
            }
        ]
    },
```

```
    "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
    "Version": "63c57963-c7c2-4a26-a7e2-7bf478ea2623"
}
```

如需詳細資訊，請參閱《AWS IoT Greengrass 開發人員指南》中的使用 Greengrass 連接器整合服務和通訊協定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetConnectorDefinitionVersion。

get-connector-definition

以下程式碼範例顯示如何使用 get-connector-definition。

AWS CLI

擷取連接器定義的相關資訊

下列get-connector-definition範例會擷取指定連接器定義的相關資訊。若要擷取連接器定義的 IDs，請使用 list-connector-definitions命令。

```
aws greengrass get-connector-definition \
    --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
    "CreationTimestamp": "2019-06-19T19:30:01.300Z",
    "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
    "LastUpdatedTimestamp": "2019-06-19T19:30:01.300Z",
    "LatestVersion": "63c57963-c7c2-4a26-a7e2-7bf478ea2623",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/versions/63c57963-
c7c2-4a26-a7e2-7bf478ea2623",
    "Name": "MySNSConnector",
    "tags": {}
}
```

如需詳細資訊，請參閱《AWS IoT Greengrass 開發人員指南》中的使用 Greengrass 連接器整合服務和通訊協定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetConnectorDefinition。

get-core-definition-version

以下程式碼範例顯示如何使用 get-core-definition-version。

AWS CLI

擷取 Greengrass 核心定義特定版本的詳細資訊

下列get-core-definition-version範例會擷取指定核心定義之指定版本的相關資訊。若要擷取核心定義所有版本的 IDs，請使用 list-core-definition-versions命令。若要擷取新增至核心定義的最後一個版本 ID，請使用 get-core-definition命令並檢查 LatestVersion 屬性。

```
aws greengrass get-core-definition-version \
    --core-definition-id "c906ed39-a1e3-4822-a981-7b9bd57b4b46"  \
    --core-definition-version-id "42aeeac3-fd9d-4312-a8fd-ffa9404a20e0"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/
c906ed39-a1e3-4822-a981-7b9bd57b4b46/versions/42aeeac3-fd9d-4312-a8fd-ffa9404a20e0",
    "CreationTimestamp": "2019-06-18T16:21:21.351Z",
    "Definition": {
        "Cores": [
            {
                "CertificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/928dea7b82331b47c3ff77b0e763fc5e64e2f7c884e6ef391baed9b6b8e21b45",
                "Id": "1a39aac7-0885-4417-91f6-23e4cea6c511",
                "SyncShadow": false,
                "ThingArn": "arn:aws:iot:us-west-2:123456789012:thing/
GGGroup4Pi3_Core"
            }
        ]
    },
    "Id": "c906ed39-a1e3-4822-a981-7b9bd57b4b46",
    "Version": "42aeeac3-fd9d-4312-a8fd-ffa9404a20e0"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCoreDefinitionVersion。

get-core-definition

以下程式碼範例顯示如何使用 get-core-definition。

AWS CLI

擷取 Greengrass 核心定義的詳細資訊

下列get-core-definition範例會擷取指定核心定義的相關資訊。若要擷取核心定義的 IDs，請使用 list-core-definitions命令。

```
aws greengrass get-core-definition \
    --core-definition-id "c906ed39-a1e3-4822-a981-7b9bd57b4b46"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
cores/237d6916-27cf-457f-ba0c-e86cfb5d25cd",
    "CreationTimestamp": "2018-10-18T04:47:06.721Z",
    "Id": "237d6916-27cf-457f-ba0c-e86cfb5d25cd",
    "LastUpdatedTimestamp": "2018-10-18T04:47:06.721Z",
    "LatestVersion": "bd2cd6d4-2bc5-468a-8962-39e071e34b68",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/237d6916-27cf-457f-ba0c-e86cfb5d25cd/versions/
bd2cd6d4-2bc5-468a-8962-39e071e34b68",
    "tags": {}
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCoreDefinition。

get-deployment-status

以下程式碼範例顯示如何使用 get-deployment-status。

AWS CLI

擷取部署的狀態

下列get-deployment-status範例會擷取指定 Greengrass 群組之指定部署的狀態。若要取得部署 ID，請使用 list-deployments命令並指定群組 ID。

```
aws greengrass get-deployment-status \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731" \
    --deployment-id "1065b8a0-812b-4f21-9d5d-e89b232a530f"
```

輸出：

```
{
    "DeploymentStatus": "Success",
    "DeploymentType": "NewDeployment",
    "UpdatedAt": "2019-06-18T17:04:44.761Z"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDeploymentStatus。

get-device-definition-version

以下程式碼範例顯示如何使用 get-device-definition-version。

AWS CLI

取得裝置定義版本

下列get-device-definition-version範例會擷取指定裝置定義之指定版本的相關資訊。若要擷取裝置定義所有版本的 IDs，請使用 list-device-definition-versions命令。若要擷取新增至裝置定義的最後一個版本 ID，請使用 get-device-definition命令並檢查 LatestVersion 屬性。

```
aws greengrass get-device-definition-version \
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \
    --device-definition-version-id "83c13984-6fed-447e-84d5-5b8aa45d5f71"
```

輸出：

```
{
    "Definition": {
        "Devices": [
            {
                "CertificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/6c52ce1b47bde88a637e9ccdd45fe4e4c2c0a75a6866f8f63d980ee22fa51e02",
```

```
                "ThingArn": "arn:aws:iot:us-west-2:123456789012:thing/
ExteriorTherm",
                "SyncShadow": true,
                "Id": "ExteriorTherm"
            },
            {
                "CertificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92",
                "ThingArn": "arn:aws:iot:us-west-2:123456789012:thing/
InteriorTherm",
                "SyncShadow": true,
                "Id": "InteriorTherm"
            }
        ]
    },
    "Version": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
    "CreationTimestamp": "2019-09-11T00:15:09.838Z",
    "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/83c13984-6fed-447e-84d5-5b8aa45d5f71"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDeviceDefinitionVersion。

get-device-definition

以下程式碼範例顯示如何使用 get-device-definition。

AWS CLI

取得裝置定義

下列get-device-definition範例會擷取指定裝置定義的相關資訊。若要擷取裝置定義的 IDs，請使用 list-device-definitions命令。

```
aws greengrass get-device-definition \
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

輸出：

```
{
```

```
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/83c13984-6fed-447e-84d5-5b8aa45d5f71",
    "Name": "TemperatureSensors",
    "tags": {},
    "LastUpdatedTimestamp": "2019-09-11T00:19:03.698Z",
    "LatestVersion": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
    "CreationTimestamp": "2019-09-11T00:11:06.197Z",
    "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDeviceDefinition。

get-function-definition-version

以下程式碼範例顯示如何使用 get-function-definition-version。

AWS CLI

擷取特定 Lambda 函數版本的詳細資訊

以下內容會get-function-definition-version擷取指定函數定義之指定版本的相關資訊。若要擷取函數定義所有版本的 IDs，請使用 list-function-definition-versions命令。若要擷取新增至函數定義的最後一個版本 ID，請使用 get-function-definition命令並檢查 LatestVersion 屬性。

```
aws greengrass get-function-definition-version \
    --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85" \
    --function-definition-version-id "9748fda7-1589-4fcc-ac94-f5559e88678b"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/versions/9748fda7-1589-4fcc-ac94-
f5559e88678b",
    "CreationTimestamp": "2019-06-18T17:04:30.776Z",
    "Definition": {
        "Functions": [
            {
```

```
                "FunctionArn": "arn:aws:lambda:::function:GGIPDetector:1",
                "FunctionConfiguration": {
                    "Environment": {},
                    "MemorySize": 32768,
                    "Pinned": true,
                    "Timeout": 3
                },
                "Id": "26b69bdb-e547-46bc-9812-84ec04b6cc8c"
            },
            {
                "FunctionArn": "arn:aws:lambda:us-
west-2:123456789012:function:Greengrass_HelloWorld:GG_HelloWorld",
                "FunctionConfiguration": {
                    "EncodingType": "json",
                    "Environment": {
                        "Variables": {}
                    },
                    "MemorySize": 16384,
                    "Pinned": true,
                    "Timeout": 25
                },
                "Id": "384465a8-eedf-48c6-b793-4c35f7bfae9b"
            }
        ]
    },
    "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
    "Version": "9748fda7-1589-4fcc-ac94-f5559e88678b"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFunctionDefinitionVersion。

get-function-definition

以下程式碼範例顯示如何使用 get-function-definition。

AWS CLI

擷取函數定義

下列get-function-definition範例顯示指定函數定義的詳細資訊。若要擷取函數定義的 IDs，請使用 list-function-definitions命令。

```
aws greengrass get-function-definition \
```

```
    --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
    "CreationTimestamp": "2019-06-18T16:21:21.431Z",
    "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
    "LastUpdatedTimestamp": "2019-06-18T16:21:21.431Z",
    "LatestVersion": "9748fda7-1589-4fcc-ac94-f5559e88678b",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/9748fda7-1589-4fcc-ac94-f5559e88678b",
    "tags": {}
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFunctionDefinition。

get-group-certificate-authority

以下程式碼範例顯示如何使用 get-group-certificate-authority。

AWS CLI

擷取與 Greengrass 群組相關聯的 CA

下列get-group-certificate-authority範例會擷取與指定 Greengrass 群組相關聯的憑證授權機構 (CA)。若要取得憑證授權機構 ID，請使用 list-group-certificate-authorities命令並指定群組 ID。

```
aws greengrass get-group-certificate-authority \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731" \
    --certificate-authority-
id "f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6"
```

輸出：

```
{
    "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/certificateauthorities/
f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",
```

```
    "GroupCertificateAuthorityId":
 "f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",
    "PemEncodedCertificate": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBWEXAMPLEGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDEXAMPLEEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAEXAMPLESBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jEXAMPLENMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0EXAMPLEBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWEXAMPLEDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWEXAMPLEgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5EXAMPLE8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CEXAMPLE93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswYEXAMPLEgpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKEXAMPLEAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----\n"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetGroupCertificateAuthority。

get-group-certificate-configuration

以下程式碼範例顯示如何使用 get-group-certificate-configuration。

AWS CLI

擷取 Greengrass 群組使用的憑證授權單位組態

下列get-group-certificate-configuration範例會擷取指定 Greengrass 群組所使用的憑證授權機構 (CA) 組態。

```
aws greengrass get-group-certificate-configuration \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

輸出：

```
{
    "CertificateAuthorityExpiryInMilliseconds": 2524607999000,
    "CertificateExpiryInMilliseconds": 604800000,
    "GroupId": "1013db12-8b58-45ff-acc7-704248f66731"
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetGroupCertificateConfiguration。

get-group-version

以下程式碼範例顯示如何使用 get-group-version。

AWS CLI

擷取 Greengrass 群組版本的相關資訊

下列get-group-version範例會擷取指定版本之指定群組的相關資訊。若要擷取群組所有版本的 IDs，請使用 list-group-versions命令。若要擷取新增至群組之最後一個版本的 ID，請使用 get-group命令並檢查 LatestVersion 屬性。

```
aws greengrass get-group-version \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"  \
    --group-version-id "115136b3-cfd7-4462-b77f-8741a4b00e5e"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-
b77f-8741a4b00e5e",
    "CreationTimestamp": "2019-06-18T17:04:30.915Z",
    "Definition": {
        "CoreDefinitionVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/c906ed39-a1e3-4822-a981-7b9bd57b4b46/versions/42aeeac3-
fd9d-4312-a8fd-ffa9404a20e0",
        "FunctionDefinitionVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/9748fda7-1589-4fcc-ac94-f5559e88678b",
        "SubscriptionDefinitionVersionArn": "arn:aws:greengrass:us-
west-2:123456789012:/greengrass/definition/subscriptions/70e49321-83d5-45d2-
bc09-81f4917ae152/versions/88ae8699-12ac-4663-ba3f-4d7f0519140b"
    },
    "Id": "1013db12-8b58-45ff-acc7-704248f66731",
    "Version": "115136b3-cfd7-4462-b77f-8741a4b00e5e"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetGroupVersion。

get-group

以下程式碼範例顯示如何使用 get-group。

AWS CLI

擷取 Greengrass 群組的相關資訊

下列get-group範例會擷取指定 Greengrass 群組的相關資訊。若要擷取 群組IDs，請使用 list-groups命令。

```
aws greengrass get-group \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731",
    "CreationTimestamp": "2019-06-18T16:21:21.457Z",
    "Id": "1013db12-8b58-45ff-acc7-704248f66731",
    "LastUpdatedTimestamp": "2019-06-18T16:21:21.457Z",
    "LatestVersion": "115136b3-cfd7-4462-b77f-8741a4b00e5e",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-
b77f-8741a4b00e5e",
    "Name": "GGGroup4Pi3",
    "tags": {}
}
```

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetGroup。

get-logger-definition-version

以下程式碼範例顯示如何使用 get-logger-definition-version。

AWS CLI

擷取有關記錄器定義版本的資訊

下列get-logger-definition-version範例會擷取指定版本之指定記錄器定義的相關資訊。若要擷取記錄器定義所有版本的 IDs，請使用 list-logger-definition-versions命令。若要擷取新增至記錄器定義的最後一個版本的 ID，請使用 get-logger-definition命令並檢查 LatestVersion 屬性。

```
aws greengrass get-logger-definition-version \
    --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23" \
    --logger-definition-version-id "5e3f6f64-a565-491e-8de0-3c0d8e0f2073"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/versions/5e3f6f64-
a565-491e-8de0-3c0d8e0f2073",
    "CreationTimestamp": "2019-05-08T16:10:13.866Z",
    "Definition": {
        "Loggers": []
    },
    "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
    "Version": "5e3f6f64-a565-491e-8de0-3c0d8e0f2073"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLoggerDefinitionVersion。

get-logger-definition

以下程式碼範例顯示如何使用 get-logger-definition。

AWS CLI

擷取記錄器定義的相關資訊

下列get-logger-definition範例會擷取指定記錄器定義的相關資訊。若要擷取記錄器定義的 IDs，請使用 list-logger-definitions命令。

```
aws greengrass get-logger-definition \
    --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23",
    "CreationTimestamp": "2019-05-08T16:10:13.809Z",
    "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
    "LastUpdatedTimestamp": "2019-05-08T16:10:13.809Z",
    "LatestVersion": "5e3f6f64-a565-491e-8de0-3c0d8e0f2073",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/versions/5e3f6f64-
a565-491e-8de0-3c0d8e0f2073",
    "tags": {}
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLoggerDefinition。

get-resource-definition-version

以下程式碼範例顯示如何使用 get-resource-definition-version。

AWS CLI

擷取特定版本資源定義的相關資訊

下列get-resource-definition-version範例會擷取指定版本之指定資源定義的相關資訊。若要擷取資源定義所有版本的 IDs，請使用 list-resource-definition-versions命令。若要擷取新增至資源定義的最後一個版本 ID，請使用 get-resource-definition命令並檢查 LatestVersion 屬性。

```
aws greengrass get-resource-definition-version \
    --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658" \
    --resource-definition-version-id "26e8829a-491a-464d-9c87-664bf6f6f2be"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/26e8829a-491a-464d-9c87-664bf6f6f2be",
    "CreationTimestamp": "2019-06-19T16:40:59.392Z",
    "Definition": {
        "Resources": [
```

```
            {
                "Id": "26ff3f7b-839a-4217-9fdc-a218308b3963",
                "Name": "usb-port",
                "ResourceDataContainer": {
                    "LocalDeviceResourceData": {
                        "GroupOwnerSetting": {
                            "AutoAddGroupOwner": false
                        },
                        "SourcePath": "/dev/bus/usb"
                    }
                }
            }
        ]
    },
    "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
    "Version": "26e8829a-491a-464d-9c87-664bf6f6f2be"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResourceDefinitionVersion。

get-resource-definition

以下程式碼範例顯示如何使用 get-resource-definition。

AWS CLI

擷取資源定義的相關資訊

下列get-resource-definition範例會擷取指定資源定義的相關資訊。若要擷取資源定義的 IDs，請使用 list-resource-definitions命令。

```
aws greengrass get-resource-definition \
    --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658",
    "CreationTimestamp": "2019-06-19T16:40:59.261Z",
    "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
    "LastUpdatedTimestamp": "2019-06-19T16:40:59.261Z",
```

```
    "LatestVersion": "26e8829a-491a-464d-9c87-664bf6f6f2be",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/26e8829a-491a-464d-9c87-664bf6f6f2be",
    "tags": {}
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResourceDefinition。

get-service-role-for-account

以下程式碼範例顯示如何使用 get-service-role-for-account。

AWS CLI

擷取連接至您帳戶之服務角色的詳細資訊

下列get-service-role-for-account範例會擷取連接到您 AWS 帳戶之服務角色的相關資訊。

```
aws greengrass get-service-role-for-account
```

輸出：

```
{
    "AssociatedAt": "2018-10-18T15:59:20Z",
    "RoleArn": "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的 Greengrass 服務角色。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetServiceRoleForAccount。

get-subscription-definition-version

以下程式碼範例顯示如何使用 get-subscription-definition-version。

AWS CLI

擷取訂閱定義特定版本的相關資訊

下列get-subscription-definition-version範例會擷取指定版本之指定訂閱定義的相關資訊。若要擷取訂閱定義所有版本的 IDs，請使用 list-subscription-definition-versions命令。若要擷取新增至訂閱定義之最後一個版本的 ID，請使用 get-subscription-definition命令並檢查 LatestVersion 屬性。

```
aws greengrass get-subscription-definition-version \
    --subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152" \
    --subscription-definition-version-id "88ae8699-12ac-4663-ba3f-4d7f0519140b"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/versions/88ae8699-12ac-4663-
ba3f-4d7f0519140b",
    "CreationTimestamp": "2019-06-18T17:03:52.499Z",
    "Definition": {
        "Subscriptions": [
            {
                "Id": "692c4484-d89f-4f64-8edd-1a041a65e5b6",
                "Source": "arn:aws:lambda:us-
west-2:123456789012:function:Greengrass_HelloWorld:GG_HelloWorld",
                "Subject": "hello/world",
                "Target": "cloud"
            }
        ]
    },
    "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
    "Version": "88ae8699-12ac-4663-ba3f-4d7f0519140b"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSubscriptionDefinitionVersion。

get-subscription-definition

以下程式碼範例顯示如何使用 get-subscription-definition。

AWS CLI

擷取訂閱定義的相關資訊

下列get-subscription-definition範例會擷取指定訂閱定義的相關資訊。若要擷取訂閱定義的 IDs，請使用 list-subscription-definitions命令。

```
aws greengrass get-subscription-definition \
    --subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152"
```

輸出：

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152",
    "CreationTimestamp": "2019-06-18T17:03:52.392Z",
    "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
    "LastUpdatedTimestamp": "2019-06-18T17:03:52.392Z",
    "LatestVersion": "88ae8699-12ac-4663-ba3f-4d7f0519140b",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/
versions/88ae8699-12ac-4663-ba3f-4d7f0519140b",
    "tags": {}
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSubscriptionDefinition。

get-thing-runtime-configuration

以下程式碼範例顯示如何使用 get-thing-runtime-configuration。

AWS CLI

擷取 Greengrass 核心的執行期組態

下列get-thing-runtime-configuration範例會擷取 Greengrass 核心的執行時間組態。在擷取執行期組態之前，您必須使用 update-thing-runtime-configuration命令來建立核心的執行期組態。

```
aws greengrass get-thing-runtime-configuration \
    --thing-name SampleGreengrassCore
```

輸出：

```
{
```

```
    "RuntimeConfiguration": {
        "TelemetryConfiguration": {
            "ConfigurationSyncStatus": "OutOfSync",
            "Telemetry": "On"
        }
    }
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的設定遙測設定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetThingRuntimeConfiguration。

`list-bulk-deployment-detailed-reports`

以下程式碼範例顯示如何使用 `list-bulk-deployment-detailed-reports`。

AWS CLI

列出大量部署中個別部署的相關資訊

下列`list-bulk-deployment-detailed-reports`範例顯示大量部署操作中個別部署的相關資訊，包括狀態。

```
aws greengrass list-bulk-deployment-detailed-reports \
    --bulk-deployment-id 42ce9c42-489b-4ed4-b905-8996aa50ef9d
```

輸出：

```
{
    "Deployments": [
        {
            "DeploymentType": "NewDeployment",
            "DeploymentStatus": "Success",
            "DeploymentId": "123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333/
deployments/123456789012:123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "GroupArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333/
versions/123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
            "CreatedAt": "2020-01-21T21:34:16.501Z"
        },
        {
```

```
            "DeploymentType": "NewDeployment",
            "DeploymentStatus": "InProgress",
            "DeploymentId": "123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "DeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/a1b2c3d4-5678-90ab-cdef-EXAMPLE55555/
deployments/123456789012:123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "GroupArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/a1b2c3d4-5678-90ab-cdef-EXAMPLE55555/versions/a1b2c3d4-5678-90ab-cdef-
EXAMPLE66666",
            "CreatedAt": "2020-01-21T21:34:16.486Z"
        },
        ...
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的為群組建立大量部署。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBulkDeploymentDetailedReports。

`list-bulk-deployments`

以下程式碼範例顯示如何使用 `list-bulk-deployments`。

AWS CLI

列出大量部署

下列`list-bulk-deployments`範例列出所有大量部署。

```
aws greengrass list-bulk-deployments
```

輸出：

```
{
    "BulkDeployments": [
        {
            "BulkDeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/bulk/deployments/870fb41b-6288-4e0c-bc76-a7ba4b4d3267",
            "BulkDeploymentId": "870fb41b-6288-4e0c-bc76-a7ba4b4d3267",
            "CreatedAt": "2019-06-25T16:11:33.265Z"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的為群組建立大量部署。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBulkDeployments。

list-connector-definition-versions

以下程式碼範例顯示如何使用 list-connector-definition-versions。

AWS CLI

列出可用於連接器定義的版本

下列list-connector-definition-versions範例列出可用於指定連接器定義的版本。使用list-connector-definitions命令來取得連接器定義 ID。

```
aws greengrass list-connector-definition-versions \
    --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

輸出：

```
{
    "Versions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/versions/63c57963-
c7c2-4a26-a7e2-7bf478ea2623",
            "CreationTimestamp": "2019-06-19T19:30:01.300Z",
            "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
            "Version": "63c57963-c7c2-4a26-a7e2-7bf478ea2623"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的使用 Greengrass 連接器整合服務和通訊協定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListConnectorDefinitionVersions。

list-connector-definitions

以下程式碼範例顯示如何使用 list-connector-definitions。

AWS CLI

列出已定義的 Greengrass 連接器

下列`list-connector-definitions`範例列出為 AWS 您的帳戶定義的所有 Greengrass 連接器。

```
aws greengrass list-connector-definitions
```

輸出：

```
{
    "Definitions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
            "CreationTimestamp": "2019-06-19T19:30:01.300Z",
            "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
            "LastUpdatedTimestamp": "2019-06-19T19:30:01.300Z",
            "LatestVersion": "63c57963-c7c2-4a26-a7e2-7bf478ea2623",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/
versions/63c57963-c7c2-4a26-a7e2-7bf478ea2623",
            "Name": "MySNSConnector"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS IoT Greengrass 開發人員指南》中的使用 Greengrass 連接器整合服務和通訊協定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListConnectorDefinitions。

`list-core-definition-versions`

以下程式碼範例顯示如何使用 `list-core-definition-versions`。

AWS CLI

列出 Greengrass 核心定義的版本

下列`list-core-definitions`範例列出指定 Greengrass 核心定義的所有版本。您可以使用`list-core-definitions`命令來取得版本 ID。

```
aws greengrass list-core-definition-versions \
    --core-definition-id "eaf280cb-138c-4d15-af36-6f681a1348f7"
```

輸出：

```
{
    "Versions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/cores/eaf280cb-138c-4d15-af36-6f681a1348f7/versions/467c36e4-c5da-440c-
a97b-084e62593b4c",
            "CreationTimestamp": "2019-06-18T16:14:17.709Z",
            "Id": "eaf280cb-138c-4d15-af36-6f681a1348f7",
            "Version": "467c36e4-c5da-440c-a97b-084e62593b4c"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCoreDefinitionVersions。

list-core-definitions

以下程式碼範例顯示如何使用 list-core-definitions。

AWS CLI

列出 Greengrass 核心定義

下列list-core-definitions範例列出您 AWS 帳戶的所有 Greengrass 核心定義。

```
aws greengrass list-core-definitions
```

輸出：

```
{
    "Definitions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/cores/0507843c-c1ef-4f06-b051-817030df7e7d",
            "CreationTimestamp": "2018-10-17T04:30:32.786Z",
```

```
            "Id": "0507843c-c1ef-4f06-b051-817030df7e7d",
            "LastUpdatedTimestamp": "2018-10-17T04:30:32.786Z",
            "LatestVersion": "bcdf9e86-3793-491e-93af-3cdfbf4e22b7",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/0507843c-c1ef-4f06-b051-817030df7e7d/versions/
bcdf9e86-3793-491e-93af-3cdfbf4e22b7"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/cores/31c22500-3509-4271-bafd-cf0655cda438",
            "CreationTimestamp": "2019-06-18T16:24:16.064Z",
            "Id": "31c22500-3509-4271-bafd-cf0655cda438",
            "LastUpdatedTimestamp": "2019-06-18T16:24:16.064Z",
            "LatestVersion": "2f350395-6d09-4c8a-8336-9ae5b57ace84",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/31c22500-3509-4271-bafd-cf0655cda438/
versions/2f350395-6d09-4c8a-8336-9ae5b57ace84"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/cores/c906ed39-a1e3-4822-a981-7b9bd57b4b46",
            "CreationTimestamp": "2019-06-18T16:21:21.351Z",
            "Id": "c906ed39-a1e3-4822-a981-7b9bd57b4b46",
            "LastUpdatedTimestamp": "2019-06-18T16:21:21.351Z",
            "LatestVersion": "42aeeac3-fd9d-4312-a8fd-ffa9404a20e0",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/c906ed39-a1e3-4822-a981-7b9bd57b4b46/versions/42aeeac3-
fd9d-4312-a8fd-ffa9404a20e0"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/cores/eaf280cb-138c-4d15-af36-6f681a1348f7",
            "CreationTimestamp": "2019-06-18T16:14:17.709Z",
            "Id": "eaf280cb-138c-4d15-af36-6f681a1348f7",
            "LastUpdatedTimestamp": "2019-06-18T16:14:17.709Z",
            "LatestVersion": "467c36e4-c5da-440c-a97b-084e62593b4c",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/eaf280cb-138c-4d15-af36-6f681a1348f7/versions/467c36e4-
c5da-440c-a97b-084e62593b4c"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCoreDefinitions。

`list-deployments`

以下程式碼範例顯示如何使用 `list-deployments`。

AWS CLI

列出 Greengrass 群組的部署

下列`list-deployments`範例列出指定 Greengrass 群組的部署。您可以使用 `list-groups`命令來查詢您的群組 ID。

```
aws greengrass list-deployments \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

輸出：

```
{
    "Deployments": [
        {
            "CreatedAt": "2019-06-18T17:04:32.702Z",
            "DeploymentId": "1065b8a0-812b-4f21-9d5d-e89b232a530f",
            "DeploymentType": "NewDeployment",
            "GroupArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-
b77f-8741a4b00e5e"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDeployments。

`list-device-definition-versions`

以下程式碼範例顯示如何使用 `list-device-definition-versions`。

AWS CLI

列出裝置定義的版本

下列`list-device-definition-versions`範例顯示與指定裝置定義相關聯的裝置定義版本。

```
aws greengrass list-device-definition-versions \
```

```
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

輸出：

```
{
    "Versions": [
        {
            "Version": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
            "CreationTimestamp": "2019-09-11T00:15:09.838Z",
            "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/83c13984-6fed-447e-84d5-5b8aa45d5f71"
        },
        {
            "Version": "3b5cc510-58c1-44b5-9d98-4ad858ffa795",
            "CreationTimestamp": "2019-09-11T00:11:06.197Z",
            "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/3b5cc510-58c1-44b5-9d98-4ad858ffa795"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDeviceDefinitionVersions。

list-device-definitions

以下程式碼範例顯示如何使用 list-device-definitions。

AWS CLI

列出您的裝置定義

下列list-device-definitions範例顯示您 AWS 帳戶中指定 AWS 區域中裝置定義的詳細資訊。

```
aws greengrass list-device-definitions \
    --region us-west-2
```

輸出：

```
{
    "Definitions": [
        {
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/devices/50f3274c-3f0a-4f57-b114-6f46085281ab/versions/
c777b0f5-1059-449b-beaa-f003ebc56c34",
            "LastUpdatedTimestamp": "2019-06-14T15:42:09.059Z",
            "LatestVersion": "c777b0f5-1059-449b-beaa-f003ebc56c34",
            "CreationTimestamp": "2019-06-14T15:42:09.059Z",
            "Id": "50f3274c-3f0a-4f57-b114-6f46085281ab",
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/devices/50f3274c-3f0a-4f57-b114-6f46085281ab"
        },
        {
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/devices/e01951c9-6134-479a-969a-1a15cac11c40/
versions/514d57aa-4ee6-401c-9fac-938a9f7a51e5",
            "Name": "TestDeviceDefinition",
            "LastUpdatedTimestamp": "2019-04-16T23:17:43.245Z",
            "LatestVersion": "514d57aa-4ee6-401c-9fac-938a9f7a51e5",
            "CreationTimestamp": "2019-04-16T23:17:43.245Z",
            "Id": "e01951c9-6134-479a-969a-1a15cac11c40",
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/devices/e01951c9-6134-479a-969a-1a15cac11c40"
        },
        {
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/83c13984-6fed-447e-84d5-5b8aa45d5f71",
            "Name": "TemperatureSensors",
            "LastUpdatedTimestamp": "2019-09-10T00:19:03.698Z",
            "LatestVersion": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
            "CreationTimestamp": "2019-09-11T00:11:06.197Z",
            "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDeviceDefinitions。

list-function-definition-versions

以下程式碼範例顯示如何使用 list-function-definition-versions。

AWS CLI

列出 Lambda 函數的版本

下列list-function-definition-versions範例列出指定 Lambda 函數的所有版本。您可以使用 list-function-definitions命令來取得 ID。

```
aws greengrass list-function-definition-versions \
    --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85"
```

輸出：

```
{
    "Versions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/9748fda7-1589-4fcc-ac94-f5559e88678b",
            "CreationTimestamp": "2019-06-18T17:04:30.776Z",
            "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
            "Version": "9748fda7-1589-4fcc-ac94-f5559e88678b"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/9b08df77-26f2-4c29-93d2-769715edcfec",
            "CreationTimestamp": "2019-06-18T17:02:44.087Z",
            "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
            "Version": "9b08df77-26f2-4c29-93d2-769715edcfec"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/4236239f-94f7-4b90-a2f8-2a24c829d21e",
            "CreationTimestamp": "2019-06-18T17:01:42.284Z",
            "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
            "Version": "4236239f-94f7-4b90-a2f8-2a24c829d21e"
        },
        {
```

```
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/343408bb-549a-4fbe-b043-853643179a39",
            "CreationTimestamp": "2019-06-18T16:21:21.431Z",
            "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
            "Version": "343408bb-549a-4fbe-b043-853643179a39"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFunctionDefinitionVersions。

list-function-definitions

以下程式碼範例顯示如何使用 list-function-definitions。

AWS CLI

列出 Lambda 函數

下列list-function-definitions範例列出為 AWS 您的帳戶定義的所有 Lambda 函數。

```
aws greengrass list-function-definitions
```

輸出：

```
{
    "Definitions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/017970a5-8952-46dd-b1c1-020b3ae8e960",
            "CreationTimestamp": "2018-10-17T04:30:32.884Z",
            "Id": "017970a5-8952-46dd-b1c1-020b3ae8e960",
            "LastUpdatedTimestamp": "2018-10-17T04:30:32.884Z",
            "LatestVersion": "4380b302-790d-4ed8-92bf-02e88afecb15",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/017970a5-8952-46dd-b1c1-020b3ae8e960/
versions/4380b302-790d-4ed8-92bf-02e88afecb15"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
```

```
            "CreationTimestamp": "2019-06-18T16:21:21.431Z",
            "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
            "LastUpdatedTimestamp": "2019-06-18T16:21:21.431Z",
            "LatestVersion": "9748fda7-1589-4fcc-ac94-f5559e88678b",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/9748fda7-1589-4fcc-ac94-f5559e88678b"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/6598e653-a262-440c-9967-e2697f64da7b",
            "CreationTimestamp": "2019-06-18T16:24:16.123Z",
            "Id": "6598e653-a262-440c-9967-e2697f64da7b",
            "LastUpdatedTimestamp": "2019-06-18T16:24:16.123Z",
            "LatestVersion": "38bc6ccd-98a2-4ce7-997e-16c84748fae4",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/6598e653-a262-440c-9967-e2697f64da7b/
versions/38bc6ccd-98a2-4ce7-997e-16c84748fae4"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/c668df84-fad2-491b-95f4-655d2cad7885",
            "CreationTimestamp": "2019-06-18T16:14:17.784Z",
            "Id": "c668df84-fad2-491b-95f4-655d2cad7885",
            "LastUpdatedTimestamp": "2019-06-18T16:14:17.784Z",
            "LatestVersion": "37dd68c4-a64f-40ba-aa13-71fecc3ebded",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/c668df84-fad2-491b-95f4-655d2cad7885/
versions/37dd68c4-a64f-40ba-aa13-71fecc3ebded"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFunctionDefinitions。

`list-group-certificate-authorities`

以下程式碼範例顯示如何使用 `list-group-certificate-authorities`。

AWS CLI

列出群組的目前 CAs

下列list-group-certificate-authorities範例列出指定 Greengrass 群組的目前憑證授權單位 (CAs)。

```
aws greengrass list-group-certificate-authorities \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

輸出：

```
{
    "GroupCertificateAuthorities": [
        {
            "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-
west-2:123456789012:/greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/
certificateauthorities/
f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",
            "GroupCertificateAuthorityId":
 "f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListGroupCertificateAuthorities。

list-group-versions

以下程式碼範例顯示如何使用 list-group-versions。

AWS CLI

列出 Greengrass 群組的版本

下列list-group-versions範例列出指定 Greengrass 群組的版本。

```
aws greengrass list-group-versions \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

輸出：

```
{
    "Versions": [
```

```
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-
b77f-8741a4b00e5e",
            "CreationTimestamp": "2019-06-18T17:04:30.915Z",
            "Id": "1013db12-8b58-45ff-acc7-704248f66731",
            "Version": "115136b3-cfd7-4462-b77f-8741a4b00e5e"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/versions/4340669d-
d14d-44e3-920c-46c928750750",
            "CreationTimestamp": "2019-06-18T17:03:52.663Z",
            "Id": "1013db12-8b58-45ff-acc7-704248f66731",
            "Version": "4340669d-d14d-44e3-920c-46c928750750"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/
versions/1b06e099-2d5b-4f10-91b9-78c4e060f5da",
            "CreationTimestamp": "2019-06-18T17:02:44.189Z",
            "Id": "1013db12-8b58-45ff-acc7-704248f66731",
            "Version": "1b06e099-2d5b-4f10-91b9-78c4e060f5da"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/2d3f27f1-3b43-4554-
ab7a-73ec30477efe",
            "CreationTimestamp": "2019-06-18T17:01:42.401Z",
            "Id": "1013db12-8b58-45ff-acc7-704248f66731",
            "Version": "2d3f27f1-3b43-4554-ab7a-73ec30477efe"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/d20f7ae9-3444-4c1c-b025-
e2ede23cdd31",
            "CreationTimestamp": "2019-06-18T16:21:21.457Z",
            "Id": "1013db12-8b58-45ff-acc7-704248f66731",
            "Version": "d20f7ae9-3444-4c1c-b025-e2ede23cdd31"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListGroupVersions。

list-groups

以下程式碼範例顯示如何使用 list-groups。

AWS CLI

列出 Greengrass 群組

下列list-groups範例列出您 AWS 帳戶中定義的所有 Greengrass 群組。

```
aws greengrass list-groups
```

輸出：

```
{
    "Groups": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731",
            "CreationTimestamp": "2019-06-18T16:21:21.457Z",
            "Id": "1013db12-8b58-45ff-acc7-704248f66731",
            "LastUpdatedTimestamp": "2019-06-18T16:21:21.457Z",
            "LatestVersion": "115136b3-cfd7-4462-b77f-8741a4b00e5e",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-
b77f-8741a4b00e5e",
            "Name": "GGGroup4Pi3"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1402daf9-71cf-4cfe-8be0-d5e80526d0d8",
            "CreationTimestamp": "2018-10-31T21:52:46.603Z",
            "Id": "1402daf9-71cf-4cfe-8be0-d5e80526d0d8",
            "LastUpdatedTimestamp": "2018-10-31T21:52:46.603Z",
            "LatestVersion": "749af901-60ab-456f-a096-91b12d983c29",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1402daf9-71cf-4cfe-8be0-d5e80526d0d8/versions/749af901-60ab-456f-
a096-91b12d983c29",
            "Name": "MyTestGroup"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/504b5c8d-bbed-4635-aff1-48ec5b586db5",
```

```
            "CreationTimestamp": "2018-12-31T21:39:36.771Z",
            "Id": "504b5c8d-bbed-4635-aff1-48ec5b586db5",
            "LastUpdatedTimestamp": "2018-12-31T21:39:36.771Z",
            "LatestVersion": "46911e8e-f9bc-4898-8b63-59c7653636ec",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/504b5c8d-bbed-4635-aff1-48ec5b586db5/versions/46911e8e-
f9bc-4898-8b63-59c7653636ec",
            "Name": "smp-ggrass-group"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListGroups。

list-logger-definition-versions

以下程式碼範例顯示如何使用 list-logger-definition-versions。

AWS CLI

取得記錄器定義的版本清單

下列list-logger-definition-versions範例會取得指定記錄器定義之所有版本的清單。

```
aws greengrass list-logger-definition-versions \
    --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23"
```

輸出：

```
{
    "Versions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/versions/5e3f6f64-
a565-491e-8de0-3c0d8e0f2073",
            "CreationTimestamp": "2019-05-08T16:10:13.866Z",
            "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
            "Version": "5e3f6f64-a565-491e-8de0-3c0d8e0f2073"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/versions/3ec6d3af-eb85-48f9-
a16d-1c795fe696d7",
```

```
            "CreationTimestamp": "2019-05-08T16:10:13.809Z",
            "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
            "Version": "3ec6d3af-eb85-48f9-a16d-1c795fe696d7"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListLoggerDefinitionVersions。

list-logger-definitions

以下程式碼範例顯示如何使用 list-logger-definitions。

AWS CLI

取得記錄器定義的清單

下列list-logger-definitions範例列出您 AWS 帳戶的所有記錄器定義。

```
aws greengrass list-logger-definitions
```

輸出：

```
{
    "Definitions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23",
            "CreationTimestamp": "2019-05-08T16:10:13.809Z",
            "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
            "LastUpdatedTimestamp": "2019-05-08T16:10:13.809Z",
            "LatestVersion": "5e3f6f64-a565-491e-8de0-3c0d8e0f2073",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/
versions/5e3f6f64-a565-491e-8de0-3c0d8e0f2073"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListLoggerDefinitions。

list-resource-definition-versions

以下程式碼範例顯示如何使用 list-resource-definition-versions。

AWS CLI

列出資源定義的版本

下列list-resource-definition-versions範例列出指定 Greengrass 資源的版本。

```
aws greengrass list-resource-definition-versions \
    --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

輸出：

```
{
    "Versions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/26e8829a-491a-464d-9c87-664bf6f6f2be",
            "CreationTimestamp": "2019-06-19T16:40:59.392Z",
            "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
            "Version": "26e8829a-491a-464d-9c87-664bf6f6f2be"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/432d92f6-12de-4ec9-a704-619a942a62aa",
            "CreationTimestamp": "2019-06-19T16:40:59.261Z",
            "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
            "Version": "432d92f6-12de-4ec9-a704-619a942a62aa"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourceDefinitionVersions。

list-resource-definitions

以下程式碼範例顯示如何使用 list-resource-definitions。

AWS CLI

列出定義的資源

下列`list-resource-definitions`範例列出為 AWS IoT Greengrass 定義的要使用的資源。

```
aws greengrass list-resource-definitions
```

輸出：

```
{
    "Definitions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658",
            "CreationTimestamp": "2019-06-19T16:40:59.261Z",
            "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
            "LastUpdatedTimestamp": "2019-06-19T16:40:59.261Z",
            "LatestVersion": "26e8829a-491a-464d-9c87-664bf6f6f2be",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/26e8829a-491a-464d-9c87-664bf6f6f2be"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",
            "CreationTimestamp": "2019-06-19T21:51:28.212Z",
            "Id": "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",
            "LastUpdatedTimestamp": "2019-06-19T21:51:28.212Z",
            "LatestVersion": "a5f94d0b-f6bc-40f4-bb78-7a1c5fe13ba1",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38/versions/
a5f94d0b-f6bc-40f4-bb78-7a1c5fe13ba1",
            "Name": "MyGreengrassResources"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourceDefinitions。

list-subscription-definition-versions

以下程式碼範例顯示如何使用 list-subscription-definition-versions。

AWS CLI

列出訂閱定義的版本

下列list-subscription-definition-versions範例列出指定訂閱的所有版本。您可以使用list-subscription-definitions命令來查詢訂閱 ID。

```
aws greengrass list-subscription-definition-versions \
    --subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152"
```

輸出：

```
{
    "Versions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/
versions/88ae8699-12ac-4663-ba3f-4d7f0519140b",
            "CreationTimestamp": "2019-06-18T17:03:52.499Z",
            "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
            "Version": "88ae8699-12ac-4663-ba3f-4d7f0519140b"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/versions/7e320ba3-
c369-4069-a2f0-90acb7f219d6",
            "CreationTimestamp": "2019-06-18T17:03:52.392Z",
            "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
            "Version": "7e320ba3-c369-4069-a2f0-90acb7f219d6"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSubscriptionDefinitionVersions。

list-subscription-definitions

以下程式碼範例顯示如何使用 list-subscription-definitions。

AWS CLI

若要取得清單訂閱定義

下列`list-subscription-definitions`範例列出您 AWS 帳戶中定義的所有 AWS IoT Greengrass 訂閱。

```
aws greengrass list-subscription-definitions
```

輸出：

```
{
    "Definitions": [
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152",
            "CreationTimestamp": "2019-06-18T17:03:52.392Z",
            "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
            "LastUpdatedTimestamp": "2019-06-18T17:03:52.392Z",
            "LatestVersion": "88ae8699-12ac-4663-ba3f-4d7f0519140b",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/
versions/88ae8699-12ac-4663-ba3f-4d7f0519140b"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/subscriptions/cd6f1c37-d9a4-4e90-be94-01a7404f5967",
            "CreationTimestamp": "2018-10-18T15:45:34.024Z",
            "Id": "cd6f1c37-d9a4-4e90-be94-01a7404f5967",
            "LastUpdatedTimestamp": "2018-10-18T15:45:34.024Z",
            "LatestVersion": "d1cf8fac-284f-4f6a-98fe-a2d36d089373",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/subscriptions/cd6f1c37-d9a4-4e90-be94-01a7404f5967/versions/
d1cf8fac-284f-4f6a-98fe-a2d36d089373"
        },
        {
            "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/subscriptions/fa81bc84-3f59-4377-a84b-5d0134da359b",
            "CreationTimestamp": "2018-10-22T17:09:31.429Z",
            "Id": "fa81bc84-3f59-4377-a84b-5d0134da359b",
            "LastUpdatedTimestamp": "2018-10-22T17:09:31.429Z",
            "LatestVersion": "086d1b08-b25a-477c-a16f-6f9b3a9c295a",
```

```
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/subscriptions/fa81bc84-3f59-4377-a84b-5d0134da359b/
versions/086d1b08-b25a-477c-a16f-6f9b3a9c295a"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSubscriptionDefinitions。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出連接至資源的標籤

下列list-tags-for-resource範例列出連接到指定資源的標籤及其值。

```
aws greengrass list-tags-for-resource \
    --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

輸出：

```
{
    "tags": {
        "ResourceSubType": "USB",
        "ResourceType": "Device"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的標記您的 Greengrass 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

reset-deployments

以下程式碼範例顯示如何使用 reset-deployments。

AWS CLI

清除 Greengrass 群組的部署資訊

下列reset-deployments範例會清除指定 Greengrass 群組的部署資訊。當您新增 時--force option，部署資訊會重設，而無需等待核心裝置回應。

```
aws greengrass reset-deployments \
    --group-id "1402daf9-71cf-4cfe-8be0-d5e80526d0d8" \
    --force
```

輸出：

```
{
    "DeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1402daf9-71cf-4cfe-8be0-d5e80526d0d8/
deployments/7dd4e356-9882-46a3-9e28-6d21900c011a",
    "DeploymentId": "7dd4e356-9882-46a3-9e28-6d21900c011a"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的重設部署。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetDeployments。

start-bulk-deployment

以下程式碼範例顯示如何使用 start-bulk-deployment。

AWS CLI

啟動大量部署操作

下列start-bulk-deployment範例使用存放在 S3 儲存貯體中的檔案來指定要部署的群組，以啟動大量部署操作。

```
aws greengrass start-bulk-deployment \
    --cli-input-json "{\"InputFileUri\":\"https://gg-group-deployment1.s3-us-
west-2.amazonaws.com/MyBulkDeploymentInputFile.txt\", \"ExecutionRoleArn\":
\"arn:aws:iam::123456789012:role/ggCreateDeploymentRole\",\"AmznClientToken\":
\"yourAmazonClientToken\"}"
```

輸出：

```
{
    "BulkDeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
bulk/deployments/870fb41b-6288-4e0c-bc76-a7ba4b4d3267",
    "BulkDeploymentId": "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的為群組建立大量部署。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartBulkDeployment。

stop-bulk-deployment

以下程式碼範例顯示如何使用 stop-bulk-deployment。

AWS CLI

停止大量部署

下列stop-bulk-deployment範例會停止指定的大量部署。如果您嘗試停止已完成的大量部署，您會收到錯誤： InvalidInputException: Cannot change state of finished execution.

```
aws greengrass stop-bulk-deployment \
    --bulk-deployment-id "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的為群組建立大量部署。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopBulkDeployment。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤套用至資源

下列tag-resource範例會將兩個標籤 ResourceType和 ResourceSubType套用至指定的 Greengrass 資源。此操作可以同時新增標籤和值，或更新現有標籤的值。使用 untag-resource命令移除標籤。

```
aws greengrass tag-resource \
    --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658" \
    --tags "ResourceType=Device,ResourceSubType=USB"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的標記您的 Greengrass 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源移除標籤及其值

下列untag-resource範例會移除金鑰Category來自指定 Greengrass 群組的標籤。如果指定的資源Category不存在金鑰，則不會傳回錯誤。

```
aws greengrass untag-resource \
    --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731" \
    --tag-keys "Category"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的標記您的 Greengrass 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-connectivity-info

以下程式碼範例顯示如何使用 update-connectivity-info。

AWS CLI

更新 Greengrass 核心的連線資訊

下列update-connectivity-info範例會變更裝置可用來連線到指定 Greengrass 核心的端點。連線資訊是 IP 地址或網域名稱的清單，其中包含對應的連接埠號碼和選用的客戶定義中繼資料。當本機網路變更時，您可能需要更新連線資訊。

```
aws greengrass update-connectivity-info \
    --thing-name "MyGroup_Core" \
    --connectivity-info "[{\"Metadata\":\"\",\"PortNumber\":8883,\"HostAddress\":
\"127.0.0.1\",\"Id\":\"localhost_127.0.0.1_0\"},{\"Metadata\":\"\",\"PortNumber
\":8883,\"HostAddress\":\"192.168.1.3\",\"Id\":\"localIP_192.168.1.3\"}]"
```

輸出：

```
{
    "Version": "312de337-59af-4cf9-a278-2a23bd39c300"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateConnectivityInfo。

update-connector-definition

以下程式碼範例顯示如何使用 update-connector-definition。

AWS CLI

更新連接器定義的名稱

下列update-connector-definition範例會更新指定連接器定義的名稱。如果您想要更新連接器的詳細資訊，請使用 create-connector-definition-version命令來建立新的版本。

```
aws greengrass update-connector-definition \
    --connector-definition-id "55d0052b-0d7d-44d6-b56f-21867215e118" \
    --name "GreengrassConnectors2019"
```

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的使用連接器整合服務和通訊協定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateConnectorDefinition。

update-core-definition

以下程式碼範例顯示如何使用 update-core-definition。

AWS CLI

更新核心定義

下列update-core-definition範例會變更指定核心定義的名稱。您只能更新核心定義的 name 屬性。

```
aws greengrass update-core-definition \
    --core-definition-id "582efe12-b05a-409e-9a24-a2ba1bcc4a12" \
    --name "MyCoreDevices"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的設定 AWS IoT Greengrass 核心。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateCoreDefinition。

update-device-definition

以下程式碼範例顯示如何使用 update-device-definition。

AWS CLI

更新裝置定義

下列update-device-definition範例會變更指定裝置定義的名稱。您只能更新裝置定義的 name 屬性。

```
aws greengrass update-device-definition \
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \
    --name "TemperatureSensors"
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDeviceDefinition。

update-function-definition

以下程式碼範例顯示如何使用 update-function-definition。

AWS CLI

更新函數定義的名稱

下列update-function-definition範例會更新指定函數定義的名稱。如果您想要更新函數的詳細資訊，請使用 create-function-definition-version命令來建立新的版本。

```
aws greengrass update-function-definition \
    --function-definition-id "e47952bd-dea9-4e2c-a7e1-37bbe8807f46" \
    --name ObsoleteFunction
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的執行本機 Lambda 函數。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateFunctionDefinition。

update-group-certificate-configuration

以下程式碼範例顯示如何使用 update-group-certificate-configuration。

AWS CLI

更新群組憑證的到期時間

下列update-group-certificate-configuration範例會為指定群組產生的憑證設定 10 天的到期時間。

```
aws greengrass update-group-certificate-configuration \
    --group-id "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1" \
    --certificate-expiry-in-milliseconds 864000000
```

輸出：

```
{
    "CertificateExpiryInMilliseconds": 864000000,
    "CertificateAuthorityExpiryInMilliseconds": 2524607999000,
    "GroupId": "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1"
}
```

如需詳細資訊，請參閱《AWS IoT Greengrass 開發人員指南AWS 》中的 IoT Greengrass 安全性。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateGroupCertificateConfiguration。

update-group

以下程式碼範例顯示如何使用 update-group。

AWS CLI

更新群組名稱

下列update-group範例會更新指定 Greengrass 群組的名稱。如果您想要更新群組的詳細資訊，請使用 create-group-version命令來建立新的版本。

```
aws greengrass update-group \
    --group-id "1402daf9-71cf-4cfe-8be0-d5e80526d0d8" \
    --name TestGroup4of6
```

如需詳細資訊，請參閱 IoT Greengrass 開發人員指南中的 Configure AWS IoT Greengrass on AWS IoT。 AWS IoT

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UpdateGroup。

update-logger-definition

以下程式碼範例顯示如何使用 update-logger-definition。

AWS CLI

更新記錄器定義

下列update-logger-definition範例會變更指定記錄器定義的名稱。您只能更新記錄器定義的 name 屬性。

```
aws greengrass update-logger-definition \
    --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0" \
    --name "LoggingConfigsForSensors"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的使用 IoT Greengrass 日誌進行監控。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateLoggerDefinition。

update-resource-definition

以下程式碼範例顯示如何使用 update-resource-definition。

AWS CLI

更新資源定義的名稱

下列update-resource-definition範例會更新指定資源定義的名稱。如果您想要變更資源的詳細資訊，請使用 create-resource-definition-version命令來建立新的版本。

```
aws greengrass update-resource-definition \
    --resource-definition-id "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38" \
    --name GreengrassConnectorResources
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的使用 Lambda 函數和連接器存取本機資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateResourceDefinition。

update-subscription-definition

以下程式碼範例顯示如何使用 update-subscription-definition。

AWS CLI

更新訂閱定義的名稱

下列update-subscription-definition範例會更新指定訂閱定義的名稱。如果您想要變更訂閱的詳細資訊，請使用 create-subscription-definition-version命令來建立新的版本。

```
aws greengrass update-subscription-definition \
    --subscription-definition-id "fa81bc84-3f59-4377-a84b-5d0134da359b" \
    --name "ObsoleteSubscription"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 指南中的標題。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSubscriptionDefinition。

update-thing-runtime-configuration

以下程式碼範例顯示如何使用 update-thing-runtime-configuration。

AWS CLI

在 Greengrass 核心的執行時間組態中開啟遙測

下列update-thing-runtime-configuration範例會更新 Greengrass 核心的執行時間組態，以開啟遙測。

```
aws greengrass update-thing-runtime-configuration \
    --thing-name SampleGreengrassCore \
    --telemetry-configuration {\"Telemetry\":\"On\"}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Greengrass 開發人員指南中的設定遙測設定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateThingRuntimeConfiguration。

AWS IoT Greengrass V2 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS IoT Greengrass V2。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

associate-service-role-to-account

以下程式碼範例顯示如何使用 associate-service-role-to-account。

AWS CLI

將 Greengrass 服務角色與 AWS 您的帳戶建立關聯

下列associate-service-role-to-account範例會將服務角色與 AWS 帳戶的 AWS IoT Greengrass 建立關聯。

```
aws greengrassv2 associate-service-role-to-account \
    --role-arn arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole
```

輸出：

```
{
    "associatedAt": "2022-01-19T19:21:53Z"
}
```

如需詳細資訊，請參閱 AWS IoT V2 Greengrass V2 開發人員指南中的 Greengrass 服務角色。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateServiceRoleToAccount。

batch-associate-client-device-with-core-device

以下程式碼範例顯示如何使用 batch-associate-client-device-with-core-device。

AWS CLI

將用戶端裝置與核心裝置建立關聯

下列batch-associate-client-device-with-core-device範例會將兩個用戶端裝置與核心裝置建立關聯。

```
aws greengrassv2 batch-associate-client-device-with-core-device \
  --core-device-thing-name MyGreengrassCore \
  --entries thingName=MyClientDevice1 thingName=MyClientDevice2
```

輸出：

```
{
    "errorEntries": []
}
```

如需詳細資訊，請參閱 IoT Greengrass V2 開發人員指南中的與本機 IoT 裝置互動。 AWS IoT V2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchAssociateClientDeviceWithCoreDevice。

batch-disassociate-client-device-from-core-device

以下程式碼範例顯示如何使用 batch-disassociate-client-device-from-core-device。

AWS CLI

取消用戶端裝置與核心裝置的關聯

下列batch-disassociate-client-device-from-core-device範例會取消兩個用戶端裝置與核心裝置的關聯。

```
aws greengrassv2 batch-disassociate-client-device-from-core-device \
  --core-device-thing-name MyGreengrassCore \
  --entries thingName=MyClientDevice1 thingName=MyClientDevice2
```

輸出：

```
{
    "errorEntries": []
}
```

如需詳細資訊，請參閱 IoT Greengrass V2 開發人員指南中的與本機 IoT 裝置互動。 AWS IoT V2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDisassociateClientDeviceFromCoreDevice。

cancel-deployment

以下程式碼範例顯示如何使用 cancel-deployment。

AWS CLI

若要取消部署

下列cancel-deployment範例會停止物件群組的連續部署。

```
aws greengrassv2 cancel-deployment \
```

```
    --deployment-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "message": "SUCCESS"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的取消部署。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelDeployment。

create-component-version

以下程式碼範例顯示如何使用 create-component-version。

AWS CLI

範例 1：從配方建立元件版本

下列create-component-version範例會從配方檔案建立 Hello World 元件的版本。

```
aws greengrassv2 create-component-version \
    --inline-recipe fileb://com.example.HelloWorld-1.0.0.json
```

com.example.HelloWorld-1.0.0.json 的內容：

```
{
    "RecipeFormatVersion": "2020-01-25",
    "ComponentName": "com.example.HelloWorld",
    "ComponentVersion": "1.0.0",
    "ComponentDescription": "My first AWS IoT Greengrass component.",
    "ComponentPublisher": "Amazon",
    "ComponentConfiguration": {
        "DefaultConfiguration": {
            "Message": "world"
        }
    },
    "Manifests": [
        {
            "Platform": {
```

```
                "os": "linux"
            },
            "Lifecycle": {
                "Run": "echo 'Hello {configuration:/Message}'"
            }
        }
    ]
}
```

輸出：

```
{
    "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0",
    "componentName": "com.example.HelloWorld",
    "componentVersion": "1.0.0",
    "creationTimestamp": "2021-01-07T16:24:33.650000-08:00",
    "status": {
        "componentState": "REQUESTED",
        "message": "NONE",
        "errors": {}
    }
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的建立自訂元件和上傳要部署的元件。 https://docs.aws.amazon.com/greengrass/v2/developerguide/upload-components.html

範例 2：從 AWS Lambda 函數建立元件版本

下列`create-component-version`範例會從 AWS Lambda 函數建立 Hello World 元件的版本。

```
aws greengrassv2 create-component-version \
    --cli-input-json file://lambda-function-component.json
```

`lambda-function-component.json` 的內容：

```
{
    "lambdaFunction": {
        "lambdaArn": "arn:aws:lambda:us-
west-2:123456789012:function:HelloWorldPythonLambda:1",
        "componentName": "com.example.HelloWorld",
```

```
        "componentVersion": "1.0.0",
        "componentLambdaParameters": {
            "eventSources": [
                {
                    "topic": "hello/world/+",
                    "type": "IOT_CORE"
                }
            ]
        }
    }
}
```

輸出：

```
{
    "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0",
    "componentName": "com.example.HelloWorld",
    "componentVersion": "1.0.0",
    "creationTimestamp": "2021-01-07T17:05:27.347000-08:00",
    "status": {
        "componentState": "REQUESTED",
        "message": "NONE",
        "errors": {}
    }
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的執行 AWS Lambda 函數。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateComponentVersion。

`create-deployment`

以下程式碼範例顯示如何使用 `create-deployment`。

AWS CLI

範例 1：建立部署

下列`create-deployment`範例會將 AWS IoT Greengrass 命令列界面部署至核心裝置。

```
aws greengrassv2 create-deployment \
```

```
    --cli-input-json file://cli-deployment.json
```

cli-deployment.json 的內容：

```
{
    "targetArn": "arn:aws:iot:us-west-2:123456789012:thing/MyGreengrassCore",
    "deploymentName": "Deployment for MyGreengrassCore",
    "components": {
        "aws.greengrass.Cli": {
            "componentVersion": "2.0.3"
        }
    },
    "deploymentPolicies": {
        "failureHandlingPolicy": "DO_NOTHING",
        "componentUpdatePolicy": {
            "timeoutInSeconds": 60,
            "action": "NOTIFY_COMPONENTS"
        },
        "configurationValidationPolicy": {
            "timeoutInSeconds": 60
        }
    },
    "iotJobConfiguration": {}
}
```

輸出：

```
{
    "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的建立部署。

範例 2：建立更新元件組態的部署

下列create-deployment範例會將 AWS IoT Greengrass 核元件部署至一組核心裝置。此部署會針對 nucleus 元件套用下列組態更新：

將目標裝置的代理設定重設為其預設無代理設定。將目標裝置的 MQTT 設定重設為其預設值。設定核心的 JVM 選項。設定核心的記錄層級。

```
aws greengrassv2 create-deployment \
```

```
    --cli-input-json file://nucleus-deployment.json
```

nucleus-deployment.json 的內容：

```
{
    "targetArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
MyGreengrassCoreGroup",
    "deploymentName": "Deployment for MyGreengrassCoreGroup",
    "components": {
        "aws.greengrass.Nucleus": {
            "componentVersion": "2.0.3",
            "configurationUpdate": {
                "reset": [
                    "/networkProxy",
                    "/mqtt"
                ],
                "merge": "{\"jvmOptions\":\"-Xmx64m\",\"logging\":{\"level\":\"WARN
\"}}"
            }
        }
    },
    "deploymentPolicies": {
        "failureHandlingPolicy": "ROLLBACK",
        "componentUpdatePolicy": {
            "timeoutInSeconds": 60,
            "action": "NOTIFY_COMPONENTS"
        },
        "configurationValidationPolicy": {
            "timeoutInSeconds": 60
        }
    },
    "iotJobConfiguration": {}
}
```

輸出：

```
{
    "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "iotJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "iotJobArn": "arn:aws:iot:us-west-2:123456789012:job/a1b2c3d4-5678-90ab-cdef-
EXAMPLE22222"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的建立部署和更新元件組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDeployment。

delete-component

以下程式碼範例顯示如何使用 delete-component。

AWS CLI

刪除元件版本

下列delete-component範例會刪除 Hello World 元件。

```
aws greengrassv2 delete-component \
    --arn arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的管理元件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteComponent。

delete-core-device

以下程式碼範例顯示如何使用 delete-core-device。

AWS CLI

刪除核心裝置

下列delete-core-device範例會刪除 an AWS IoT Greengrass 核心裝置。

```
aws greengrassv2 delete-core-device \
    --core-device-thing-name MyGreengrassCore
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的解除安裝 IoT Greengrass Core 軟體。 AWS IoT V2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCoreDevice。

`describe-component`

以下程式碼範例顯示如何使用 `describe-component`。

AWS CLI

描述元件版本

下列`describe-component`範例說明 Hello World 元件。

```
aws greengrassv2 describe-component \
    --arn arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0
```

輸出：

```
{
    "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0",
    "componentName": "com.example.HelloWorld",
    "componentVersion": "1.0.0",
    "creationTimestamp": "2021-01-07T17:12:11.133000-08:00",
    "publisher": "Amazon",
    "description": "My first AWS IoT Greengrass component.",
    "status": {
        "componentState": "DEPLOYABLE",
        "message": "NONE",
        "errors": {}
    },
    "platforms": [
        {
            "attributes": {
                "os": "linux"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的管理元件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeComponent。

disassociate-service-role-from-account

以下程式碼範例顯示如何使用 disassociate-service-role-from-account。

AWS CLI

取消 Greengrass 服務角色與 AWS 帳戶的關聯

下列disassociate-service-role-from-account範例會取消您 AWS 帳戶的 Greengrass 服務角色與 AWS IoT Greengrass 的關聯。

```
aws greengrassv2 disassociate-service-role-from-account
```

輸出：

```
{
    "disassociatedAt": "2022-01-19T19:26:09Z"
}
```

如需詳細資訊，請參閱 AWS IoT V2 Greengrass V2 開發人員指南中的 Greengrass 服務角色。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateServiceRoleFromAccount。

get-component-version-artifact

以下程式碼範例顯示如何使用 get-component-version-artifact。

AWS CLI

取得 URL 以下載元件成品

下列get-component-version-artifact範例會取得 URL 來下載本機偵錯主控台元件的 JAR 檔案。

```
aws greengrassv2 get-component-version-artifact \
    --arn arn:aws:greengrass:us-
west-2:aws:components:aws.greengrass.LocalDebugConsole:versions:2.0.3 \
    --artifact-name "Uvt6ZEzQ9TKiAuLbfXBX_APdY0TWks3uc46tHFHTzBM=/
aws.greengrass.LocalDebugConsole.jar"
```

輸出：

```
{
    "preSignedUrl": "https://evergreencomponentmanageme-
artifactbucket7410c9ef-g18n1iya8kwr.s3.us-west-2.amazonaws.com/public/
aws.greengrass.LocalDebugConsole/2.0.3/s3/ggv2-component-releases-prod-pdx/
EvergreenHttpDebugView/2ffc496ba41b39568968b22c582b4714a937193ee7687a45527238e696672521/
aws.greengrass.LocalDebugConsole/aws.greengrass.LocalDebugConsole.jar?X-Amz-
Security-Token=KwfLKSdEXAMPLE..."
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的管理元件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetComponentVersionArtifact。

get-component

以下程式碼範例顯示如何使用 get-component。

AWS CLI

範例 1：下載 YAML 格式的元件配方 (Linux、macOS 或 Unix)

下列get-component範例會將 Hello World 元件的配方下載到 YAML 格式的檔案。此命令會執行下列動作：

使用 --output和 --query 參數來控制命令的輸出。這些參數會從命令的輸出中擷取配方 Blob。如需控制輸出的詳細資訊，請參閱《命令列界面使用者指南》中的控制命令輸出。使用 base64公用程式。 AWS 此公用程式會將解壓縮的 Blob 解碼為原始文字。成功get-component命令傳回的 Blob 為 base64 編碼文字。您必須解碼此 Blob 以取得原始文字。將解碼的文字儲存到檔案。命令 (> com.example.HelloWorld-1.0.0.json) 的最後一個區段會將解碼的文字儲存到 檔案。

```
aws greengrassv2 get-component \
    --arn arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0 \
    --recipe-output-format YAML \
    --query recipe \
    --output text | base64 --decode > com.example.HelloWorld-1.0.0.json
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的管理元件。

範例 2：下載 YAML 格式的元件配方 (Windows CMD)

下列get-component範例會將 Hello World 元件的配方下載到 YAML 格式的檔案。此命令使用certutil公用程式。

```
aws greengrassv2 get-component ^
    --arn arn:aws:greengrass:us-
west-2:675946970638:components:com.example.HelloWorld:versions:1.0.0 ^
    --recipe-output-format YAML ^
    --query recipe ^
    --output text > com.example.HelloWorld-1.0.0.yaml.b64

certutil -
decode com.example.HelloWorld-1.0.0.yaml.b64 com.example.HelloWorld-1.0.0.yaml
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的管理元件。

範例 3：下載 YAML 格式的元件配方 (Windows PowerShell)

下列get-component範例會將 Hello World 元件的配方下載到 YAML 格式的檔案。此命令使用certutil公用程式。

```
aws greengrassv2 get-component `
    --arn arn:aws:greengrass:us-
west-2:675946970638:components:com.example.HelloWorld:versions:1.0.0 `
    --recipe-output-format YAML `
    --query recipe `
    --output text > com.example.HelloWorld-1.0.0.yaml.b64

certutil -
decode com.example.HelloWorld-1.0.0.yaml.b64 com.example.HelloWorld-1.0.0.yaml
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的管理元件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetComponent。

get-connectivity-info

以下程式碼範例顯示如何使用 get-connectivity-info。

AWS CLI

取得 Greengrass 核心裝置的連線資訊

下列get-connectivity-info範例會取得 Greengrass 核心裝置的連線資訊。用戶端裝置會使用此資訊來連線至在此核心裝置上執行的 MQTT 代理程式。

```
aws greengrassv2 get-connectivity-info \
    --thing-name MyGreengrassCore
```

輸出：

```
{
    "connectivityInfo": [
        {
            "id": "localIP_192.0.2.0",
            "hostAddress": "192.0.2.0",
            "portNumber": 8883
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的管理核心裝置端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetConnectivityInfo。

get-core-device

以下程式碼範例顯示如何使用 get-core-device。

AWS CLI

取得核心裝置

下列get-core-device範例取得有關 an AWS IoT Greengrass 核心裝置的資訊。

```
aws greengrassv2 get-core-device \
    --core-device-thing-name MyGreengrassCore
```

輸出：

```
{
    "coreDeviceThingName": "MyGreengrassCore",
    "coreVersion": "2.0.3",
    "platform": "linux",
```

```
    "architecture": "amd64",
    "status": "HEALTHY",
    "lastStatusUpdateTimestamp": "2021-01-08T04:57:58.838000-08:00",
    "tags": {}
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的檢查核心裝置狀態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCoreDevice。

get-deployment

以下程式碼範例顯示如何使用 get-deployment。

AWS CLI

取得部署

下列get-deployment範例取得 AWS IoT Greengrass 核元件部署至一組核心裝置的相關資訊。

```
aws greengrassv2 get-deployment \
    --deployment-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "targetArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
MyGreengrassCoreGroup",
    "revisionId": "14",
    "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "deploymentName": "Deployment for MyGreengrassCoreGroup",
    "deploymentStatus": "ACTIVE",
    "iotJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "iotJobArn": "arn:aws:iot:us-west-2:123456789012:job/a1b2c3d4-5678-90ab-cdef-
EXAMPLE22222",
    "components": {
        "aws.greengrass.Nucleus": {
            "componentVersion": "2.0.3",
            "configurationUpdate": {
                "merge": "{\"jvmOptions\":\"-Xmx64m\",\"logging\":{\"level\":\"WARN
\"}}",
                "reset": [
                    "/networkProxy",
```

```
                    "/mqtt"
                ]
            }
        }
    },
    "deploymentPolicies": {
        "failureHandlingPolicy": "ROLLBACK",
        "componentUpdatePolicy": {
            "timeoutInSeconds": 60,
            "action": "NOTIFY_COMPONENTS"
        },
        "configurationValidationPolicy": {
            "timeoutInSeconds": 60
        }
    },
    "iotJobConfiguration": {},
    "creationTimestamp": "2021-01-07T17:21:20.691000-08:00",
    "isLatestForTarget": false,
    "tags": {}
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的將元件部署至裝置。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDeployment。

get-service-role-for-account

以下程式碼範例顯示如何使用 get-service-role-for-account。

AWS CLI

取得您 AWS 帳戶的 Greengrass 服務角色

下列get-service-role-for-account範例會取得與您 AWS 帳戶 AWS IoT Greengrass 相關聯的服務角色。

```
aws greengrassv2 get-service-role-for-account
```

輸出：

```
{
    "associatedAt": "2022-01-19T19:21:53Z",
```

```
    "roleArn": "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
}
```

如需詳細資訊，請參閱 AWS IoT V2 Greengrass V2 開發人員指南中的 Greengrass 服務角色。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetServiceRoleForAccount。

`list-client-devices-associated-with-core-device`

以下程式碼範例顯示如何使用 `list-client-devices-associated-with-core-device`。

AWS CLI

列出與核心裝置相關聯的用戶端裝置

下列`list-client-devices-associated-with-core-device`範例列出與核心裝置相關聯的所有用戶端裝置。

```
aws greengrassv2 list-client-devices-associated-with-core-device \
  --core-device-thing-name MyTestGreengrassCore
```

輸出：

```
{
    "associatedClientDevices": [
        {
            "thingName": "MyClientDevice2",
            "associationTimestamp": "2021-07-12T16:33:55.843000-07:00"
        },
        {
            "thingName": "MyClientDevice1",
            "associationTimestamp": "2021-07-12T16:33:55.843000-07:00"
        }
    ]
}
```

如需詳細資訊，請參閱 IoT Greengrass V2 開發人員指南中的與本機 IoT 裝置互動。 AWS IoT V2

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListClientDevicesAssociatedWithCoreDevice。

list-component-versions

以下程式碼範例顯示如何使用 list-component-versions。

AWS CLI

列出元件的版本

下列list-component-versions範例列出 Hello World 元件的所有版本。

```
aws greengrassv2 list-component-versions \
    --arn arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld
```

輸出：

```
{
    "componentVersions": [
        {
            "componentName": "com.example.HelloWorld",
            "componentVersion": "1.0.1",
            "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.1"
        },
        {
            "componentName": "com.example.HelloWorld",
            "componentVersion": "1.0.0",
            "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的管理元件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListComponentVersions。

list-components

以下程式碼範例顯示如何使用 list-components。

AWS CLI

列出元件

下列list-components範例列出您 AWS 帳戶中在目前區域中定義的每個元件及其最新版本。

```
aws greengrassv2 list-components
```

輸出：

```
{
    "components": [
        {
            "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld",
            "componentName": "com.example.HelloWorld",
            "latestVersion": {
                "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.1",
                "componentVersion": "1.0.1",
                "creationTimestamp": "2021-01-08T16:51:07.352000-08:00",
                "description": "My first AWS IoT Greengrass component.",
                "publisher": "Amazon",
                "platforms": [
                    {
                        "attributes": {
                            "os": "linux"
                        }
                    }
                ]
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的管理元件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListComponents。

list-core-devices

以下程式碼範例顯示如何使用 list-core-devices。

AWS CLI

列出核心裝置

下列`list-core-devices`範例列出目前區域中 AWS 您帳戶中的 AWS IoT Greengrass 核心裝置。

```
aws greengrassv2 list-core-devices
```

輸出：

```
{
    "coreDevices": [
        {
            "coreDeviceThingName": "MyGreengrassCore",
            "status": "HEALTHY",
            "lastStatusUpdateTimestamp": "2021-01-08T04:57:58.838000-08:00"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的檢查核心裝置狀態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCoreDevices。

`list-deployments`

以下程式碼範例顯示如何使用 `list-deployments`。

AWS CLI

列出部署

下列`list-deployments`範例列出您 AWS 帳戶中在目前區域中定義之每個部署的最新修訂版。

```
aws greengrassv2 list-deployments
```

輸出：

```
{
    "deployments": [
```

```
        {
            "targetArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
MyGreengrassCoreGroup",
            "revisionId": "14",
            "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "deploymentName": "Deployment for MyGreengrassCoreGroup",
            "creationTimestamp": "2021-01-07T17:21:20.691000-08:00",
            "deploymentStatus": "ACTIVE",
            "isLatestForTarget": false
        },
        {
            "targetArn": "arn:aws:iot:us-west-2:123456789012:thing/
MyGreengrassCore",
            "revisionId": "1",
            "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "deploymentName": "Deployment for MyGreengrassCore",
            "creationTimestamp": "2021-01-06T16:10:42.407000-08:00",
            "deploymentStatus": "COMPLETED",
            "isLatestForTarget": false
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的將元件部署至裝置。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDeployments。

`list-effective-deployments`

以下程式碼範例顯示如何使用 `list-effective-deployments`。

AWS CLI

列出部署任務

下列`list-effective-deployments`範例列出適用於 an AWS IoT Greengrass 核心裝置的部署。

```
aws greengrassv2 list-effective-deployments \
    --core-device-thing-name MyGreengrassCore
```

輸出：

```
{
    "effectiveDeployments": [
        {
            "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "deploymentName": "Deployment for MyGreengrassCore",
            "iotJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "targetArn": "arn:aws:iot:us-west-2:123456789012:thing/
MyGreengrassCore",
            "coreDeviceExecutionStatus": "COMPLETED",
            "reason": "SUCCESSFUL",
            "creationTimestamp": "2021-01-06T16:10:42.442000-08:00",
            "modifiedTimestamp": "2021-01-08T17:21:27.830000-08:00"
        },
        {
            "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "deploymentName": "Deployment for MyGreengrassCoreGroup",
            "iotJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
            "iotJobArn": "arn:aws:iot:us-west-2:123456789012:job/a1b2c3d4-5678-90ab-
cdef-EXAMPLE44444",
            "targetArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
MyGreengrassCoreGroup",
            "coreDeviceExecutionStatus": "SUCCEEDED",
            "reason": "SUCCESSFUL",
            "creationTimestamp": "2021-01-07T17:19:20.394000-08:00",
            "modifiedTimestamp": "2021-01-07T17:21:20.721000-08:00"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的檢查核心裝置狀態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEffectiveDeployments。

`list-installed-components`

以下程式碼範例顯示如何使用 `list-installed-components`。

AWS CLI

列出安裝在核心裝置上的元件

下列`list-installed-components`範例列出安裝在 an AWS IoT Greengrass 核心裝置上的元件。

```
aws greengrassv2 list-installed-components \
    --core-device-thing-name MyGreengrassCore
```

輸出：

```
{
    "installedComponents": [
        {
            "componentName": "aws.greengrass.Cli",
            "componentVersion": "2.0.3",
            "lifecycleState": "RUNNING",
            "isRoot": true
        },
        {
            "componentName": "aws.greengrass.Nucleus",
            "componentVersion": "2.0.3",
            "lifecycleState": "FINISHED",
            "isRoot": true
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的檢查核心裝置狀態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListInstalledComponents。

list-tags-for-resource

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出資源的標籤

下列`list-tags-for-resource`範例列出 an AWS IoT Greengrass 核心裝置的所有標籤。

```
aws greengrassv2 list-tags-for-resource \
    --resource-arn arn:aws:greengrass:us-
west-2:123456789012:coreDevices:MyGreengrassCore
```

輸出：

```
{
    "tags": {
        "Owner": "richard-roe"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的標記您的 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`tag-resource`

以下程式碼範例顯示如何使用 `tag-resource`。

AWS CLI

將標籤加入資源

下列`tag-resource`範例會將擁有者標籤新增至 an AWS IoT Greengrass 核心裝置。您可以使用此標籤，根據擁有它的人員來控制對核心裝置的存取。

```
aws greengrassv2 tag-resource \
    --resource-arn arn:aws:greengrass:us-
west-2:123456789012:coreDevices:MyGreengrassCore \
    --tags Owner=richard-roe
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的標記您的 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

`untag-resource`

以下程式碼範例顯示如何使用 `untag-resource`。

AWS CLI

從資源移除標籤

下列`untag-resource`範例會從 AWS IoT Greengrass 核心裝置移除擁有者標籤。

```
aws iotsitewise untag-resource \
    --resource-arn arn:aws:greengrass:us-
west-2:123456789012:coreDevices:MyGreengrassCore \
    --tag-keys Owner
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的標記您的 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-connectivity-info

以下程式碼範例顯示如何使用 update-connectivity-info。

AWS CLI

更新 Greengrass 核心裝置的連線資訊

下列update-connectivity-info範例會取得 Greengrass 核心裝置的連線資訊。用戶端裝置會使用此資訊來連線至在此核心裝置上執行的 MQTT 代理程式。

```
aws greengrassv2 update-connectivity-info \
    --thing-name MyGreengrassCore \
    --cli-input-json file://core-device-connectivity-info.json
```

core-device-connectivity-info.json 的內容：

```
{
    "connectivityInfo": [
        {
            "hostAddress": "192.0.2.0",
            "portNumber": 8883,
            "id": "localIP_192.0.2.0"
        }
    ]
}
```

輸出：

```
{
```

```
    "version": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

如需詳細資訊，請參閱 AWS IoT Greengrass V2 開發人員指南中的管理核心裝置端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateConnectivityInfo。

AWS IoT Jobs SDK release 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS IoT Jobs SDK release。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`describe-job-execution`

以下程式碼範例顯示如何使用 `describe-job-execution`。

AWS CLI

取得任務執行的詳細資訊

下列`describe-job-execution`範例會擷取指定任務和物件的最新執行詳細資訊。

```
aws iot-jobs-data describe-job-execution \
    --job-id SampleJob \
    --thing-name MotionSensor1 \
    --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

輸出：

```
{
```

```
    "execution": {
        "approximateSecondsBeforeTimedOut": 88,
        "executionNumber": 2939653338,
        "jobId": "SampleJob",
        "lastUpdatedAt": 1567701875.743,
        "queuedAt": 1567701902.444,
        "status": "QUEUED",
        "thingName": "MotionSensor1 ",
        "versionNumber": 3
   }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的裝置和任務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeJobExecution。

get-pending-job-executions

以下程式碼範例顯示如何使用 get-pending-job-executions。

AWS CLI

取得物件未處於終端機狀態的所有任務清單

下列get-pending-job-executions範例會顯示指定物件未處於結束狀態的所有任務清單。

```
aws iot-jobs-data get-pending-job-executions \
    --thing-name MotionSensor1
    --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

輸出：

```
{
    "inProgressJobs": [
    ],
    "queuedJobs": [
        {
            "executionNumber": 2939653338,
            "jobId": "SampleJob",
            "lastUpdatedAt": 1567701875.743,
            "queuedAt": 1567701902.444,
            "versionNumber": 3
```

```
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的裝置和任務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPendingJobExecutions。

`start-next-pending-job-execution`

以下程式碼範例顯示如何使用 `start-next-pending-job-execution`。

AWS CLI

取得並開始物件的下一個待定任務執行

下列`start-next-pending-job-execution`範例會擷取並啟動下一個任務執行，其狀態為指定物件的 IN_PROGRESS 或 QUEUED。

```
aws iot-jobs-data start-next-pending-job-execution \
    --thing-name MotionSensor1
    --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

輸出：

```
{
    "execution": {
        "approximateSecondsBeforeTimedOut": 88,
        "executionNumber": 2939653338,
        "jobId": "SampleJob",
        "lastUpdatedAt": 1567714853.743,
        "queuedAt": 1567701902.444,
        "startedAt": 1567714871.690,
        "status": "IN_PROGRESS",
        "thingName": "MotionSensor1 ",
        "versionNumber": 3
   }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的裝置和任務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartNextPendingJobExecution。

update-job-execution

以下程式碼範例顯示如何使用 update-job-execution。

AWS CLI

更新任務執行的狀態

下列update-job-execution範例會更新指定任務和物件的狀態。

```
aws iot-jobs-data update-job-execution \
    --job-id SampleJob \
    --thing-name MotionSensor1 \
    --status REMOVED \
    --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

輸出：

```
{
    "executionState": {
        "status": "REMOVED",
        "versionNumber": 3
    },
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的裝置和任務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateJobExecution。

AWS IoT SiteWise 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS IoT SiteWise。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-assets`

以下程式碼範例顯示如何使用 `associate-assets`。

AWS CLI

將子資產與父資產建立關聯

下列`associate-assets`範例會將風力發電機資產與風力發電廠資產建立關聯，其中風力發電機資產模型以階層形式存在於風力發電廠資產模型中。

```
aws iotsitewise associate-assets \
    --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \
    --hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE \
    --child-asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的關聯資產。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateAssets。

`batch-associate-project-assets`

以下程式碼範例顯示如何使用 `batch-associate-project-assets`。

AWS CLI

將資產與專案建立關聯

下列`batch-associate-project-assets`範例會將風力發電廠資產與專案建立關聯。

```
aws iotsitewise batch-associate-project-assets \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE \
    --asset-ids a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT SiteWise Monitor 應用程式指南中的將資產新增至專案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchAssociateProjectAssets。

batch-disassociate-project-assets

以下程式碼範例顯示如何使用 batch-disassociate-project-assets。

AWS CLI

取消資產與專案的關聯

下列batch-disassociate-project-assets範例會取消風力發電廠資產與專案的關聯。

```
aws iotsitewise batch-disassociate-project-assets \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE \
    --asset-ids a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT SiteWise Monitor 應用程式指南中的將資產新增至專案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDisassociateProjectAssets。

batch-put-asset-property-value

以下程式碼範例顯示如何使用 batch-put-asset-property-value。

AWS CLI

將資料傳送至資產屬性

下列batch-put-asset-property-value範例會將電源和溫度資料傳送至屬性別名識別的資產屬性。

```
aws iotsitewise batch-put-asset-property-value \
    --cli-input-json file://batch-put-asset-property-value.json
```

batch-put-asset-property-value.json 的內容：

```
{
    "entries": [
        {
            "entryId": "1575691200-company-windfarm-3-turbine-7-power",
            "propertyAlias": "company-windfarm-3-turbine-7-power",
```

```
            "propertyValues": [
                {
                    "value": {
                        "doubleValue": 4.92
                    },
                    "timestamp": {
                        "timeInSeconds": 1575691200
                    },
                    "quality": "GOOD"
                }
            ]
        },
        {
            "entryId": "1575691200-company-windfarm-3-turbine-7-temperature",
            "propertyAlias": "company-windfarm-3-turbine-7-temperature",
            "propertyValues": [
                {
                    "value": {
                        "integerValue": 38
                    },
                    "timestamp": {
                        "timeInSeconds": 1575691200
                    }
                }
            ]
        }
    ]
}
```

輸出：

```
{
    "errorEntries": []
}
```

如需詳細資訊，請參閱 IoT SiteWise 使用者指南中的使用 AWS IoT SiteWise API 擷取資料。AWS IoT SiteWise

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchPutAssetPropertyValue。

create-access-policy

以下程式碼範例顯示如何使用 create-access-policy。

AWS CLI

範例 1：授予使用者入口網站的管理存取權

下列create-access-policy範例會建立存取政策，授予使用者對風力發電廠公司 Web 入口網站的管理存取權。

```
aws iotsitewise create-access-policy \
    --cli-input-json file://create-portal-administrator-access-policy.json
```

create-portal-administrator-access-policy.json 的內容：

```
{
    "accessPolicyIdentity": {
        "user": {
            "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
        }
    },
    "accessPolicyPermission": "ADMINISTRATOR",
    "accessPolicyResource": {
        "portal": {
            "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE"
        }
    }
}
```

輸出：

```
{
    "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE",
    "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/
a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE"
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的新增或移除入口網站管理員。

範例 2：授予使用者對專案的唯讀存取權

下列create-access-policy範例會建立存取政策，授予使用者對風力發電廠專案的唯讀存取權。

```
aws iotsitewise create-access-policy \
```

```
    --cli-input-json file://create-project-viewer-access-policy.json
```

create-project-viewer-access-policy.json 的內容：

```
{
    "accessPolicyIdentity": {
        "user": {
            "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
        }
    },
    "accessPolicyPermission": "VIEWER",
    "accessPolicyResource": {
        "project": {
            "id": "a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE"
        }
    }
}
```

輸出：

```
{
    "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE",
    "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/
a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE"
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise Monitor 應用程式指南中的指派專案檢視器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAccessPolicy。

create-asset-model

以下程式碼範例顯示如何使用 create-asset-model。

AWS CLI

建立資產模型

下列create-asset-model範例會建立資產模型，以定義具有下列屬性的風力渦輪機：

序號 - 風turbineGenerated功率 - 從風turbineTemperature C - 從風力發電機以 CelsiusTemperature F 產生的溫度資料串流 - 從攝氏到華氏的映射溫度資料點

```
aws iotsitewise create-asset-model \
    --cli-input-json file://create-wind-turbine-model.json
```

create-wind-turbine-model.json 的內容：

```
{
    "assetModelName": "Wind Turbine Model",
    "assetModelDescription": "Represents a wind turbine",
    "assetModelProperties": [
        {
            "name": "Serial Number",
            "dataType": "STRING",
            "type": {
                "attribute": {}
            }
        },
        {
            "name": "Generated Power",
            "dataType": "DOUBLE",
            "unit": "kW",
            "type": {
                "measurement": {}
            }
        },
        {
            "name": "Temperature C",
            "dataType": "DOUBLE",
            "unit": "Celsius",
            "type": {
                "measurement": {}
            }
        },
        {
            "name": "Temperature F",
            "dataType": "DOUBLE",
            "unit": "Fahrenheit",
            "type": {
                "transform": {
                    "expression": "temp_c * 9 / 5 + 32",
                    "variables": [
                        {
                            "name": "temp_c",
                            "value": {
```

```
                                "propertyId": "Temperature C"
                            }
                        }
                    ]
                }
            }
        },
        {
            "name": "Total Generated Power",
            "dataType": "DOUBLE",
            "unit": "kW",
            "type": {
                "metric": {
                    "expression": "sum(power)",
                    "variables": [
                        {
                            "name": "power",
                            "value": {
                                "propertyId": "Generated Power"
                            }
                        }
                    ],
                    "window": {
                        "tumbling": {
                            "interval": "1h"
                        }
                    }
                }
            }
        }
    ]
}
```

輸出：

```
{
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "assetModelStatus": {
        "state": "CREATING"
    }
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的定義資產模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAssetModel。

create-asset

以下程式碼範例顯示如何使用 create-asset。

AWS CLI

建立資產

下列create-asset範例會從風力渦輪機資產模型建立風力渦輪機資產。

```
aws iotsitewise create-asset \
    --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --asset-name "Wind Turbine 1"
```

輸出：

```
{
    "assetId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
    "assetArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
    "assetStatus": {
        "state": "CREATING"
    }
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的建立資產。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAsset。

create-dashboard

以下程式碼範例顯示如何使用 create-dashboard。

AWS CLI

建立儀表板

下列create-dashboard範例會建立具有折線圖的儀表板，顯示風力發電廠的總產生功率。

```
aws iotsitewise create-dashboard \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE \
    --dashboard-name "Wind Farm" \
    --dashboard-definition file://create-wind-farm-dashboard.json
```

create-wind-farm-dashboard.json 的內容：

```
{
    "widgets": [
        {
            "type": "monitor-line-chart",
            "title": "Generated Power",
            "x": 0,
            "y": 0,
            "height": 3,
            "width": 3,
            "metrics": [
                {
                    "label": "Power",
                    "type": "iotsitewise",
                    "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
                    "propertyId": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE"
                }
            ]
        }
    ]
}
```

輸出：

```
{
    "dashboardId": "a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",
    "dashboardArn": "arn:aws:iotsitewise:us-west-2:123456789012:dashboard/
a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE"
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的建立儀表板 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDashboard。

create-gateway

以下程式碼範例顯示如何使用 create-gateway。

AWS CLI

建立閘道

下列create-gateway範例會建立在 AWS IoT Greengrass 上執行的閘道。

```
aws iotsitewise create-gateway \
    --gateway-name ExampleCorpGateway \
    --gateway-platform greengrass={groupArn=arn:aws:greengrass:us-
west-2:123456789012:/greengrass/groups/a1b2c3d4-5678-90ab-cdef-1b1b1EXAMPLE}
```

輸出：

```
{
    "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
    "gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的設定閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateGateway。

create-portal

以下程式碼範例顯示如何使用 create-portal。

AWS CLI

建立入口網站

下列create-portal範例會為風力發電廠公司建立 Web 入口網站。您只能在啟用 AWS 單一登入的相同區域中建立入口網站。

```
aws iotsitewise create-portal \
    --portal-name WindFarmPortal \
    --portal-description "A portal that contains wind farm projects for Example
 Corp." \
    --portal-contact-email support@example.com \
```

```
    --role-arn arn:aws:iam::123456789012:role/service-role/
MySiteWiseMonitorServiceRole
```

輸出：

```
{
    "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
    "portalArn": "arn:aws:iotsitewise:us-west-2:123456789012:portal/
a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
    "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-
aaaaaEXAMPLE.app.iotsitewise.aws",
    "portalStatus": {
        "state": "CREATING"
    },
    "ssoApplicationId": "ins-a1b2c3d4-EXAMPLE"
}
```

如需詳細資訊，請參閱 IoT SiteWise 使用者指南中的 AWS IoT SiteWise Monitor 入門和 AWS IoT SiteWise 使用者指南中的啟用 AWS SSO。 AWS IoT SiteWise

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePortal。

create-project

以下程式碼範例顯示如何使用 create-project。

AWS CLI

建立專案

下列create-project範例會建立風力發電廠專案。

```
aws iotsitewise create-project \
    --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE \
    --project-name "Wind Farm 1" \
    --project-description "Contains asset visualizations for Wind Farm #1 for
 Example Corp."
```

輸出：

```
{
    "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE",
```

```
    "projectArn": "arn:aws:iotsitewise:us-west-2:123456789012:project/
a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE"
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise Monitor 應用程式指南中的建立專案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateProject。

delete-access-policy

以下程式碼範例顯示如何使用 delete-access-policy。

AWS CLI

撤銷使用者對專案或入口網站的存取權

下列delete-access-policy範例會刪除授予使用者入口網站管理存取權的存取政策。

```
aws iotsitewise delete-access-policy \
    --access-policy-id a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的新增或移除入口網站管理員。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAccessPolicy。

delete-asset-model

以下程式碼範例顯示如何使用 delete-asset-model。

AWS CLI

刪除資產模型

下列delete-asset-model範例會刪除風力渦輪機資產模型。

```
aws iotsitewise delete-asset-model \
    --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

輸出：

```
{
```

```
    "assetModelStatus": {
        "state": "DELETING"
    }
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的刪除資產模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAssetModel。

delete-asset

以下程式碼範例顯示如何使用 delete-asset。

AWS CLI

刪除資產

下列delete-asset範例會刪除風力發電機資產。

```
aws iotsitewise delete-asset \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

輸出：

```
{
    "assetStatus": {
        "state": "DELETING"
    }
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的刪除資產。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAsset。

delete-dashboard

以下程式碼範例顯示如何使用 delete-dashboard。

AWS CLI

刪除儀表板

下列delete-dashboard範例會刪除風力發電機儀表板。

```
aws iotsitewise delete-dashboard \
    --dashboard-id a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT SiteWise Monitor 應用程式指南中的刪除儀表板。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDashboard。

delete-gateway

以下程式碼範例顯示如何使用 delete-gateway。

AWS CLI

刪除閘道

下列delete-gateway範例會刪除閘道。

```
aws iotsitewise delete-gateway \
    --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的使用閘道擷取資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteGateway。

delete-portal

以下程式碼範例顯示如何使用 delete-portal。

AWS CLI

刪除入口網站

下列delete-portal範例會刪除風力發電廠公司的 Web 入口網站。

```
aws iotsitewise delete-portal \
    --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

輸出：

```
{
    "portalStatus": {
        "state": "DELETING"
    }
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的刪除入口網站。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePortal。

delete-project

以下程式碼範例顯示如何使用 delete-project。

AWS CLI

刪除專案

下列delete-project範例會刪除風力發電廠專案。

```
aws iotsitewise delete-project \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT SiteWise Monitor 應用程式指南中的刪除專案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteProject。

describe-access-policy

以下程式碼範例顯示如何使用 describe-access-policy。

AWS CLI

描述存取政策

下列describe-access-policy範例說明 存取政策，授予使用者對風力發電廠公司 Web 入口網站的管理存取權。

```
aws iotsitewise describe-access-policy \
    --access-policy-id a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE
```

輸出：

```
{
    "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE",
    "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/
a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE",
    "accessPolicyIdentity": {
        "user": {
            "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
        }
    },
    "accessPolicyResource": {
        "portal": {
            "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE"
        }
    },
    "accessPolicyPermission": "ADMINISTRATOR",
    "accessPolicyCreationDate": "2020-02-20T22:35:15.552880124Z",
    "accessPolicyLastUpdateDate": "2020-02-20T22:35:15.552880124Z"
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的新增或移除入口網站管理員。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAccessPolicy。

describe-asset-model

以下程式碼範例顯示如何使用 describe-asset-model。

AWS CLI

描述資產模型

下列describe-asset-model範例說明風力發電廠資產模型。

```
aws iotsitewise describe-asset-model \
    --asset-model-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

輸出：

```
{
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
```

```
    "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/
a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetModelName": "Wind Farm Model",
    "assetModelDescription": "Represents a wind farm that comprises many wind
 turbines",
    "assetModelProperties": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
            "name": "Total Generated Power",
            "dataType": "DOUBLE",
            "unit": "kW",
            "type": {
                "metric": {
                    "expression": "sum(power)",
                    "variables": [
                        {
                            "name": "power",
                            "value": {
                                "propertyId": "a1b2c3d4-5678-90ab-
cdef-66666EXAMPLE",
                                "hierarchyId": "a1b2c3d4-5678-90ab-
cdef-77777EXAMPLE"
                            }
                        }
                    ],
                    "window": {
                        "tumbling": {
                            "interval": "1h"
                        }
                    }
                }
            }
        },
        {
            "id": "a1b2c3d4-5678-90ab-cdef-88888EXAMPLE",
            "name": "Region",
            "dataType": "STRING",
            "type": {
                "attribute": {
                    "defaultValue": " "
                }
            }
        }
    ],
```

```
    "assetModelHierarchies": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
            "name": "Wind Turbines",
            "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
        }
    ],
    "assetModelCreationDate": 1575671284.0,
    "assetModelLastUpdateDate": 1575671988.0,
    "assetModelStatus": {
        "state": "ACTIVE"
    }
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的描述特定資產模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAssetModel。

describe-asset-property

以下程式碼範例顯示如何使用 describe-asset-property。

AWS CLI

描述資產屬性

下列describe-asset-property範例說明風力發電廠資產的總產生功率屬性。

```
aws iotsitewise describe-asset-property \
    --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \
    --property-id a1b2c3d4-5678-90ab-cdef-99999EXAMPLE
```

輸出：

```
{
    "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
    "assetName": "Wind Farm 1",
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetProperty": {
        "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
        "name": "Total Generated Power",
        "notification": {
```

```
            "topic": "$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-
cdef-22222EXAMPLE/assets/a1b2c3d4-5678-90ab-cdef-44444EXAMPLE/properties/
a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
            "state": "DISABLED"
        },
        "dataType": "DOUBLE",
        "unit": "kW",
        "type": {
            "metric": {
                "expression": "sum(power)",
                "variables": [
                    {
                        "name": "power",
                        "value": {
                            "propertyId": "a1b2c3d4-5678-90ab-cdef-66666EXAMPLE",
                            "hierarchyId": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE"
                        }
                    }
                ],
                "window": {
                    "tumbling": {
                        "interval": "1h"
                    }
                }
            }
        }
    }
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的描述特定資產屬性。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAssetProperty。

describe-asset

以下程式碼範例顯示如何使用 describe-asset。

AWS CLI

描述資產

下列describe-asset範例說明風力發電廠資產。

```
aws iotsitewise describe-asset \
```

```
    --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

輸出：

```
{
    "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
    "assetArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
    "assetName": "Wind Farm 1",
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetProperties": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-88888EXAMPLE",
            "name": "Region",
            "dataType": "STRING"
        },
        {
            "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
            "name": "Total Generated Power",
            "dataType": "DOUBLE",
            "unit": "kW"
        }
    ],
    "assetHierarchies": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
            "name": "Wind Turbines"
        }
    ],
    "assetCreationDate": 1575672453.0,
    "assetLastUpdateDate": 1575672453.0,
    "assetStatus": {
        "state": "ACTIVE"
    }
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的描述特定資產。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAsset。

describe-dashboard

以下程式碼範例顯示如何使用 describe-dashboard。

AWS CLI

描述儀表板

下列describe-dashboard範例說明指定的風力發電廠儀表板。

```
aws iotsitewise describe-dashboard \
    --dashboard-id a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE
```

輸出：

```
{
    "dashboardId": "a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",
    "dashboardArn": "arn:aws:iotsitewise:us-west-2:123456789012:dashboard/
a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",
    "dashboardName": "Wind Farm",
    "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE",
    "dashboardDefinition": "{\"widgets\":[{\"type\":\"monitor-line-chart\",\"title
\":\"Generated Power\",\"x\":0,\"y\":0,\"height\":3,\"width\":3,\"metrics\":
[{\"label\":\"Power\",\"type\":\"iotsitewise\",\"assetId\":\"a1b2c3d4-5678-90ab-
cdef-44444EXAMPLE\",\"propertyId\":\"a1b2c3d4-5678-90ab-cdef-99999EXAMPLE\"}]}]}",
    "dashboardCreationDate": "2020-05-01T20:32:12.228476348Z",
    "dashboardLastUpdateDate": "2020-05-01T20:32:12.228476348Z"
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise Monitor 應用程式指南中的檢視儀表板。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDashboard。

describe-gateway-capability-configuration

以下程式碼範例顯示如何使用 describe-gateway-capability-configuration。

AWS CLI

描述閘道功能

下列describe-gateway-capability-configuration範例說明 OPC-UA 來源功能。

```
aws iotsitewise describe-gateway-capability-configuration \
    --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
    --capability-namespace "iotsitewise:opcuacollector:1"
```

輸出：

```
{
    "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
    "capabilityNamespace": "iotsitewise:opcuacollector:1",
    "capabilityConfiguration": "{\"sources\":[{\"name\":\"Wind Farm #1\",
\"endpoint\":{\"certificateTrust\":{\"type\":\"TrustAny\"},\"endpointUri
\":\"opc.tcp://203.0.113.0:49320\",\"securityPolicy\":\"BASIC256\",
\"messageSecurityMode\":\"SIGN_AND_ENCRYPT\",\"identityProvider\":
{\"type\":\"Username\",\"usernameSecretArn\":\"arn:aws:secretsmanager:us-
east-1:123456789012:secret:greengrass-factory1-auth-3QNDmM\"},\"nodeFilterRules\":
[]},\"measurementDataStreamPrefix\":\"\"}]}",
    "capabilitySyncStatus": "IN_SYNC"
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的設定資料來源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeGatewayCapabilityConfiguration。

describe-gateway

以下程式碼範例顯示如何使用 describe-gateway。

AWS CLI

描述閘道

下列describe-gateway範例說明閘道。

```
aws iotsitewise describe-gateway \
    --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

輸出：

```
{
    "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
    "gatewayName": "ExampleCorpGateway",
    "gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
    "gatewayPlatform": {
        "greengrass": {
```

```
            "groupArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/a1b2c3d4-5678-90ab-cdef-1b1b1EXAMPLE"
        }
    },
    "gatewayCapabilitySummaries": [
        {
            "capabilityNamespace": "iotsitewise:opcuacollector:1",
            "capabilitySyncStatus": "IN_SYNC"
        }
    ],
    "creationDate": 1588369971.457,
    "lastUpdateDate": 1588369971.457
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的使用閘道擷取資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeGateway。

describe-logging-options

以下程式碼範例顯示如何使用 `describe-logging-options`。

AWS CLI

擷取目前的 AWS IoT SiteWise 記錄選項

下列`describe-logging-options`範例會擷取目前區域中 AWS 您帳戶的目前 AWS IoT SiteWise 記錄選項。

```
aws iotsitewise describe-logging-options
```

輸出：

```
{
    "loggingOptions": {
        "level": "INFO"
    }
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的使用 Amazon CloudWatch Logs 監控IoT SiteWise。 AWS IoT SiteWise

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLoggingOptions。

describe-portal

以下程式碼範例顯示如何使用 describe-portal。

AWS CLI

描述入口網站

下列describe-portal範例說明風力發電廠公司的 Web 入口網站。

```
aws iotsitewise describe-portal \
    --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

輸出：

```
{
    "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
    "portalArn": "arn:aws:iotsitewise:us-west-2:123456789012:portal/
a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
    "portalName": "WindFarmPortal",
    "portalDescription": "A portal that contains wind farm projects for Example
 Corp.",
    "portalClientId": "E-a1b2c3d4e5f6_a1b2c3d4e5f6EXAMPLE",
    "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-
aaaaaEXAMPLE.app.iotsitewise.aws",
    "portalContactEmail": "support@example.com",
    "portalStatus": {
        "state": "ACTIVE"
    },
    "portalCreationDate": "2020-02-04T23:01:52.90248068Z",
    "portalLastUpdateDate": "2020-02-04T23:01:52.90248078Z",
    "roleArn": "arn:aws:iam::123456789012:role/MySiteWiseMonitorServiceRole"
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的管理入口網站。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePortal。

describe-project

以下程式碼範例顯示如何使用 describe-project。

AWS CLI

描述專案

下列describe-project範例說明風力發電廠專案。

```
aws iotsitewise describe-project \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE
```

輸出：

```
{
    "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE",
    "projectArn": "arn:aws:iotsitewise:us-west-2:123456789012:project/
a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE",
    "projectName": "Wind Farm 1",
    "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
    "projectDescription": "Contains asset visualizations for Wind Farm #1 for
 Example Corp.",
    "projectCreationDate": "2020-02-20T21:58:43.362246001Z",
    "projectLastUpdateDate": "2020-02-20T21:58:43.362246095Z"
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise Monitor 應用程式指南中的檢視專案詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeProject。

disassociate-assets

以下程式碼範例顯示如何使用 disassociate-assets。

AWS CLI

取消子資產與父資產的關聯

下列disassociate-assets範例會取消風力渦輪機資產與風力發電廠資產的關聯。

```
aws iotsitewise disassociate-assets \
    --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \
    --hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE \
    --child-asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的關聯資產。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateAssets。

get-asset-property-aggregates

以下程式碼範例顯示如何使用 get-asset-property-aggregates。

AWS CLI

擷取資產屬性的彙總平均值和計數值

下列get-asset-property-aggregates範例會擷取風力渦輪機資產的平均總功率，以及 1 小時內的總功率資料點計數。

```
aws iotsitewise get-asset-property-aggregates \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE \
    --start-date 1580849400 \
    --end-date 1580853000 \
    --aggregate-types AVERAGE COUNT \
    --resolution 1h
```

輸出：

```
{
    "aggregatedValues": [
        {
            "timestamp": 1580850000.0,
            "quality": "GOOD",
            "value": {
                "average": 8723.46538886233,
                "count": 12.0
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的查詢資產屬性彙總。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAssetPropertyAggregates。

get-asset-property-value-history

以下程式碼範例顯示如何使用 get-asset-property-value-history。

AWS CLI

擷取資產屬性的歷史值

下列get-asset-property-value-history範例會擷取 20 分鐘內風力渦輪機資產的總功率值。

```
aws iotsitewise get-asset-property-value-history \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE \
    --start-date 1580851800 \
    --end-date 1580853000
```

輸出：

```
{
    "assetPropertyValueHistory": [
        {
            "value": {
                "doubleValue": 7217.787046814844
            },
            "timestamp": {
                "timeInSeconds": 1580852100,
                "offsetInNanos": 0
            },
            "quality": "GOOD"
        },
        {
            "value": {
                "doubleValue": 6941.242811875451
            },
            "timestamp": {
                "timeInSeconds": 1580852400,
                "offsetInNanos": 0
            },
            "quality": "GOOD"
        },
        {
            "value": {
```

```
                "doubleValue": 6976.797662266717
            },
            "timestamp": {
                "timeInSeconds": 1580852700,
                "offsetInNanos": 0
            },
            "quality": "GOOD"
        },
        {
            "value": {
                "doubleValue": 6890.8677520453875
            },
            "timestamp": {
                "timeInSeconds": 1580853000,
                "offsetInNanos": 0
            },
            "quality": "GOOD"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的查詢歷史資產屬性值。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAssetPropertyValueHistory。

get-asset-property-value

以下程式碼範例顯示如何使用 get-asset-property-value。

AWS CLI

擷取資產屬性的目前值

下列get-asset-property-value範例會擷取風力渦輪機資產目前的總功率。

```
aws iotsitewise get-asset-property-value \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE
```

輸出：

```
{
```

```
    "propertyValue": {
        "value": {
            "doubleValue": 6890.8677520453875
        },
        "timestamp": {
            "timeInSeconds": 1580853000,
            "offsetInNanos": 0
        },
        "quality": "GOOD"
    }
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的查詢目前的資產屬性值。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAssetPropertyValue。

list-access-policies

以下程式碼範例顯示如何使用 list-access-policies。

AWS CLI

列出所有存取政策

下列list-access-policies範例列出入口網站管理員使用者的所有存取政策。

```
aws iotsitewise list-access-policies \
    --identity-type USER \
    --identity-id a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE
```

輸出：

```
{
    "accessPolicySummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE",
            "identity": {
                "user": {
                    "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
                }
            },
            "resource": {
```

```
                "portal": {
                    "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE"
                }
            },
            "permission": "ADMINISTRATOR"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的管理入口網站。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAccessPolicies。

list-asset-models

以下程式碼範例顯示如何使用 list-asset-models。

AWS CLI

列出所有資產模型

下列list-asset-models範例列出目前區域中 AWS 您帳戶中定義的所有資產模型。

```
aws iotsitewise list-asset-models
```

輸出：

```
{
    "assetModelSummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
            "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/
a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
            "name": "Wind Farm Model",
            "description": "Represents a wind farm that comprises many wind
 turbines",
            "creationDate": 1575671284.0,
            "lastUpdateDate": 1575671988.0,
            "status": {
                "state": "ACTIVE"
            }
        },
```

```
        {
            "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "name": "Wind Turbine Model",
            "description": "Represents a wind turbine manufactured by Example Corp",
            "creationDate": 1575671207.0,
            "lastUpdateDate": 1575686273.0,
            "status": {
                "state": "ACTIVE"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的列出所有資產模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAssetModels。

list-assets

以下程式碼範例顯示如何使用 `list-assets`。

AWS CLI

範例 1：列出所有最上層資產

下列`list-assets`範例會列出資產階層樹狀結構中最上層，並在目前區域中 AWS 的帳戶中定義的所有資產。

```
aws iotsitewise list-assets \
    --filter TOP_LEVEL
```

輸出：

```
{
    "assetSummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
            "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
            "name": "Wind Farm 1",
```

```
            "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
            "creationDate": 1575672453.0,
            "lastUpdateDate": 1575672453.0,
            "status": {
                "state": "ACTIVE"
            },
            "hierarchies": [
                {
                    "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
                    "name": "Wind Turbines"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的列出資產。

範例 2：根據資產模型列出所有資產

下列list-assets範例會根據資產模型列出所有資產，並在目前區域中 AWS 於您的帳戶中定義。

```
aws iotsitewise list-assets \
    --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

輸出：

```
{
    "assetSummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
            "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
            "name": "Wind Turbine 1",
            "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "creationDate": 1575671550.0,
            "lastUpdateDate": 1575686308.0,
            "status": {
                "state": "ACTIVE"
            },
            "hierarchies": []
        }
    ]
```

```
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的列出資產。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAssets。

`list-associated-assets`

以下程式碼範例顯示如何使用 `list-associated-assets`。

AWS CLI

列出與特定階層中的資產相關聯的所有資產

下列`list-associated-assets`範例列出與指定風力發電廠資產相關聯的所有風力發電機資產。

```
aws iotsitewise list-associated-assets \
    --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \
    --hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE
```

輸出：

```
{
    "assetSummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
            "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
            "name": "Wind Turbine 1",
            "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "creationDate": 1575671550.0,
            "lastUpdateDate": 1575686308.0,
            "status": {
                "state": "ACTIVE"
            },
            "hierarchies": []
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的列出與特定資產相關聯的資產。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAssociatedAssets。

list-dashboards

以下程式碼範例顯示如何使用 list-dashboards。

AWS CLI

列出專案中的所有儀表板

下列list-dashboards範例列出專案中定義的所有儀表板。

```
aws iotsitewise list-dashboards \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE
```

輸出：

```
{
    "dashboardSummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",
            "name": "Wind Farm",
            "creationDate": "2020-05-01T20:32:12.228476348Z",
            "lastUpdateDate": "2020-05-01T20:32:12.228476348Z"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise Monitor 應用程式指南中的檢視儀表板。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDashboards。

list-gateways

以下程式碼範例顯示如何使用 list-gateways。

AWS CLI

列出所有閘道

下列list-gateways範例列出目前區域中 AWS 您帳戶中定義的所有閘道。

```
aws iotsitewise list-gateways
```

輸出：

```
{
    "gatewaySummaries": [
        {
            "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
            "gatewayName": "ExampleCorpGateway",
            "gatewayCapabilitySummaries": [
                {
                    "capabilityNamespace": "iotsitewise:opcuacollector:1",
                    "capabilitySyncStatus": "IN_SYNC"
                }
            ],
            "creationDate": 1588369971.457,
            "lastUpdateDate": 1588369971.457
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的使用閘道擷取資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListGateways。

list-portals

以下程式碼範例顯示如何使用 `list-portals`。

AWS CLI

列出所有入口網站

下列`list-portals`範例列出目前區域中 AWS 您帳戶中定義的所有入口網站。

```
aws iotsitewise list-portals
```

輸出：

```
{
    "portalSummaries": [
```

```
        {
            "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
            "name": "WindFarmPortal",
            "description": "A portal that contains wind farm projects for Example
 Corp.",
            "startUrl": "https://a1b2c3d4-5678-90ab-cdef-
aaaaaEXAMPLE.app.iotsitewise.aws",
            "creationDate": "2020-02-04T23:01:52.90248068Z",
            "lastUpdateDate": "2020-02-04T23:01:52.90248078Z",
            "roleArn": "arn:aws:iam::123456789012:role/service-role/
MySiteWiseMonitorServiceRole"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的管理入口網站。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPortals。

`list-project-assets`

以下程式碼範例顯示如何使用 `list-project-assets`。

AWS CLI

列出與專案相關聯的所有資產

下列`list-project-assets`範例列出與風力發電廠專案相關聯的所有資產。

```
aws iotsitewise list-projects \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE
```

輸出：

```
{
    "assetIds": [
        "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE"
    ]
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise Monitor 應用程式指南中的將資產新增至專案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListProjectAssets。

`list-projects`

以下程式碼範例顯示如何使用 `list-projects`。

AWS CLI

列出入口網站中的所有專案

下列`list-projects`範例列出入口網站中定義的所有專案。

```
aws iotsitewise list-projects \
    --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

輸出：

```
{
    "projectSummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE",
            "name": "Wind Farm 1",
            "description": "Contains asset visualizations for Wind Farm #1 for
 Example Corp.",
            "creationDate": "2020-02-20T21:58:43.362246001Z",
            "lastUpdateDate": "2020-02-20T21:58:43.362246095Z"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise Monitor 應用程式指南中的檢視專案詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListProjects。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出資源的所有標籤

下列list-tags-for-resource範例列出風力渦輪機資產的所有標籤。

```
aws iotsitewise list-tags-for-resource \
    --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

輸出：

```
{
    "tags": {
        "Owner": "richard-roe"
    }
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的標記您的資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

put-logging-options

以下程式碼範例顯示如何使用 put-logging-options。

AWS CLI

指定記錄層級

下列put-logging-options範例會在 AWS IoT SiteWise 中啟用INFO關卡記錄。其他層級包括DEBUG和 OFF。

```
aws iotsitewise put-logging-options \
    --logging-options level=INFO
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的使用 Amazon CloudWatch Logs 監控IoT SiteWise。 AWS IoT SiteWise

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutLoggingOptions。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤加入資源

下列tag-resource範例會將擁有者標籤新增至風力渦輪機資產。這可讓您根據擁有資產的人員來控制對資產的存取。

```
aws iotsitewise tag-resource \
    --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --tags Owner=richard-roe
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的標記您的資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源移除標籤

下列untag-resource範例會從風力渦輪機資產移除擁有者標籤。

```
aws iotsitewise untag-resource \
    --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --tag-keys Owner
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的標記您的資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-access-policy

以下程式碼範例顯示如何使用 update-access-policy。

AWS CLI

授予專案的專案檢視器擁有權

下列update-access-policy範例會更新授予專案檢視器擁有權的存取政策。

```
aws iotsitewise update-access-policy \
    --access-policy-id a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE \
    --cli-input-json file://update-project-viewer-access-policy.json
```

update-project-viewer-access-policy.json 的內容：

```
{
    "accessPolicyIdentity": {
        "user": {
            "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
        }
    },
    "accessPolicyPermission": "ADMINISTRATOR",
    "accessPolicyResource": {
        "project": {
            "id": "a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE"
        }
    }
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT SiteWise Monitor 應用程式指南中的指派專案擁有者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAccessPolicy。

update-asset-model

以下程式碼範例顯示如何使用 update-asset-model。

AWS CLI

更新資產模型

下列update-asset-model範例會更新風力發電廠資產模型的描述。此範例包含模型的現有 IDs 和定義，因為 會使用新模型update-asset-model覆寫現有模型。

```
aws iotsitewise update-asset-model \
    --cli-input-json file://update-wind-farm-model.json
```

update-wind-farm-model.json 的內容：

```
{
    "assetModelName": "Wind Farm Model",
    "assetModelDescription": "Represents a wind farm that comprises many wind
 turbines",
    "assetModelProperties": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-88888EXAMPLE",
            "name": "Region",
            "dataType": "STRING",
            "type": {
                "attribute": {}
            }
        },
        {
            "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
            "name": "Total Generated Power",
            "dataType": "DOUBLE",
            "unit": "kW",
            "type": {
                "metric": {
                    "expression": "sum(power)",
                    "variables": [
                        {
                            "name": "power",
                            "value": {
                                "hierarchyId": "a1b2c3d4-5678-90ab-
cdef-77777EXAMPLE",
                                "propertyId": "a1b2c3d4-5678-90ab-cdef-66666EXAMPLE"
                            }
                        }
                    ],
                    "window": {
                        "tumbling": {
                            "interval": "1h"
                        }
                    }
                }
            }
```

```
        }
    ],
    "assetModelHierarchies": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
            "name": "Wind Turbines",
            "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
        }
    ]
}
```

輸出：

```
{
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/
a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetModelStatus": {
        "state": "CREATING"
    }
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的更新資產模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAssetModel。

update-asset-property

以下程式碼範例顯示如何使用 update-asset-property。

AWS CLI

範例 1：更新資產屬性的別名

下列update-asset-property範例會更新風力渦輪機資產的電源屬性別名。

```
aws iotsitewise update-asset-property \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --property-id a1b2c3d4-5678-90ab-cdef-55555EXAMPLE \
    --property-alias "/examplecorp/windfarm/1/turbine/1/power" \
    --property-notification-state DISABLED
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的將工業資料串流映射至資產屬性。

範例 2：啟用資產屬性通知

下列update-asset-property範例會啟用風力渦輪機資產功率屬性的資產屬性更新通知。屬性值更新會發佈至 MQTT 主題 $aws/sitewise/asset-models/<assetModelId>/assets/<assetId>/properties/<propertyId>，其中每個 ID 都會被資產屬性的屬性、資產和模型 ID 取代。

```
aws iotsitewise update-asset-property \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE \
    --property-notification-state ENABLED \
    --property-alias "/examplecorp/windfarm/1/turbine/1/power"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的與其他 服務互動。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAssetProperty。

update-asset

以下程式碼範例顯示如何使用 update-asset。

AWS CLI

更新資產的名稱

下列update-asset範例會更新風力渦輪機資產的名稱。

```
aws iotsitewise update-asset \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --asset-name "Wind Turbine 2"
```

輸出：

```
{
    "assetStatus": {
        "state": "UPDATING"
    }
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的更新資產。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateAsset。

update-dashboard

以下程式碼範例顯示如何使用 update-dashboard。

AWS CLI

更新儀表板

下列update-dashboard範例會變更儀表板折線圖的標題，顯示風力發電廠產生的總功率。

```
aws iotsitewise update-dashboard \
    --project-id a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE \
    --dashboard-name "Wind Farm" \
    --dashboard-definition file://update-wind-farm-dashboard.json
```

update-wind-farm-dashboard.json 的內容：

```
{
    "widgets": [
        {
            "type": "monitor-line-chart",
            "title": "Total Generated Power",
            "x": 0,
            "y": 0,
            "height": 3,
            "width": 3,
            "metrics": [
                {
                    "label": "Power",
                    "type": "iotsitewise",
                    "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
                    "propertyId": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE"
                }
            ]
        }
    ]
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的建立儀表板 (CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDashboard。

update-gateway-capability-configuration

以下程式碼範例顯示如何使用 update-gateway-capability-configuration。

AWS CLI

更新閘道功能

下列update-gateway-capability-configuration範例會設定具有下列屬性的 OPC-UA 來源：

信任任何憑證。使用 Basic256 演算法保護訊息。使用 SignAndEncrypt 模式保護連線。使用存放在 AWS Secrets Manager 秘密中的身分驗證憑證。

```
aws iotsitewise update-gateway-capability-configuration \
    --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
    --capability-namespace "iotsitewise:opcuacollector:1" \
    --capability-configuration file://opc-ua-capability-configuration.json
```

opc-ua-capability-configuration.json 的內容：

```
{
    "sources": [
        {
            "name": "Wind Farm #1",
            "endpoint": {
                "certificateTrust": {
                    "type": "TrustAny"
                },
                "endpointUri": "opc.tcp://203.0.113.0:49320",
                "securityPolicy": "BASIC256",
                "messageSecurityMode": "SIGN_AND_ENCRYPT",
                "identityProvider": {
                    "type": "Username",
                    "usernameSecretArn": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:greengrass-windfarm1-auth-1ABCDE"
                },
                "nodeFilterRules": []
```

```
            },
            "measurementDataStreamPrefix": ""
        }
    ]
}
```

輸出：

```
{
    "capabilityNamespace": "iotsitewise:opcuacollector:1",
    "capabilitySyncStatus": "OUT_OF_SYNC"
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的設定資料來源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateGatewayCapabilityConfiguration。

update-gateway

以下程式碼範例顯示如何使用 update-gateway。

AWS CLI

更新閘道的名稱

下列update-gateway範例會更新閘道的名稱。

```
aws iotsitewise update-gateway \
    --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
    --gateway-name ExampleCorpGateway1
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的使用閘道擷取資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateGateway。

update-portal

以下程式碼範例顯示如何使用 update-portal。

AWS CLI

更新入口網站的詳細資訊

下列update-portal範例會更新風力發電廠公司的 Web 入口網站。

```
aws iotsitewise update-portal \
    --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE \
    --portal-name WindFarmPortal \
    --portal-description "A portal that contains wind farm projects for Example
 Corp." \
    --portal-contact-email support@example.com \
    --role-arn arn:aws:iam::123456789012:role/MySiteWiseMonitorServiceRole
```

輸出：

```
{
    "portalStatus": {
        "state": "UPDATING"
    }
}
```

如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的管理入口網站。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePortal。

update-project

以下程式碼範例顯示如何使用 update-project。

AWS CLI

更新專案的詳細資訊

下列update-project範例會更新風力發電廠專案。

```
aws iotsitewise update-project \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE \
    --project-name "Wind Farm 1" \
    --project-description "Contains asset visualizations for Wind Farm #1 for
 Example Corp."
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT SiteWise Monitor 應用程式指南中的變更專案詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateProject。

AWS IoT Things Graph 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS IoT Things Graph。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-entity-to-thing`

以下程式碼範例顯示如何使用 `associate-entity-to-thing`。

AWS CLI

將物件與裝置建立關聯

下列`associate-entity-to-thing`範例會將物件與裝置建立關聯。此範例使用公有命名空間中的動作感應器裝置。

```
aws iotthingsgraph associate-entity-to-thing \
    --thing-name "MotionSensorName" \
    --entity-id "urn:tdm:aws/examples:Device:HCSR501MotionSensor"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的建立和上傳模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateEntityToThing。

create-flow-template

以下程式碼範例顯示如何使用 create-flow-template。

AWS CLI

建立流程

下列create-flow-template範例會建立流程（工作流程）。的值MyFlowDefinition是建立流程模型的 GraphQL。

```
aws iotthingsgraph create-flow-template \
    --definition language=GRAPHQL,text="MyFlowDefinition"
```

輸出：

```
{
    "summary": {
        "createdAt": 1559248067.545,
        "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
        "revisionNumber": 1
    }
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的使用流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFlowTemplate。

create-system-instance

以下程式碼範例顯示如何使用 create-system-instance。

AWS CLI

建立系統執行個體

下列create-system-instance範例會建立系統執行個體。的值MySystemInstanceDefinition是建立系統執行個體模型的 GraphQL。

```
aws iotthingsgraph create-system-instance -\
    -definition language=GRAPHQL,text="MySystemInstanceDefinition" \
    --target CLOUD \
```

```
    --flow-actions-role-arn myRoleARN
```

輸出：

```
{
    "summary": {
        "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room218",
        "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/default/
Room218",
        "status": "NOT_DEPLOYED",
        "target": "CLOUD",
        "createdAt": 1559249315.208,
        "updatedAt": 1559249315.208
    }
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的使用系統和流程組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSystemInstance。

create-system-template

以下程式碼範例顯示如何使用 create-system-template。

AWS CLI

建立系統

下列create-system-template範例會建立系統。MySystemDefinition 的值是建立系統模型的 GraphQL。

```
aws iotthingsgraph create-system-template \
    --definition language=GRAPHQL,text="MySystemDefinition"
```

輸出：

```
{
    "summary": {
        "createdAt": 1559249776.254,
        "id": "urn:tdm:us-west-2/123456789012/default:System:MySystem",
        "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/
MySystem",
```

```
        "revisionNumber": 1
    }
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的建立系統。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSystemTemplate。

delete-flow-template

以下程式碼範例顯示如何使用 delete-flow-template。

AWS CLI

刪除流程

下列delete-flow-template範例會刪除流程 （工作流程）。

```
aws iotthingsgraph delete-flow-template \
    --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的 IoT Things Graph 實體、流程、系統和部署的生命週期管理。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFlowTemplate。

delete-namespace

以下程式碼範例顯示如何使用 delete-namespace。

AWS CLI

刪除命名空間

下列delete-namespace範例會刪除命名空間。

```
aws iotthingsgraph delete-namespace
```

輸出：

```
{
   "namespaceArn": "arn:aws:iotthingsgraph:us-west-2:123456789012",
   "namespaceName": "us-west-2/123456789012/default"
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的 IoT Things Graph 實體、流程、系統和部署的生命週期管理。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNamespace。

delete-system-instance

以下程式碼範例顯示如何使用 delete-system-instance。

AWS CLI

刪除系統執行個體

下列delete-system-instance範例會刪除系統執行個體。

```
aws iotthingsgraph delete-system-instance \
    --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的 IoT Things Graph 實體、流程、系統和部署的生命週期管理。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSystemInstance。

delete-system-template

以下程式碼範例顯示如何使用 delete-system-template。

AWS CLI

刪除系統

下列delete-system-template範例會刪除系統。

```
aws iotthingsgraph delete-system-template \
```

```
    --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的 IoT Things Graph 實體、流程、系統和部署的生命週期管理。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSystemTemplate。

deploy-system-instance

以下程式碼範例顯示如何使用 deploy-system-instance。

AWS CLI

部署系統執行個體

下列delete-system-template範例會部署系統執行個體。

```
aws iotthingsgraph deploy-system-instance \
    --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

輸出：

```
{
   "summary": {
      "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment:Room218",
      "createdAt": 1559249776.254,
      "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room218",
      "status": "DEPLOYED_IN_TARGET",
      "target": "CLOUD",
      "updatedAt": 1559249776.254
   }
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的使用系統和流程組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeploySystemInstance。

deprecate-flow-template

以下程式碼範例顯示如何使用 deprecate-flow-template。

AWS CLI

棄用流程

下列deprecate-flow-template範例會取代流程（工作流程）。

```
aws iotthingsgraph deprecate-flow-template \
    --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的 IoT Things Graph 實體、流程、系統和部署的生命週期管理。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeprecateFlowTemplate。

deprecate-system-template

以下程式碼範例顯示如何使用 deprecate-system-template。

AWS CLI

棄用系統

下列deprecate-system-template範例會取代系統。

```
aws iotthingsgraph deprecate-system-template \
    --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的 IoT Things Graph 實體、流程、系統和部署的生命週期管理。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeprecateSystemTemplate。

describe-namespace

以下程式碼範例顯示如何使用 describe-namespace。

AWS CLI

取得命名空間的描述

下列describe-namespace範例會取得命名空間的描述。

```
aws iotthingsgraph describe-namespace
```

輸出：

```
{
    "namespaceName": "us-west-2/123456789012/default",
    "trackingNamespaceName": "aws",
    "trackingNamespaceVersion": 1,
    "namespaceVersion": 5
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的命名空間。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeNamespace。

dissociate-entity-from-thing

以下程式碼範例顯示如何使用 dissociate-entity-from-thing。

AWS CLI

取消物件與裝置的關聯

下列dissociate-entity-from-thing範例會取消物件與裝置的關聯。

```
aws iotthingsgraph dissociate-entity-from-thing \
    --thing-name "MotionSensorName" \
    --entity-type "DEVICE"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的建立和上傳模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DissociateEntityFromThing。

get-entities

以下程式碼範例顯示如何使用 get-entities。

AWS CLI

取得實體的定義

下列get-entities範例會取得裝置模型的定義。

```
aws iotthingsgraph get-entities \
    --ids "urn:tdm:aws/examples:DeviceModel:MotionSensor"
```

輸出：

```
{
    "descriptions": [
        {
            "id": "urn:tdm:aws/examples:DeviceModel:MotionSensor",
            "type": "DEVICE_MODEL",
            "createdAt": 1559256190.599,
            "definition": {
                "language": "GRAPHQL",
                "text": "##\n# Specification of motion sensor devices interface.\n##
\ntype MotionSensor @deviceModel(id: \"urn:tdm:aws/examples:deviceModel:MotionSensor
\",\n        capability: \"urn:tdm:aws/examples:capability:MotionSensorCapability\")
 {ignore:void}"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的建立和上傳模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetEntities。

get-flow-template-revisions

以下程式碼範例顯示如何使用 get-flow-template-revisions。

AWS CLI

取得流程的修訂資訊

下列get-flow-template-revisions範例取得流程（工作流程）的修訂資訊。

```
aws iotthingsgraph get-flow-template-revisions \
```

```
    --id urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow
```

輸出：

```
{
    "summaries": [
        {
            "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
            "revisionNumber": 1,
            "createdAt": 1559247540.292
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的使用流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFlowTemplateRevisions。

get-flow-template

以下程式碼範例顯示如何使用 get-flow-template。

AWS CLI

取得流程定義

下列get-flow-template範例取得流程的定義（工作流程）。

```
aws iotthingsgraph get-flow-template \
    --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

輸出：

```
{
    "description": {
        "summary": {
            "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
            "revisionNumber": 1,
            "createdAt": 1559247540.292
        },
        "definition": {
            "language": "GRAPHQL",
```

```
            "text": "{\nquery MyFlow($camera: string!, $screen: string!)
 @workflowType(id: \"urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow\")
 @annotation(type: \"tgc:FlowEvent\", id: \"sledged790c1b2bcd949e09da0c9bfc077f79d
\", x: 1586, y: 653) @triggers(definition: \"{MotionSensor(description:
 \\\"\\\") @position(x: 1045, y: 635.6666564941406) {\\n  condition(expr:
 \\\"devices[name == \\\\\\\"motionSensor\\\\\\\"].events[name == \\\
\\\"StateChanged\\\\\\\"].lastEvent\\\")\\n  action(expr: \\\"\\\")\
\n}}\") {\n  variables {\n    cameraResult @property(id: \"urn:tdm:aws/
examples:property:CameraStateProperty\")\n  }\n  steps {\n    step(name: \"Camera
\", outEvent: [\"sledged790c1b2bcd949e09da0c9bfc077f79d\"]) @position(x: 1377,
 y: 638.6666564941406) {\n      DeviceActivity(deviceModel: \"urn:tdm:aws/
examples:deviceModel:Camera\", out: \"cameraResult\", deviceId: \"${camera}\")
 {\n        capture\n      }\n    }\n    step(name: \"Screen\", inEvent:
 [\"sledged790c1b2bcd949e09da0c9bfc077f79d\"]) @position(x: 1675.6666870117188,
 y: 637.9999847412109) {\n      DeviceActivity(deviceModel: \"urn:tdm:aws/
examples:deviceModel:Screen\", deviceId: \"${screen}\") {\n        display(imageUrl:
 \"${cameraResult.lastClickedImage}\")\n      }\n    }\n  }\n}\n}"
        },
        "validatedNamespaceVersion": 5
    }
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的使用流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFlowTemplate。

get-namespace-deletion-status

以下程式碼範例顯示如何使用 get-namespace-deletion-status。

AWS CLI

取得命名空間刪除任務的狀態

下列get-namespace-deletion-status範例會取得命名空間刪除任務的狀態。

```
aws iotthingsgraph get-namespace-deletion-status
```

輸出：

```
{
   "namespaceArn": "arn:aws:iotthingsgraph:us-west-2:123456789012",
```

```
    "namespaceName": "us-west-2/123456789012/default"
    "status": "SUCCEEDED "
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的命名空間。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetNamespaceDeletionStatus。

get-system-instance

以下程式碼範例顯示如何使用 get-system-instance。

AWS CLI

取得系統執行個體

下列get-system-instance範例會取得系統執行個體的定義。

```
aws iotthingsgraph get-system-instance \
    --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

輸出：

```
{
    "description": {
        "summary": {
            "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room218",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room218",
            "status": "NOT_DEPLOYED",
            "target": "CLOUD",
            "createdAt": 1559249315.208,
            "updatedAt": 1559249315.208
        },
        "definition": {
            "language": "GRAPHQL",
            "text": "{\r\nquery Room218 @deployment(id: \"urn:tdm:us-
west-2/123456789012/default:Deployment:Room218\", systemId: \"urn:tdm:us-
west-2/123456789012/default:System:SecurityFlow\") {\r\n    motionSensor(deviceId:
 \"MotionSensorName\")\r\n    screen(deviceId: \"ScreenName\")\r\n
 camera(deviceId: \"CameraName\") \r\n    triggers {MotionEventTrigger(description:
 \"a trigger\") {  \r\n    condition(expr: \"devices[name ==
```

```
 'motionSensor'].events[name == 'StateChanged'].lastEvent\") \r\n    action(expr:
 \"ThingsGraph.startFlow('SecurityFlow', bindings[name == 'camera'].deviceId,
 bindings[name == 'screen'].deviceId)\")\r\n    }\r\n   }\r\n  }\r\n  }"
        },
        "metricsConfiguration": {
            "cloudMetricEnabled": false
        },
        "validatedNamespaceVersion": 5,
        "flowActionsRoleArn": "arn:aws:iam::123456789012:role/ThingsGraphRole"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的使用系統和流程組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSystemInstance。

get-system-template-revisions

以下程式碼範例顯示如何使用 get-system-template-revisions。

AWS CLI

取得系統的修訂資訊

下列get-system-template-revisions範例會取得系統的修訂資訊。

```
aws iotthingsgraph get-system-template-revisions \
    --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

輸出：

```
{
    "summaries": [
        {
            "id": "urn:tdm:us-west-2/123456789012/default:System:MySystem",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/
MySystem",
            "revisionNumber": 1,
            "createdAt": 1559247540.656
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的使用系統和流程組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSystemTemplateRevisions。

get-system-template

以下程式碼範例顯示如何使用 get-system-template。

AWS CLI

取得系統

下列get-system-template範例會取得系統的定義。

```
aws iotthingsgraph get-system-template \
    --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

輸出：

```
{
    "description": {
        "summary": {
            "id": "urn:tdm:us-west-2/123456789012/default:System:MySystem",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/
MyFlow",
            "revisionNumber": 1,
            "createdAt": 1559247540.656
        },
        "definition": {
            "language": "GRAPHQL",
            "text": "{\ntype MySystem @systemType(id: \"urn:tdm:us-
west-2/123456789012/default:System:MySystem\", description: \"\") {\n  camera:
 Camera @thing(id: \"urn:tdm:aws/examples:deviceModel:Camera\")\n  screen:
 Screen @thing(id: \"urn:tdm:aws/examples:deviceModel:Screen\")\n  motionSensor:
 MotionSensor @thing(id: \"urn:tdm:aws/examples:deviceModel:MotionSensor
\")\n  MyFlow: MyFlow @workflow(id: \"urn:tdm:us-west-2/123456789012/
default:Workflow:MyFlow\")\n}\n}"
        },
        "validatedNamespaceVersion": 5
    }
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的使用系統和流程組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSystemTemplate。

get-upload-status

以下程式碼範例顯示如何使用 get-upload-status。

AWS CLI

若要取得實體上傳的狀態

下列get-upload-status範例會取得實體上傳操作的狀態。的值MyUploadId是 upload-entity-definitions操作傳回的 ID 值。

```
aws iotthingsgraph get-upload-status \
    --upload-id "MyUploadId"
```

輸出：

```
{
    "namespaceName": "us-west-2/123456789012/default",
    "namespaceVersion": 5,
    "uploadId": "f6294f1e-b109-4bbe-9073-f451a2dda2da",
    "uploadStatus": "SUCCEEDED"
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的建立實體模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetUploadStatus。

list-flow-execution-messages

以下程式碼範例顯示如何使用 list-flow-execution-messages。

AWS CLI

取得流程執行中事件的相關資訊

下列list-flow-execution-messages範例取得流程執行中事件的相關資訊。

```
aws iotthingsgraph list-flow-execution-messages \
```

```
    --flow-execution-id "urn:tdm:us-west-2/123456789012/
default:Workflow:SecurityFlow_2019-05-11T19:39:55.317Z_MotionSensor_69b151ad-
a611-42f5-ac21-fe537f9868ad"
```

輸出：

```
{
    "messages": [
        {
         "eventType": "EXECUTION_STARTED",
         "messageId": "f6294f1e-b109-4bbe-9073-f451a2dda2da",
         "payload": "Flow execution started",
         "timestamp": 1559247540.656
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的使用流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFlowExecutionMessages。

list-tags-for-resource

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出資源的所有標籤

下列`list-tags-for-resource`範例列出 an AWS IoT Things Graph 資源的所有標籤。

```
aws iotthingsgraph list-tags-for-resource \
    --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room218"
```

輸出：

```
{
   "tags": [
      {
         "key": "Type",
         "value": "Residential"
```

```
      }
   ]
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的標記您的 IoT Things Graph 資源。AWS IoT

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

search-entities

以下程式碼範例顯示如何使用 search-entities。

AWS CLI

搜尋實體

下列search-entities範例會搜尋類型 的所有實體EVENT。

```
aws iotthingsgraph search-entities \
    --entity-types "EVENT"
```

輸出：

```
{
    "descriptions": [
        {
            "id": "urn:tdm:aws/examples:Event:MotionSensorEvent",
            "type": "EVENT",
            "definition": {
                "language": "GRAPHQL",
                "text": "##\n# Description of events emitted by motion
 sensor.\n##\ntype MotionSensorEvent @eventType(id: \"urn:tdm:aws/
examples:event:MotionSensorEvent\",\n            payload: \"urn:tdm:aws/
examples:property:MotionSensorStateProperty\") {ignore:void}"
            }
        },
        {
            "id": "urn:tdm:us-west-2/123456789012/
default:Event:CameraClickedEventV2",
            "type": "EVENT",
            "definition": {
                "language": "GRAPHQL",
```

```
                "text": "type CameraClickedEventV2 @eventType(id: \"urn:tdm:us-
west-2/123456789012/default:event:CameraClickedEventV2\",\r\npayload:
 \"urn:tdm:aws:Property:Boolean\"){ignore:void}"
            }
        },
        {
            "id": "urn:tdm:us-west-2/123456789012/
default:Event:MotionSensorEventV2",
            "type": "EVENT",
            "definition": {
                "language": "GRAPHQL",
                "text": "# Event emitted by the motion sensor.\r\ntype
 MotionSensorEventV2 @eventType(id: \"urn:tdm:us-west-2/123456789012/
default:event:MotionSensorEventV2\",\r\npayload: \"urn:tdm:us-west-2/123456789012/
default:property:MotionSensorStateProperty2\") {ignore:void}"
            }
        }
    ],
    "nextToken": "urn:tdm:us-west-2/123456789012/default:Event:MotionSensorEventV2"
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的 IoT Things Graph 資料模型參考。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchEntities。

search-flow-executions

以下程式碼範例顯示如何使用 search-flow-executions。

AWS CLI

搜尋流程執行

下列search-flow-executions範例會搜尋指定系統執行個體中流程的所有執行。

```
aws iotthingsgraph search-flow-executions \
    --system-instance-id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

輸出：

```
{
   "summaries": [
```

```
      {
         "createdAt": 1559247540.656,
         "flowExecutionId": "f6294f1e-b109-4bbe-9073-f451a2dda2da",
         "flowTemplateId": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
         "status": "RUNNING ",
         "systemInstanceId": "urn:tdm:us-west-2/123456789012/
default:System:MySystem",
         "updatedAt": 1559247540.656
      }
   ]
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的使用系統和流程組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchFlowExecutions。

search-flow-templates

以下程式碼範例顯示如何使用 search-flow-templates。

AWS CLI

搜尋流程 （或工作流程）

下列search-flow-templates範例會搜尋包含攝影機裝置模型的所有流程 （工作流程）。

```
aws iotthingsgraph search-flow-templates \
    --filters name="DEVICE_MODEL_ID",value="urn:tdm:aws/examples:DeviceModel:Camera"
```

輸出：

```
{
    "summaries": [
        {
            "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
            "revisionNumber": 1,
            "createdAt": 1559247540.292
        },
        {
            "id": "urn:tdm:us-west-2/123456789012/default:Workflow:SecurityFlow",
            "revisionNumber": 3,
            "createdAt": 1548283099.27
        }
```

```
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的使用流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchFlowTemplates。

search-system-instances

以下程式碼範例顯示如何使用 search-system-instances。

AWS CLI

搜尋系統執行個體

下列search-system-instances範例會搜尋包含指定系統的所有系統執行個體。

```
aws iotthingsgraph search-system-instances \
    --filters name="SYSTEM_TEMPLATE_ID",value="urn:tdm:us-west-2/123456789012/
default:System:SecurityFlow"
```

輸出：

```
{
    "summaries": [
        {
            "id": "urn:tdm:us-west-2/123456789012/
default:Deployment:DeploymentForSample",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/DeploymentForSample",
            "status": "NOT_DEPLOYED",
            "target": "GREENGRASS",
            "greengrassGroupName": "ThingsGraphGrnGr",
            "createdAt": 1555716314.707,
            "updatedAt": 1555716314.707
        },
        {
            "id": "urn:tdm:us-west-2/123456789012/
default:Deployment:MockDeployment",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/MockDeployment",
            "status": "DELETED_IN_TARGET",
            "target": "GREENGRASS",
```

```
            "greengrassGroupName": "ThingsGraphGrnGr",
            "createdAt": 1549416462.049,
            "updatedAt": 1549416722.361,
            "greengrassGroupId": "01d04b07-2a51-467f-9d03-0c90b3cdcaaf",
            "greengrassGroupVersionId": "7365aed7-2d3e-4d13-aad8-75443d45eb05"
        },
        {
            "id": "urn:tdm:us-west-2/123456789012/
default:Deployment:MockDeployment2",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/MockDeployment2",
            "status": "DEPLOYED_IN_TARGET",
            "target": "GREENGRASS",
            "greengrassGroupName": "ThingsGraphGrnGr",
            "createdAt": 1549572385.774,
            "updatedAt": 1549572418.408,
            "greengrassGroupId": "01d04b07-2a51-467f-9d03-0c90b3cdcaaf",
            "greengrassGroupVersionId": "bfa70ab3-2bf7-409c-a4d4-bc8328ae5b86"
        },
        {
            "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room215",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room215",
            "status": "NOT_DEPLOYED",
            "target": "GREENGRASS",
            "greengrassGroupName": "ThingsGraphGG",
            "createdAt": 1547056918.413,
            "updatedAt": 1547056918.413
        },
        {
            "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room218",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room218",
            "status": "NOT_DEPLOYED",
            "target": "CLOUD",
            "createdAt": 1559249315.208,
            "updatedAt": 1559249315.208
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的使用系統和流程組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchSystemInstances。

search-system-templates

以下程式碼範例顯示如何使用 search-system-templates。

AWS CLI

搜尋系統

下列search-system-templates範例會搜尋包含指定流程的所有系統。

```
aws iotthingsgraph search-system-templates \
    --filters name="FLOW_TEMPLATE_ID",value="urn:tdm:us-west-2/123456789012/
default:Workflow:SecurityFlow"
```

輸出：

```
{
    "summaries": [
        {
            "id": "urn:tdm:us-west-2/123456789012/default:System:SecurityFlow",
            "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/
SecurityFlow",
            "revisionNumber": 1,
            "createdAt": 1548283099.433
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的使用流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchSystemTemplates。

search-things

以下程式碼範例顯示如何使用 search-things。

AWS CLI

搜尋與裝置和裝置模型相關聯的物件

下列search-things範例會搜尋與 HCSR501MotionSensor 裝置相關聯的所有物件。

```
aws iotthingsgraph search-things \
```

```
    --entity-id "urn:tdm:aws/examples:Device:HCSR501MotionSensor"
```

輸出：

```
{
    "things": [
        {
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MotionSensor1",
            "thingName": "MotionSensor1"
        },
        {
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/TG_MS",
            "thingName": "TG_MS"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的建立和上傳模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchThings。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

為資源建立標籤

下列tag-resource範例會為指定的資源建立標籤。

```
aws iotthingsgraph tag-resource \
    --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room218" \
    --tags key="Type",value="Residential"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的標記您的 IoT Things Graph 資源。
AWS IoT

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

undeploy-system-instance

以下程式碼範例顯示如何使用 undeploy-system-instance。

AWS CLI

從其目標中取消部署系統執行個體

下列undeploy-system-instance範例會從其目標中移除系統執行個體。

```
aws iotthingsgraph undeploy-system-instance \
    --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room215"
```

輸出：

```
{
    "summary": {
        "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room215",
        "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/default/
Room215",
        "status": "PENDING_DELETE",
        "target": "GREENGRASS",
        "greengrassGroupName": "ThingsGraphGrnGr",
        "createdAt": 1553189694.255,
        "updatedAt": 1559344549.601,
        "greengrassGroupId": "01d04b07-2a51-467f-9d03-0c90b3cdcaaf",
        "greengrassGroupVersionId": "731b371d-d644-4b67-ac64-3934e99b75d7"
    }
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的 IoT Things Graph 實體、流程、系統和部署的生命週期管理。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UndeploySystemInstance。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

移除資源的標籤

下列untag-resource範例會移除指定資源的標籤。

```
aws iotthingsgraph untag-resource \
    --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room218" \
    --tag-keys "Type"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的標記您的 IoT Things Graph 資源。AWS IoT

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-flow-template

以下程式碼範例顯示如何使用 update-flow-template。

AWS CLI

更新流程

下列update-flow-template範例會更新流程（工作流程）。的值MyFlowDefinition是建立流程模型的 GraphQL。

```
aws iotthingsgraph update-flow-template \
    --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow" \
    --definition language=GRAPHQL,text="MyFlowDefinition"
```

輸出：

```
{
    "summary": {
        "createdAt": 1559248067.545,
        "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
        "revisionNumber": 2
    }
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的使用流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateFlowTemplate。

update-system-template

以下程式碼範例顯示如何使用 update-system-template。

AWS CLI

更新系統

下列update-system-template範例會更新系統。的值MySystemDefinition是建立系統模型的 GraphQL。

```
aws iotthingsgraph update-system-template \
    --id "urn:tdm:us-west-2/123456789012/default:System:MySystem" \
    --definition language=GRAPHQL,text="MySystemDefinition"
```

輸出：

```
{
    "summary": {
        "createdAt": 1559249776.254,
        "id": "urn:tdm:us-west-2/123456789012/default:System:MySystem",
        "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/
MySystem",
        "revisionNumber": 2
    }
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的建立系統。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSystemTemplate。

upload-entity-definitions

以下程式碼範例顯示如何使用 upload-entity-definitions。

AWS CLI

上傳實體定義

下列upload-entity-definitions範例會將實體定義上傳至您的命名空間。的值MyEntityDefinitions是建立實體模型的 GraphQL。

```
aws iotthingsgraph upload-entity-definitions \
```

```
    --document language=GRAPHQL,text="MyEntityDefinitions"
```

輸出：

```
{
    "uploadId": "f6294f1e-b109-4bbe-9073-f451a2dda2da"
}
```

如需詳細資訊，請參閱 AWS IoT Things Graph 使用者指南中的建立實體模型。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UploadEntityDefinitions。

AWS IoT Wireless 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS IoT Wireless。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-aws-account-with-partner-account`

以下程式碼範例顯示如何使用 `associate-aws-account-with-partner-account`。

AWS CLI

將合作夥伴帳戶與 AWS 您的帳戶建立關聯

下列`associate-aws-account-with-partner-account`範例會將下列 Sidewalk 帳戶登入資料與 AWS 您的帳戶建立關聯。

```
aws iotwireless associate-aws-account-with-partner-account \
    --sidewalk
 AmazonId="12345678901234",AppServerPrivateKey="a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78
```

輸出：

```
{
    "Sidewalk": {
        "AmazonId": "12345678901234",
        "AppServerPrivateKey":
 "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Amazon Sidewalk Integration for AWS IoT Core。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateAwsAccountWithPartnerAccount。

associate-wireless-device-with-thing

以下程式碼範例顯示如何使用 associate-wireless-device-with-thing。

AWS CLI

將物件與無線裝置建立關聯

下列associate-wireless-device-with-thing範例會將物件與具有指定 ID 的無線裝置建立關聯。

```
aws iotwireless associate-wireless-device-with-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MyIoTWirelessThing"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將閘道和無線裝置新增至 IoT Core for LoRaWAN。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateWirelessDeviceWithThing。

associate-wireless-gateway-with-certificate

以下程式碼範例顯示如何使用 associate-wireless-gateway-with-certificate。

AWS CLI

將憑證與無線閘道建立關聯

以下內容會將無線閘道與憑證建立associate-wireless-gateway-with-certificate關聯。

```
aws iotwireless associate-wireless-gateway-with-certificate \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --iot-certificate-
id "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
```

輸出：

```
{
    "IotCertificateId":
 "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將閘道和無線裝置新增至 IoT Core for LoRaWAN。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateWirelessGatewayWithCertificate。

associate-wireless-gateway-with-thing

以下程式碼範例顯示如何使用 associate-wireless-gateway-with-thing。

AWS CLI

將物件與無線閘道建立關聯

下列associate-wireless-gateway-with-thing範例會將物件與無線閘道建立關聯。

```
aws iotwireless associate-wireless-gateway-with-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MyIoTWirelessThing"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將閘道和無線裝置新增至 IoT Core for LoRaWAN。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateWirelessGatewayWithThing。

create-destination

以下程式碼範例顯示如何使用 create-destination。

AWS CLI

建立 IoT 無線目的地

下列create-destination範例會建立將裝置訊息映射至 AWS IoT 規則的目的地。執行此命令之前，您必須已建立 IAM 角色，為 AWS IoT Core for LoRaWAN 提供將資料傳送至 AWS IoT 規則所需的許可。

```
aws iotwireless create-destination \
    --name IoTWirelessDestination \
    --expression-type RuleName \
    --expression IoTWirelessRule \
    --role-arn arn:aws:iam::123456789012:role/IoTWirelessDestinationRole
```

輸出：

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination",
    "Name": "IoTWirelessDestination"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將目的地新增至 IoT Core for LoRaWAN。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDestination。

create-device-profile

以下程式碼範例顯示如何使用 create-device-profile。

AWS CLI

建立新的裝置設定檔

下列create-device-profile範例會建立新的 IoT 無線裝置設定檔。

```
aws iotwireless create-device-profile
```

輸出：

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將設定檔新增至 IoT Core for LoRaWAN。
AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDeviceProfile。

create-service-profile

以下程式碼範例顯示如何使用 create-service-profile。

AWS CLI

建立新的服務設定檔

下列create-service-profile範例會建立新的 IoT 無線服務設定檔。

```
aws iotwireless create-service-profile
```

輸出：

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將設定檔新增至 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateServiceProfile。

create-wireless-device

以下程式碼範例顯示如何使用 create-wireless-device。

AWS CLI

建立 IoT 無線裝置

下列create-wireless-device範例會建立 LoRaWAN 類型的無線裝置資源。

```
aws iotwireless create-wireless-device \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "Description": "My LoRaWAN wireless device"
    "DestinationName": "IoTWirelessDestination"
    "LoRaWAN": {
        "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",
        "ServiceProfileId": "fe98dc76-cd12-001e-2d34-5550432da100",
        "OtaaV1_1": {
            "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",
            "JoinEui": "b4c231a359bc2e3d",
            "NwkKey": "01c3f004a2d6efffe32c4eda14bcd2b4"
        },
        "DevEui": "ac12efc654d23fc2"
    },
    "Name": "SampleIoTWirelessThing"
    "Type": LoRaWAN
}
```

輸出：

```
{
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f",
```

```
    "Id": "1ffd32c8-8130-4194-96df-622f072a315f"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateWirelessDevice。

create-wireless-gateway-task-definition

以下程式碼範例顯示如何使用 create-wireless-gateway-task-definition。

AWS CLI

建立無線閘道任務定義

以下內容會針對具有指定目前版本的所有閘道，使用此任務定義create-wireless-gateway-task-definition自動建立任務。

```
aws iotwireless create-wireless-gateway-task-definition \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "AutoCreateTasks": true,
    "Name": "TestAutoUpdate",
    "Update":{
        "UpdateDataSource" : "s3://cupsalphagafirmwarebin/station",
        "UpdateDataRole" : "arn:aws:iam::001234567890:role/SDK_Test_Role",
        "LoRaWAN" :{
            "CurrentVersion" :{
                "PackageVersion" : "1.0.0",
                "Station" : "2.0.5",
                "Model" : "linux"
            },
            "UpdateVersion" :{
                "PackageVersion" : "1.0.1",
                "Station" : "2.0.5",
                "Model" : "minihub"
            }
        }
```

```
    }
}
```

輸出：

```
{
    "Id": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateWirelessGatewayTaskDefinition。

create-wireless-gateway-task

以下程式碼範例顯示如何使用 create-wireless-gateway-task。

AWS CLI

建立無線閘道的任務

下列create-wireless-gateway-task範例會建立無線閘道的任務。

```
aws iotwireless create-wireless-gateway-task \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --wireless-gateway-task-definition-id "aa000102-0304-b0cd-ef56-a1b23cde456a"
```

輸出：

```
{
    "WirelessGatewayTaskDefinitionId": "aa204003-0604-30fb-ac82-a4f95aaf450a",
    "Status": "Success"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateWirelessGatewayTask。

create-wireless-gateway

以下程式碼範例顯示如何使用 create-wireless-gateway。

AWS CLI

建立無線閘道

下列create-wireless-gateway範例會建立無線 LoRaWAN 裝置閘道。

```
aws iotwireless create-wireless-gateway \
    --lorawan GatewayEui="a1b2c3d4567890ab",RfRegion="US915" \
    --name "myFirstLoRaWANGateway" \
    --description "Using my first LoRaWAN gateway"
```

輸出：

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateWirelessGateway。

delete-destination

以下程式碼範例顯示如何使用 delete-destination。

AWS CLI

刪除 IoT 無線目的地

下列delete-destination範例會使用IoTWirelessDestination您建立的名稱刪除無線目的地資源。

```
aws iotwireless delete-destination \
    --name "IoTWirelessDestination"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將目的地新增至 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDestination。

delete-device-profile

以下程式碼範例顯示如何使用 delete-device-profile。

AWS CLI

刪除裝置設定檔

下列delete-device-profile範例會刪除具有您建立之指定 ID 的裝置設定檔。

```
aws iotwireless delete-device-profile \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將設定檔新增至 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDeviceProfile。

delete-service-profile

以下程式碼範例顯示如何使用 delete-service-profile。

AWS CLI

刪除服務設定檔

下列delete-service-profile範例會刪除具有您建立之指定 ID 的服務設定檔。

```
aws iotwireless delete-service-profile \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將設定檔新增至 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteServiceProfile。

delete-wireless-device

以下程式碼範例顯示如何使用 delete-wireless-device。

AWS CLI

刪除無線裝置

下列delete-wireless-device範例會刪除具有指定 ID 的無線裝置。

```
aws iotwireless delete-wireless-device \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連線至 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteWirelessDevice。

delete-wireless-gateway-task-definition

以下程式碼範例顯示如何使用 delete-wireless-gateway-task-definition。

AWS CLI

刪除無線閘道任務定義

下列delete-wireless-gateway-task-definition範例會刪除您使用下列 ID 建立的無線閘道任務定義。

```
aws iotwireless delete-wireless-gateway-task-definition \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteWirelessGatewayTaskDefinition。

delete-wireless-gateway-task

以下程式碼範例顯示如何使用 delete-wireless-gateway-task。

AWS CLI

刪除無線閘道任務

下列delete-wireless-gateway-task範例會刪除具有指定 ID 的無線閘道任務。

```
aws iotwireless delete-wireless-gateway-task \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。
AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteWirelessGatewayTask。

delete-wireless-gateway

以下程式碼範例顯示如何使用 delete-wireless-gateway。

AWS CLI

刪除無線閘道

下列delete-wireless-gateway範例會刪除具有指定 ID 的無線閘道。

```
aws iotwireless delete-wireless-gateway \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。
AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteWirelessGateway。

`disassociate-aws-account-from-partner-account`

以下程式碼範例顯示如何使用 `disassociate-aws-account-from-partner-account`。

AWS CLI

取消合作夥伴帳戶與 AWS 帳戶的關聯

下列`disassociate-aws-account-from-partner-account`範例會取消合作夥伴帳戶與您目前關聯 AWS 帳戶的關聯。

```
aws iotwireless disassociate-aws-account-from-partner-account \
    --partner-account-id "12345678901234" \
    --partner-type "Sidewalk"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將閘道和無線裝置新增至 IoT Core for LoRaWAN。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateAwsAccountFromPartnerAccount。

`disassociate-wireless-device-from-thing`

以下程式碼範例顯示如何使用 `disassociate-wireless-device-from-thing`。

AWS CLI

取消物件與無線裝置的關聯

下列`disassociate-wireless-device-from-thing`範例會取消無線裝置與其目前關聯物件的關聯。

```
aws iotwireless disassociate-wireless-device-from-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將閘道和無線裝置新增至 IoT Core for LoRaWAN。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateWirelessDeviceFromThing。

disassociate-wireless-gateway-from-certificate

以下程式碼範例顯示如何使用 disassociate-wireless-gateway-from-certificate。

AWS CLI

取消憑證與無線閘道的關聯

下列會disassociate-wireless-gateway-from-certificate取消無線閘道與其目前相關聯憑證的關聯。

```
aws iotwireless disassociate-wireless-gateway-from-certificate \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將閘道和無線裝置新增至 IoT Core for LoRaWAN。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateWirelessGatewayFromCertificate。

disassociate-wireless-gateway-from-thing

以下程式碼範例顯示如何使用 disassociate-wireless-gateway-from-thing。

AWS CLI

取消物件與無線閘道的關聯

下列disassociate-wireless-gateway-from-thing範例會取消無線閘道與其目前關聯物件的關聯。

```
aws iotwireless disassociate-wireless-gateway-from-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將閘道和無線裝置新增至 IoT Core for LoRaWAN。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateWirelessGatewayFromThing。

get-destination

以下程式碼範例顯示如何使用 get-destination。

AWS CLI

取得 IoT 無線目的地的相關資訊

下列get-destination範例會使用IoTWirelessDestination您建立的名稱取得目的地資源的相關資訊。

```
aws iotwireless get-destination \
    --name "IoTWirelessDestination"
```

輸出：

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination",
    "Name": "IoTWirelessDestination",
    "Expression": "IoTWirelessRule",
    "ExpressionType": "RuleName",
    "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將目的地新增至 IoT Core for LoRaWAN。 AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDestination。

get-device-profile

以下程式碼範例顯示如何使用 get-device-profile。

AWS CLI

取得裝置設定檔的相關資訊

下列get-device-profile範例會使用您建立的指定 ID，取得裝置設定檔的相關資訊。

```
aws iotwireless get-device-profile \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

輸出：

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
    "LoRaWAN": {
    "MacVersion": "1.0.3",
    "MaxDutyCycle": 10,
    "Supports32BitFCnt": false,
    "RegParamsRevision": "RP002-1.0.1",
    "SupportsJoin": true,
    "RfRegion": "US915",
    "MaxEirp": 13,
    "SupportsClassB": false,
    "SupportsClassC": false
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將設定檔新增至 IoT Core for LoRaWAN。
AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDeviceProfile。

get-partner-account

以下程式碼範例顯示如何使用 get-partner-account。

AWS CLI

取得合作夥伴帳戶資訊

下列get-partner-account範例會取得具有下列 ID 之 Sidewalk 帳戶的相關資訊。

```
aws iotwireless get-partner-account \
    --partner-account-id "12345678901234" \
    --partner-type "Sidewalk"
```

輸出：

```
{
    "Sidewalk": {
        "AmazonId": "12345678901234",
        "Fingerprint":
 "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
    },
    "AccountLinked": false
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Amazon Sidewalk Integration for AWS IoT Core。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPartnerAccount。

get-service-endpoint

以下程式碼範例顯示如何使用 get-service-endpoint。

AWS CLI

取得服務端點

下列get-service-endpoint範例會取得 CUPS 通訊協定的帳戶特定端點。

```
aws iotwireless get-service-endpoint
```

輸出：

```
{
    "ServiceType": "CUPS",
    "ServiceEndpoint": "https://A1RMKZ37ACAGOT.cups.lorawan.us-
east-1.amazonaws.com:443",
    "ServerTrust": "-----BEGIN CERTIFICATE-----\n
    MIIESTCCAzGgAwIBAgITBn+UV4WH6Kx33rJTMlu8mYtWDTANBgkqhkiG9w0BAQsF\n
```

```
    ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6\n
    b24gUm9vdCBDQSAxMB4XDTE1MTAyMjAwMDAwMFoXDTI1MTAxOTAwMDAwMFowRjEL\n
    MAkGA1UEBhMCVVMxDzANBgNVBAoTBkFtYXpvbjEVMBMGA1UECxMMU2VydmVyIENB\n
    IDFCMQ8wDQYDVQQDEwZBbWF6b24wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK\n
    AoIBAQDCThZn3c68asg3Wuw6MLAd5tES6BIoSMzoKcG5blPVo+sDORrMd4f2AbnZ\n
    cMzPa43j4wNxhplty6aUKk4T1qe9BOwKFjwK6zmxxLVYo7bHViXsPlJ6qOMpFge5\n
    blDP+18x+B26A0piiQOuPkfyDyeR4xQghfj66Yo19V+emU3nazfvpFA+ROz6WoVm\n
    B5x+F2pV8xeKNR7u6azDdU5YVX1TawprmxRC1+WsAYmz6qP+z8ArDITC2FMVy2fw\n
    0IjKOtEXc/VfmtTFch5+AfGYMGMqqvJ6LcXiAhqG5TI+Dr0RtM88k+8XUBCeQ8IG\n
    KuANaL7TiItKZYxK1MMuTJtV9IblAgMBAAGjggE7MIIBNzASBgNVHRMBAf8ECDAG\n
    AQH/AgEAMA4GA1UdDwEB/wQEAwIBhjAdBgNVHQ4EFgQUWaRmBlKge5WSPKOUByeW\n
    dFv5PdAwHwYDVR0jBBgwFoAUhBjMhTTsvAyUlC4IWZzHshBOCggwewYIKwYBBQUH\n
    AQEEbzBtMC8GCCsGAQUFBzABhiNodHRwOi8vb2NzcC5yb290Y2ExLmFtYXpvbnRy\n
    dXN0LmNvbTA6BggrBgEFBQcwAoYuaHR0cDovL2NydC5yb290Y2ExLmFtYXpvbnRy\n
    dXN0LmNvbS9yb290Y2ExLmNlcjA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3Js\n
    LnJvb3RjYTEuYW1hem9udHJ1c3QuY29tL3Jvb3RjYTEuY3JsMBMGA1UdIAQMMAow\n
    CAYGZ4EMAQIBMA0GCSqGSIb3DQEBCwUAA4IBAQCFkr41u3nPo4FCHOTjY3NTOVI1\n
    59Gt/a6ZiqyJEi+752+a1U5y6iAwYfmXss2lJwJFqMp2PphKg5625kXg8kP2CN5t\n
    6G7bMQcT8C8xDZNtYTd7WPD8UZiRKAJPBXa30/AbwuZe0GaFEQ8ugcYQgSn+IGBI\n
    8/LwhBNTZTUVEWuCUUBVV18YtbAiPq3yXqMB48Oz+ctBWuZSkbvkNodPLamkB2g1\n
    upRyzQ7qDn1X8nn8N8V7YJ6y68AtkHcNSRAnpTitxBKjtKPISLMVCx7i4hncxHZS\n
    yLyKQXhw2W2Xs0qLeC1etA+jTGDK4UfLeC0SF7FSi8o5LL21L8IzApar2pR/\n
    -----END CERTIFICATE-----\n"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetServiceEndpoint。

get-service-profile

以下程式碼範例顯示如何使用 get-service-profile。

AWS CLI

取得服務設定檔的相關資訊

下列get-service-profile範例會使用您建立的指定 ID，取得服務設定檔的相關資訊。

```
aws iotwireless get-service-profile \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

輸出：

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:651419225604:ServiceProfile/538185bb-
d7e7-4b95-96a0-c51aa4a5b9a0",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
    "LoRaWAN": {
        "HrAllowed": false,
        "NwkGeoLoc": false,
        "DrMax": 15,
        "UlBucketSize": 4096,
        "PrAllowed": false,
        "ReportDevStatusBattery": false,
        "DrMin": 0,
        "DlRate": 60,
        "AddGwMetadata": false,
        "ReportDevStatusMargin": false,
        "MinGwDiversity": 1,
        "RaAllowed": false,
        "DlBucketSize": 4096,
        "DevStatusReqFreq": 24,
        "TargetPer": 5,
        "UlRate": 60
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將設定檔新增至 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetServiceProfile。

get-wireless-device-statistics

以下程式碼範例顯示如何使用 get-wireless-device-statistics。

AWS CLI

取得無線裝置的操作資訊

下列get-wireless-device-statistics範例會取得無線裝置的操作資訊。

```
aws iotwireless get-wireless-device-statistics \
    --wireless-device-id "1ffd32c8-8130-4194-96df-622f072a315f"
```

輸出：

```
{
    "WirelessDeviceId": "1ffd32c8-8130-4194-96df-622f072a315f"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetWirelessDeviceStatistics。

get-wireless-device

以下程式碼範例顯示如何使用 get-wireless-device。

AWS CLI

取得無線裝置的相關資訊

下列get-wireless-device範例列出您 AWS 帳戶中可用的小工具。

```
aws iotwireless get-wireless-device \
    --identifier "1ffd32c8-8130-4194-96df-622f072a315f" \
    --identifier-type WirelessDeviceID
```

輸出：

```
{
    "Name": "myLoRaWANDevice",
    "ThingArn": "arn:aws:iot:us-east-1:123456789012:thing/44b87eb4-9bce-423d-
b5fc-973f5ecc358b",
    "DestinationName": "IoTWirelessDestination",
    "Id": "1ffd32c8-8130-4194-96df-622f072a315f",
    "ThingName": "44b87eb4-9bce-423d-b5fc-973f5ecc358b",
    "Type": "LoRaWAN",
    "LoRaWAN": {
        "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",
        "ServiceProfileId": "fe98dc76-cd12-001e-2d34-5550432da100",
        "OtaaV1_1": {
            "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",
            "JoinEui": "b4c231a359bc2e3d",
            "NwkKey": "01c3f004a2d6efffe32c4eda14bcd2b4"
```

```
        },
        "DevEui": "ac12efc654d23fc2"
    },
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f",
    "Description": "My LoRaWAN wireless device"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連線至 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetWirelessDevice。

get-wireless-gateway-certificate

以下程式碼範例顯示如何使用 get-wireless-gateway-certificate。

AWS CLI

取得與無線閘道相關聯的憑證 ID

下列get-wireless-gateway-certificate範例會取得與具有指定 ID 之無線閘道相關聯的憑證 ID。

```
aws iotwireless get-wireless-gateway-certificate \
    --id "6c44ab31-8b4d-407a-bed3-19b6c7cda551"
```

輸出：

```
{
    "IotCertificateId":
 "8ea4aeae3db34c78cce75d9abd830356869ead6972997e0603e5fd032c804b6f"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetWirelessGatewayCertificate。

get-wireless-gateway-firmware-information

以下程式碼範例顯示如何使用 get-wireless-gateway-firmware-information。

AWS CLI

取得無線閘道的韌體資訊

下列get-wireless-gateway-firmware-information範例會取得韌體版本和無線閘道的其他資訊。

```
aws iotwireless get-wireless-gateway-firmware-information \
    --id "3039b406-5cc9-4307-925b-9948c63da25b"
```

輸出：

```
{
    "LoRaWAN" :{
        "CurrentVersion" :{
            "PackageVersion" : "1.0.0",
            "Station" : "2.0.5",
            "Model" : "linux"
        }
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetWirelessGatewayFirmwareInformation。

get-wireless-gateway-statistics

以下程式碼範例顯示如何使用 get-wireless-gateway-statistics。

AWS CLI

取得無線閘道的操作資訊

下列get-wireless-gateway-statistics範例會取得無線閘道的操作資訊。

```
aws iotwireless get-wireless-gateway-statistics \
    --wireless-gateway-id "3039b406-5cc9-4307-925b-9948c63da25b"
```

輸出：

```
{
    "WirelessGatewayId": "3039b406-5cc9-4307-925b-9948c63da25b"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetWirelessGatewayStatistics。

get-wireless-gateway-task-definition

以下程式碼範例顯示如何使用 get-wireless-gateway-task-definition。

AWS CLI

取得無線閘道任務定義的相關資訊

下列get-wireless-gateway-task-definition範例會取得具有指定 ID 之無線任務定義的相關資訊。

```
aws iotwireless get-wireless-gateway-task-definition \
    --id "b7d3baad-25c7-35e7-a4e1-1683a0d61da9"
```

輸出：

```
{
    "AutoCreateTasks": true,
    "Name": "TestAutoUpdate",
    "Update":{
        "UpdateDataSource" : "s3://cupsalphagafirmwarebin/station",
        "UpdateDataRole" : "arn:aws:iam::001234567890:role/SDK_Test_Role",
        "LoRaWAN" :{
            "CurrentVersion" :{
                "PackageVersion" : "1.0.0",
                "Station" : "2.0.5",
                "Model" : "linux"
            },
            "UpdateVersion" :{
                "PackageVersion" : "1.0.1",
                "Station" : "2.0.5",
                "Model" : "minihub"
            }
```

```
        }
    }
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetWirelessGatewayTaskDefinition。

get-wireless-gateway-task

以下程式碼範例顯示如何使用 get-wireless-gateway-task。

AWS CLI

取得無線閘道任務的相關資訊

下列get-wireless-gateway-task範例會取得具有指定 ID 之無線閘道任務的相關資訊。

```
aws iotwireless get-wireless-gateway-task \
    --id "11693a46-6866-47c3-a031-c9a616e7644b"
```

輸出：

```
{
    "WirelessGatewayId": "6c44ab31-8b4d-407a-bed3-19b6c7cda551",
    "WirelessGatewayTaskDefinitionId": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9",
    "Status": "Success"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetWirelessGatewayTask。

get-wireless-gateway

以下程式碼範例顯示如何使用 get-wireless-gateway。

AWS CLI

取得無線閘道的相關資訊

下列get-wireless-gateway範例取得無線閘道 的相關資訊myFirstLoRaWANGateway。

```
aws iotwireless get-wireless-gateway \
    --identifier "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --identifier-type WirelessGatewayId
```

輸出：

```
{
    "Description": "My first LoRaWAN gateway",
    "ThingArn": "arn:aws:iot:us-east-1:123456789012:thing/a1b2c3d4-5678-90ab-
cdef-12ab345c67de",
    "LoRaWAN": {
        "RfRegion": "US915",
        "GatewayEui": "a1b2c3d4567890ab"
    },
    "ThingName": "a1b2c3d4-5678-90ab-cdef-12ab345c67de",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessGateway/6c44ab31-8b4d-407a-bed3-19b6c7cda551",
    "Name": "myFirstLoRaWANGateway"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetWirelessGateway。

list-destinations

以下程式碼範例顯示如何使用 list-destinations。

AWS CLI

列出無線目的地

下列list-destinations範例列出註冊至您 AWS 帳戶的可用目的地。

```
aws iotwireless list-destinations
```

輸出：

```
{
    "DestinationList": [
        {
            "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination",
            "Name": "IoTWirelessDestination",
            "Expression": "IoTWirelessRule",
            "Description": "Destination for messages processed using
 IoTWirelessRule",
            "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
        },
        {
            "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination2",
            "Name": "IoTWirelessDestination2",
            "Expression": "IoTWirelessRule2",
            "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將目的地新增至 IoT Core for LoRaWAN。

AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDestinations。

`list-device-profiles`

以下程式碼範例顯示如何使用 `list-device-profiles`。

AWS CLI

列出裝置設定檔

下列`list-device-profiles`範例列出註冊至您 AWS 帳戶的可用裝置設定檔。

```
aws iotwireless list-device-profiles
```

輸出：

```
{
    "DeviceProfileList": [
```

```
        {
            "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
            "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:DeviceProfile/12345678-a1b2-3c45-67d8-e90fa1b2c34d"
        },
        {
            "Id": "a1b2c3d4-5678-90ab-cdef-12ab345c67de",
            "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/
a1b2c3d4-5678-90ab-cdef-12ab345c67de"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將設定檔新增至 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDeviceProfiles。

`list-partner-accounts`

以下程式碼範例顯示如何使用 `list-partner-accounts`。

AWS CLI

列出合作夥伴帳戶

下列`list-partner-accounts`範例列出與您的帳戶相關聯的可用合作夥伴 AWS 帳戶。

```
aws iotwireless list-partner-accounts
```

輸出：

```
{
    "Sidewalk": [
        {
            "AmazonId": "78965678771228",
            "Fingerprint":
 "bd96d8ef66dbfd2160eb60e156849e82ad7018b8b73c1ba0b4fc65c32498ee35"
        },
        {
            "AmazonId": "89656787651228",
            "Fingerprint":
 "bc5e99e151c07be14be7e6603e4489c53f858b271213a36ebe3370777ba06e9b"
```

```
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Amazon Sidewalk Integration for AWS IoT Core。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPartnerAccounts。

`list-service-profiles`

以下程式碼範例顯示如何使用 `list-service-profiles`。

AWS CLI

列出服務設定檔

下列`list-service-profiles`範例列出註冊至您 AWS 帳戶的可用服務設定檔。

```
aws iotwireless list-service-profiles
```

輸出：

```
{
    "ServiceProfileList": [
        {
            "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
            "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:ServiceProfile/538185bb-d7e7-4b95-96a0-c51aa4a5b9a0"
        },
        {
            "Id": "a1b2c3d4-5678-90ab-cdef-12ab345c67de",
            "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/
ea8bc823-5d13-472e-8d26-9550737d8100"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將設定檔新增至 IoT Core for LoRaWAN。
AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListServiceProfiles。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出指派給資源的標籤

下列list-tags-for-resource範例列出指派給無線目的地資源的標籤。

```
aws iotwireless list-tags-for-resource \
    --resource-arn "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination"
```

輸出：

```
{
    "Tags": [
        {
            "Value": "MyValue",
            "Key": "MyTag"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的描述您的 IoT Core for LoRaWAN 資源。 AWS IoT

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

list-wireless-devices

以下程式碼範例顯示如何使用 list-wireless-devices。

AWS CLI

列出可用的無線裝置

下列list-wireless-devices範例列出向 AWS 您的帳戶註冊的可用無線裝置。

```
aws iotwireless list-wireless-devices
```

輸出：

```
{
    "WirelessDeviceList": [
        {
            "Name": "myLoRaWANDevice",
            "DestinationName": "IoTWirelessDestination",
            "Id": "1ffd32c8-8130-4194-96df-622f072a315f",
            "Type": "LoRaWAN",
            "LoRaWAN": {
                "DevEui": "ac12efc654d23fc2"
            },
            "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListWirelessDevices。

list-wireless-gateway-task-definitions

以下程式碼範例顯示如何使用 list-wireless-gateway-task-definitions。

AWS CLI

列出無線閘道任務定義

下列list-wireless-gateway-task-definitions範例列出註冊至您 AWS 帳戶的可用無線閘道任務定義。

```
aws iotwireless list-wireless-gateway-task-definitions
```

輸出：

```
{
    "TaskDefinitions": [
        {
            "Id": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9",
            "LoRaWAN" :
                {
```

```
                "CurrentVersion" :{
                    "PackageVersion" : "1.0.0",
                    "Station" : "2.0.5",
                    "Model" : "linux"
                },
                "UpdateVersion" :{
                    "PackageVersion" : "1.0.1",
                    "Station" : "2.0.5",
                    "Model" : "minihub"
                }
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連線至 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListWirelessGatewayTaskDefinitions。

`list-wireless-gateways`

以下程式碼範例顯示如何使用 `list-wireless-gateways`。

AWS CLI

列出無線閘道

下列`list-wireless-gateways`範例列出您 AWS 帳戶中可用的無線閘道。

```
aws iotwireless list-wireless-gateways
```

輸出：

```
{
    "WirelessGatewayList": [
        {
            "Description": "My first LoRaWAN gateway",
            "LoRaWAN": {
                "RfRegion": "US915",
                "GatewayEui": "dac632ebc01d23e4"
            },
            "Id": "3039b406-5cc9-4307-925b-9948c63da25b",
```

```
            "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessGateway/3039b406-5cc9-4307-925b-9948c63da25b",
            "Name": "myFirstLoRaWANGateway"
        },
        {
            "Description": "My second LoRaWAN gateway",
            "LoRaWAN": {
                "RfRegion": "US915",
                "GatewayEui": "cda123fffe92ecd2"
            },
            "Id": "3285bdc7-5a12-4991-84ed-dadca65e342e",
            "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessGateway/3285bdc7-5a12-4991-84ed-dadca65e342e",
            "Name": "mySecondLoRaWANGateway"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListWirelessGateways。

send-data-to-wireless-device

以下程式碼範例顯示如何使用 send-data-to-wireless-device。

AWS CLI

將資料傳送至無線裝置

下列send-data-to-wireless-device範例會將解密的應用程式資料框架傳送至無線裝置。

```
aws iotwireless send-data-to-wireless-device \
    --id "11aa5eae-2f56-4b8e-a023-b28d98494e49" \
    --transmit-mode "1" \
    --payload-data "SGVsbG8gVG8gRGV2c2lt" \
    --wireless-metadata LoRaWAN={FPort=1}
```

輸出：

```
{
    MessageId: "6011dd36-0043d6eb-0072-0008"
```

```
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SendDataToWirelessDevice。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

指定資源的標籤索引鍵和值

下列tag-resource範例使用IoTWirelessDestination索引鍵 MyTag和值 標記無線目的地MyValue。

```
aws iotwireless tag-resource \
    --resource-arn "arn:aws:iotwireless:us-east-1:651419225604:Destination/
IoTWirelessDestination" \
    --tags Key="MyTag",Value="MyValue"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的描述您的 IoT Core for LoRaWAN 資源。 AWS IoT

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

test-wireless-device

以下程式碼範例顯示如何使用 test-wireless-device。

AWS CLI

測試無線裝置

下列test-wireless-device範例會將 的上行資料Hello傳送至具有指定 ID 的裝置。

```
aws iotwireless test-wireless-device \
    --id "11aa5eae-2f56-4b8e-a023-b28d98494e49"
```

輸出：

```
{
    Result: "Test succeeded. one message is sent with payload: hello"
}
```

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TestWirelessDevice。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源移除一或多個標籤

下列untag-resource範例會從無線目的地 移除標籤MyTag及其值IoTWirelessDestination。

```
aws iotwireless untag-resource \
    --resource-arn "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination" \
    --tag-keys "MyTag"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的描述您的 IoT Core for LoRaWAN 資源。 AWS IoT

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-destination

以下程式碼範例顯示如何使用 update-destination。

AWS CLI

更新目的地的屬性

下列update-destination範例會更新無線目的地的描述屬性。

```
aws iotwireless update-destination \
    --name "IoTWirelessDestination" \
    --description "Destination for messages processed using IoTWirelessRule"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將目的地新增至 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDestination。

update-partner-account

以下程式碼範例顯示如何使用 update-partner-account。

AWS CLI

更新合作夥伴帳戶的屬性

以下內容AppServerPrivateKey會更新具有指定 ID 之帳戶的 update-partner-account 。

```
aws iotwireless update-partner-account \
    --partner-account-id "78965678771228" \
    --partner-type "Sidewalk" \
    --sidewalk
 AppServerPrivateKey="f798ab4899346a88599180fee9e14fa1ada7b6df989425b7c6d2146dd6c815bb"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 Amazon Sidewalk Integration for AWS IoT Core。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePartnerAccount。

update-wireless-device

以下程式碼範例顯示如何使用 update-wireless-device。

AWS CLI

更新無線裝置的屬性

下列update-wireless-device範例會更新已註冊至您 AWS 帳戶的無線裝置的屬性。

```
aws iotwireless update-wireless-device \
    --id "1ffd32c8-8130-4194-96df-622f072a315f" \
    --destination-name IoTWirelessDestination2 \
    --description "Using my first LoRaWAN device"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateWirelessDevice。

update-wireless-gateway

以下程式碼範例顯示如何使用 update-wireless-gateway。

AWS CLI

更新無線閘道

下列update-wireless-gateway範例會更新無線閘道的描述。

```
aws iotwireless update-wireless-gateway \
    --id "3285bdc7-5a12-4991-84ed-dadca65e342e" \
    --description "Using my LoRaWAN gateway"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS IoT 開發人員指南中的將裝置和閘道連接到 IoT Core for LoRaWAN。AWS IoT

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateWirelessGateway。

使用 的 Amazon IVS 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon IVS 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

batch-get-channel

以下程式碼範例顯示如何使用 batch-get-channel。

AWS CLI

取得多個頻道的頻道組態資訊

下列batch-get-channel範例列出指定頻道的相關資訊。

```
aws ivs batch-get-channel \
    --arns arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
        arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl
```

輸出：

```
{
    "channels": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
            "authorized": false,
            "containerFormat": "TS",
            "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
            "insecureIngest": false,
            "latencyMode": "LOW",
            "multitrackInputConfiguration": {
                "enabled": false,
                "maximumResolution": "FULL_HD",
                "policy": "ALLOW"
            },
            "name": "channel-1",
            "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/
api/video/v1/us-west-2.123456789012.channel-1.abcdEFGH.m3u8",
            "preset": "",
```

```
            "playbackRestrictionPolicyArn": "",
            "recordingConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABCD12cdEFgh",
            "srt": {
                "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
                "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
            },
            "tags": {},
            "type": "STANDARD"
        },
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl",
            "authorized": false,
            "containerFormat": "FRAGMENTED_MP4",
            "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
            "insecureIngest": false,
            "latencyMode": "LOW",
            "multitrackInputConfiguration": {
                "enabled": true,
                "maximumResolution": "FULL_HD",
                "policy": "ALLOW"
            },
            "name": "channel-2",
            "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/
api/video/v1/us-west-2.123456789012.channel-2.abcdEFGH.m3u8",
            "preset": "",
            "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ"",
            "recordingConfigurationArn": "",
            "srt": {
                "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
                "passphrase":
 "BA1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
            },
            "tags": {},
            "type": "STANDARD"
        }
    ]
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的建立頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetChannel。

batch-get-stream-key

以下程式碼範例顯示如何使用 batch-get-stream-key。

AWS CLI

取得多個串流金鑰的相關資訊

下列batch-get-stream-key範例取得指定串流金鑰的相關資訊。

```
aws ivs batch-get-stream-key \
    --arns arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh \
       arn:aws:ivs:us-west-2:123456789012:stream-key/skSKIJKLmnop
```

輸出：

```
{
    "streamKeys": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh",
            "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
            "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
            "tags": {}
        },
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/skSKIJKLmnop",
            "value": "sk_us-west-2_abcdABCDefgh_567890ghijkl",
            "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
            "tags": {}
        }
     ]
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的建立頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetStreamKey。

batch-start-viewer-session-revocation

以下程式碼範例顯示如何使用 batch-start-viewer-session-revocation。

AWS CLI

撤銷多個頻道 ARN 和檢視器 ID 對的檢視器工作階段

下列batch-start-viewer-session-revocation範例會同時對多個頻道 ARN 和檢視器 ID 對執行工作階段撤銷。如果發起人沒有撤銷指定工作階段的許可，請求可能會正常完成，但傳回錯誤欄位中的值。

```
aws ivs batch-start-viewer-session-revocation \
    --viewer-sessions '[{"channelArn":"arn:aws:ivs:us-west-2:123456789012:channel/
abcdABCDefgh1","viewerId":"abcdefg1","viewerSessionVersionsLessThanOrEqualTo":1234567890},
 \
      {"channelArn":"arn:aws:ivs:us-west-2:123456789012:channel/
abcdABCDefgh2","viewerId":"abcdefg2","viewerSessionVersionsLessThanOrEqualTo":1234567890}]'
```

輸出：

```
{
    "errors": [
        {
            "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/
abcdABCDefgh1",
            "viewerId": "abcdefg1",
            "code": "403",
            "message": "not authorized",
        },
        {
            "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/
abcdABCDefgh2",
            "viewerId": "abcdefg2",
            "code": "403",
            "message": "not authorized",
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon 互動式影片服務使用者指南》中的設定私有頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchStartViewerSessionRevocation。

create-channel

以下程式碼範例顯示如何使用 create-channel。

AWS CLI

範例 1：建立沒有錄製的頻道

下列create-channel範例會建立新的頻道和相關聯的串流金鑰，以開始串流。

```
aws ivs create-channel \
    --name 'test-channel' \
    --no-insecure-ingest
```

輸出：

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "authorized": false,
        "containerFormat": "TS",
        "name": "test-channel",
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "tags": {},
        "type": "STANDARD"
    },
    "streamKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/g1H2I3j4k5L6",
        "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
```

```
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《IVS 低延遲使用者指南》中的建立頻道。

範例 2：若要建立已啟用錄製的頻道，請使用其 ARN 指定的 RecordingConfiguration 資源

下列create-channel範例會建立新的頻道和相關聯的串流金鑰以開始串流，並設定頻道的錄製：

```
aws ivs create-channel \
    --name test-channel-with-recording \
    --insecure-ingest \
    --recording-configuration-arn 'arn:aws:ivs:us-west-2:123456789012:recording-
configuration/ABCD12cdEFgh'
```

輸出：

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "containerFormat": "TS",
        "name": "test-channel-with-recording",
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:recording-
configuration/ABCD12cdEFgh",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "BA1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": true,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
```

```
        "authorized": false,
        "tags": {},
        "type": "STANDARD"
    },
    "streamKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",
        "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的記錄到 Amazon S3。

範例 3：使用 ARN 指定的播放限制政策建立頻道

下列create-channel範例會建立新的頻道和相關聯的串流金鑰以開始串流，並設定頻道的播放限制政策：

```
aws ivs create-channel \
    --name test-channel-with-playback-restriction-policy\
    --insecure-ingest \
    --playback-restriction-policy-arn 'arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ'
```

輸出：

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "containerFormat": "TS",
        "name": "test-channel-with-playback-restriction-policy",
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ",
        "recordingConfigurationArn": "",
        "srt": {
```

```
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2edfGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": true,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {},
        "type": "STANDARD"
    },
    "streamKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",
        "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的不需要的內容和檢視器。

範例 4：建立啟用多軌的頻道

下列create-channel範例會建立新的頻道和相關聯的串流金鑰以開始串流，並啟用多軌。

```
aws ivs create-channel \
    --name 'test-channel' \
    --no-insecure-ingest \
    --container-format 'FRAGMENTED_MP4' \
    --multitrack-input-configuration '{"enabled": true,"maximumResolution":
 "FULL_HD","policy": "ALLOW"}'
```

輸出：

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "authorized": false,
        "containerFormat": "FRAGMENTED_MP4",
        "name": "test-channel",
        "latencyMode": "LOW",
```

```
        "multitrackInputConfiguration": {
            "enabled": true,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "tags": {},
        "type": "STANDARD"
    },
    "streamKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/g1H2I3j4k5L6",
        "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的建立頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateChannel。

create-playback-restriction-policy

以下程式碼範例顯示如何使用 create-playback-restriction-policy。

AWS CLI

建立播放限制政策

下列create-playback-restriction-policy範例會建立新的播放回射政策。

```
aws ivs create-playback-restriction-policy \
```

```
    --name "test-playback-restriction-policy" \
    --enable-strict-origin-enforcement \
    --tags "key1=value1, key2=value2" \
    --allowed-countries US MX \
    --allowed-origins https://www.website1.com https://www.website2.com
```

輸出：

```
{
    "playbackRestrictionPolicy": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ",
        "allowedCountries": [
            "US",
            "MX"
        ],
        "allowedOrigins": [
            "https://www.website1.com",
            "https://www.website2.com"
        ],
        "enableStrictOriginEnforcement": true,
        "name": "test-playback-restriction-policy",
        "tags": {
            "key1": "value1",
            "key2": "value2"
        }
    }
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的不需要的內容和檢視器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePlaybackRestrictionPolicy。

`create-recording-configuration`

以下程式碼範例顯示如何使用 `create-recording-configuration`。

AWS CLI

建立 RecordingConfiguration 資源

下列`create-recording-configuration`範例會建立 RecordingConfiguration 資源，以啟用錄製至 Amazon S3 的功能。

```
aws ivs create-recording-configuration \
    --name "test-recording-config" \
    --recording-reconnect-window-seconds 60 \
    --tags "key1=value1, key2=value2" \
    --rendition-configuration renditionSelection="CUSTOM",renditions="HD" \
    --thumbnail-configuration
 recordingMode="INTERVAL",targetIntervalSeconds=1,storage="LATEST",resolution="LOWEST_RESOLU
 \
    --destination-configuration s3={bucketName=demo-recording-bucket}
```

輸出：

```
{
    "recordingConfiguration": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/
ABcdef34ghIJ",
        "name": "test-recording-config",
        "destinationConfiguration": {
            "s3": {
                "bucketName": "demo-recording-bucket"
            }
        },
        "state": "CREATING",
        "tags": {
            "key1": "value1",
            "key2": "value2"
        },
        "thumbnailConfiguration": {
            "recordingMode": "INTERVAL",
            "targetIntervalSeconds": 1,
            "resolution": "LOWEST_RESOLUTION",
            "storage": [
                "LATEST"
            ]
        },
        "recordingReconnectWindowSeconds": 60,
        "renditionConfiguration": {
            "renditionSelection": "CUSTOM",
            "renditions": [
                "HD"
            ]
        }
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的錄製至 Amazon S3。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRecordingConfiguration。

create-stream-key

以下程式碼範例顯示如何使用 create-stream-key。

AWS CLI

建立串流金鑰

下列create-stream-key範例會為指定的 ARN (Amazon Resource Name) 建立串流金鑰。

```
aws ivs create-stream-key \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

輸出：

```
{
    "streamKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",
        "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《IVS 低延遲使用者指南》中的建立頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateStreamKey。

delete-channel

以下程式碼範例顯示如何使用 delete-channel。

AWS CLI

刪除頻道及其相關聯的串流金鑰

下列delete-channel範例會刪除具有指定 ARN (Amazon Resource Name) 的頻道。

```
aws ivs delete-channel \
    --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

此命令不會產生輸出。

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的建立頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteChannel。

delete-playback-key-pair

以下程式碼範例顯示如何使用 delete-playback-key-pair。

AWS CLI

刪除指定的播放金鑰對

下列delete-playback-key-pair範例會傳回指定金鑰對的指紋。

```
aws ivs delete-playback-key-pair \
    --arn arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon 互動式影片服務使用者指南》中的設定私有頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePlaybackKeyPair。

delete-playback-restriction-policy

以下程式碼範例顯示如何使用 delete-playback-restriction-policy。

AWS CLI

刪除播放限制政策

下列delete-playback-restriction-policy範例會刪除具有指定政策 ARN (Amazon Resource Name) 的播放撤銷政策。

```
aws ivs delete-playback-restriction-policy \
```

```
    --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的不需要的內容和檢視器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePlaybackRestrictionPolicy。

delete-recording-configuration

以下程式碼範例顯示如何使用 delete-recording-configuration。

AWS CLI

刪除其 ARN 指定的 RecordingConfiguration 資源

下列delete-recording-configuration範例會刪除具有指定 ARN 的 RecordingConfiguration 資源。

```
aws ivs delete-recording-configuration \
    --arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABcdef34ghIJ"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的錄製至 Amazon S3。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRecordingConfiguration。

delete-stream-key

以下程式碼範例顯示如何使用 delete-stream-key。

AWS CLI

刪除串流金鑰

下列delete-stream-key範例會刪除指定 ARN (Amazon Resource Name) 的串流金鑰，使其無法再用於串流。

```
aws ivs delete-stream-key \
    --arn arn:aws:ivs:us-west-2:123456789012:stream-key/g1H2I3j4k5L6
```

此命令不會產生輸出。

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的建立頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteStreamKey。

`get-channel`

以下程式碼範例顯示如何使用 `get-channel`。

AWS CLI

取得頻道的組態資訊

下列`get-channel`範例取得指定頻道 ARN (Amazon Resource Name) 的頻道組態。

```
aws ivs get-channel \
    --arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh'
```

輸出：

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "authorized": false,
        "containerFormat": "TS",
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "name": "channel-1",
        "playbackRestrictionPolicyArn": "",
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
```

```
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "tags": {}
        "type": "STANDARD",
    }
}
```

如需詳細資訊，請參閱《IVS 低延遲使用者指南》中的建立頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetChannel。

get-playback-key-pair

以下程式碼範例顯示如何使用 get-playback-key-pair。

AWS CLI

若要取得指定的播放金鑰對

下列get-playback-key-pair範例會傳回指定金鑰對的指紋。

```
aws ivs get-playback-key-pair \
    --arn arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh
```

輸出：

```
{
    "keyPair": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh",
        "name": "my-playback-key",
        "fingerprint": "0a:1b:2c:ab:cd:ef:34:56:70:b1:b2:71:01:2a:a3:72",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon 互動式影片服務使用者指南》中的設定私有頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPlaybackKeyPair。

get-playback-restriction-policy

以下程式碼範例顯示如何使用 get-playback-restriction-policy。

AWS CLI

若要取得播放限制政策的組態資訊

下列get-playback-restriction-policy範例使用指定的政策 ARN (Amazon Resource Name) 取得播放還原政策組態。

```
aws ivs get-playback-restriction-policy \
    --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ"
```

輸出：

```
{
    "playbackRestrictionPolicy": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ",
        "allowedCountries": [
            "US",
            "MX"
        ],
        "allowedOrigins": [
            "https://www.website1.com",
            "https://www.website2.com"
        ],
        "enableStrictOriginEnforcement": true,
        "name": "test-playback-restriction-policy",
        "tags": {
            "key1": "value1",
            "key2": "value2"
        }
    }
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的不需要的內容和檢視器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPlaybackRestrictionPolicy。

get-recording-configuration

以下程式碼範例顯示如何使用 get-recording-configuration。

AWS CLI

取得 RecordingConfiguration 資源的相關資訊

下列get-recording-configuration範例取得指定 ARN 的 RecordingConfiguration 資源相關資訊。

```
aws ivs get-recording-configuration \
    --arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABcdef34ghIJ"
```

輸出：

```
{
    "recordingConfiguration": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/
ABcdef34ghIJ",
        "destinationConfiguration": {
            "s3": {
                "bucketName": "demo-recording-bucket"
            }
        },
        "name": "test-recording-config",
        "recordingReconnectWindowSeconds": 60,
        "state": "ACTIVE",
        "tags": {
            "key1" : "value1",
            "key2" : "value2"
        },
        "thumbnailConfiguration": {
            "recordingMode": "INTERVAL",
            "targetIntervalSeconds": 1,
            "resolution": "LOWEST_RESOLUTION",
            "storage": [
                "LATEST"
            ]
        },
        "renditionConfiguration": {
            "renditionSelection": "CUSTOM",
            "renditions": [
                "HD"
            ]
        }
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的錄製至 Amazon S3。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRecordingConfiguration。

get-stream-key

以下程式碼範例顯示如何使用 get-stream-key。

AWS CLI

取得串流的相關資訊

下列get-stream-key範例取得指定串流金鑰的相關資訊。

```
aws ivs get-stream-key \
    --arn arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh --region=us-
west-2
```

輸出：

```
{
    "streamKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh",
        "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《IVS 低延遲使用者指南》中的建立頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetStreamKey。

get-stream-session

以下程式碼範例顯示如何使用 get-stream-session。

AWS CLI

取得指定串流的中繼資料

下列get-stream-session範例會取得指定頻道 ARN (Amazon Resource Name) 和指定串流的中繼資料組態；如果streamId未提供，則會選取頻道的最新串流。

```
aws ivs get-stream-session \
    --channel-arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh' \
    --stream-id 'mystream'
```

輸出：

```
{
    "streamSession": {
        "streamId": "mystream1",
        "startTime": "2023-06-26T19:09:28+00:00",
        "channel": {
            "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
            "name": "mychannel",
            "latencyMode": "LOW",
            "type": "STANDARD",
            "recordingConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABcdef34ghIJ",
            "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
            "playbackUrl": "url-string",
            "authorized": false,
            "insecureIngest": false,
            "preset": ""
        },
        "ingestConfiguration": {
            "audio": {
                "channels": 2,
                "codec": "mp4a.40.2",
                "sampleRate": 8000,
                "targetBitrate": 46875,
                "track": "Track0"
            },
            "video": {
                "avcProfile": "Baseline",
                "avcLevel": "4.2",
                "codec": "avc1.42C02A",
                "encoder": "Lavf58.45.100",
                "level": "4.2",
                "profile": "Baseline",
                "targetBitrate": 8789062,
                "targetFramerate": 60,
```

```
                "track": "Track0",
                "videoHeight": 1080,
                "videoWidth": 1920
            }
        },
        "ingestConfigurations": {
            "audioConfigurations": [
                {
                    "channels": 2,
                    "codec": "mp4a.40.2",
                    "sampleRate": 8000,
                    "targetBitrate": 46875,
                    "track": "Track0"
                }
            ],
            "videoConfigurations": [
                {
                    "codec": "avc1.42C02A",
                    "encoder": "Lavf58.45.100",
                    "level": "4.2",
                    "profile": "Baseline",
                    "targetBitrate": 8789062,
                    "targetFramerate": 60,
                    "track": "Track0",
                    "videoHeight": 1080,
                    "videoWidth": 1920
                }
            ]
        },
        "recordingConfiguration": {
            "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/
ABcdef34ghIJ",
            "name": "test-recording-config",
            "destinationConfiguration": {
                "s3": {
                    "bucketName": "demo-recording-bucket"
                }
            },
            "state": "ACTIVE",
            "tags": {
                "key1": "value1",
                "key2": "value2"
            },
            "thumbnailConfiguration": {
```

```
                "recordingMode": "INTERVAL",
                "targetIntervalSeconds": 1,
                "resolution": "LOWEST_RESOLUTION",
                "storage": [
                    "LATEST"
                ]
            },
            "recordingReconnectWindowSeconds": 60,
            "renditionConfiguration": {
                "renditionSelection": "CUSTOM",
                "renditions": [
                    "HD"
                ]
            }
        },
        "truncatedEvents": [
            {
                "code": "StreamTakeoverInvalidPriority",
                "name": "Stream Takeover Failure",
                "type": "IVS Stream State Change",
                "eventTime": "2023-06-26T19:09:48+00:00"
            },
            {
                "name": "Stream Takeover",
                "type": "IVS Stream State Change",
                "eventTime": "2023-06-26T19:09:47+00:00"
            },
            {
                "name": "Recording Start",
                "type": "IVS Recording State Change",
                "eventTime": "2023-06-26T19:09:35+00:00"
            },
            {
                "name": "Stream Start",
                "type": "IVS Stream State Change",
                "eventTime": "2023-06-26T19:09:34+00:00"
            },
            {
                "name": "Session Created",
                "type": "IVS Stream State Change",
                "eventTime": "2023-06-26T19:09:28+00:00"
            }
        ]
    }
```

```
}
```

如需詳細資訊，請參閱《IVS 低延遲使用者指南》中的建立頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetStreamSession。

get-stream

以下程式碼範例顯示如何使用 get-stream。

AWS CLI

取得串流的相關資訊

下列get-stream範例會取得指定頻道之串流的相關資訊。

```
aws ivs get-stream \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

輸出：

```
{
    "stream": {
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "startTime": "2020-05-05T21:55:38Z",
        "state": "LIVE",
        "health": "HEALTHY",
        "streamId": "st-ABCDEfghij01234KLMN5678",
        "viewerCount": 1
    }
}
```

如需詳細資訊，請參閱《IVS 低延遲使用者指南》中的建立頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetStream。

import-playback-key-pair

以下程式碼範例顯示如何使用 import-playback-key-pair。

AWS CLI

匯入新金鑰對的公有部分

下列import-playback-key-pair範例會匯入指定的公有金鑰（以 PEM 格式指定為字串），並傳回新金鑰對的 arn 和指紋。

```
aws ivs import-playback-key-pair \
    --name "my-playback-key" \
    --public-key-material "G1lbnQxOTA3BgNVBAMMMFdoeSBhcmUgeW91IGRl..."
```

輸出：

```
{
    "keyPair": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh",
        "name": "my-playback-key",
        "fingerprint": "0a:1b:2c:ab:cd:ef:34:56:70:b1:b2:71:01:2a:a3:72",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon 互動式影片服務使用者指南》中的設定私有頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ImportPlaybackKeyPair。

list-channels

以下程式碼範例顯示如何使用 list-channels。

AWS CLI

範例 1：取得所有頻道的摘要資訊

下列list-channels範例列出您 AWS 帳戶的所有頻道。

```
aws ivs list-channels
```

輸出：

```
{
    "channels": [
```

```
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
            "name": "channel-1",
            "latencyMode": "LOW",
            "authorized": false,
            "insecureIngest": false,
            "preset": "",
            "playbackRestrictionPolicyArn": "",
            "recordingConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABCD12cdEFgh",
            "tags": {},
            "type": "STANDARD"
        },
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl",
            "name": "channel-2",
            "latencyMode": "LOW",
            "authorized": false,
            "preset": "",
            "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ",
            "recordingConfigurationArn": "",
            "tags": {},
            "type": "STANDARD"
        }
    ]
}
```

如需詳細資訊，請參閱《IVS 低延遲使用者指南》中的建立頻道。

範例 2：若要取得所有頻道的摘要資訊，請依指定的 RecordingConfiguration ARN 篩選

下列`list-channels`範例列出與指定 RecordingConfiguration ARN 相關聯的 AWS 帳戶的所有頻道。

```
aws ivs list-channels \
    --filter-by-recording-configuration-arn "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABCD12cdEFgh"
```

輸出：

```
{
    "channels": [
```

```
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
            "name": "channel-1",
            "latencyMode": "LOW",
            "authorized": false,
            "insecureIngest": false,
            "preset": "",
            "playbackRestrictionPolicyArn": "",
            "recordingConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABCD12cdEFgh",
            "tags": {},
            "type": "STANDARD"
        }
    ]
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的記錄到 Amazon S3。

範例 3：若要取得所有頻道的摘要資訊，請依指定的 PlaybackRestrictionPolicy ARN 篩選

下列`list-channels`範例列出與指定 PlaybackRestrictionPolicy ARN 相關聯的 AWS 帳戶的所有頻道。

```
aws ivs list-channels \
    --filter-by-playback-restriction-policy-arn "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ"
```

輸出：

```
{
    "channels": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl",
            "name": "channel-2",
            "latencyMode": "LOW",
            "authorized": false,
            "preset": "",
            "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ",
            "recordingConfigurationArn": "",
            "tags": {},
            "type": "STANDARD"
        }
```

```
    ]
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的不需要的內容和檢視器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListChannels。

list-playback-key-pairs

以下程式碼範例顯示如何使用 list-playback-key-pairs。

AWS CLI

若要取得所有播放金鑰對的摘要資訊

下列list-playback-key-pairs範例會傳回所有金鑰對的相關資訊。

```
aws ivs list-playback-key-pairs
```

輸出：

```
{
    "keyPairs": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh",
            "name": "test-key-0",
            "tags": {}
        },
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:playback-key/ijkl5678mnop",
            "name": "test-key-1",
            "tags": {}
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon 互動式影片服務使用者指南》中的設定私有頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPlaybackKeyPairs。

list-playback-restriction-policies

以下程式碼範例顯示如何使用 list-playback-restriction-policies。

AWS CLI

取得所有播放限制政策的摘要資訊

下列list-playback-restriction-policies範例列出您 AWS 帳戶的所有播放限制政策。

```
aws ivs list-playback-restriction-policies
```

輸出：

```
{
    "playbackRestrictionPolicies": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ",
            "allowedCountries": [
                "US",
                "MX"
            ],
            "allowedOrigins": [
                "https://www.website1.com",
                "https://www.website2.com"
            ],
            "enableStrictOriginEnforcement": true,
            "name": "test-playback-restriction-policy",
            "tags": {
                "key1": "value1",
                "key2": "value2"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的不需要的內容和檢視器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPlaybackRestrictionPolicies。

list-recording-configurations

以下程式碼範例顯示如何使用 list-recording-configurations。

AWS CLI

列出在此帳戶中建立的所有 RecordingConfiguration 資源

下列`list-recording-configurations`範例會取得您帳戶中所有 RecordingConfiguration 資源的相關資訊。

```
aws ivs list-recording-configurations
```

輸出：

```
{
    "recordingConfigurations": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/
ABcdef34ghIJ",
            "name": "test-recording-config-1",
            "destinationConfiguration": {
                "s3": {
                    "bucketName": "demo-recording-bucket-1"
                }
            },
            "state": "ACTIVE",
            "tags": {}
        },
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/
CD12abcdGHIJ",
            "name": "test-recording-config-2",
            "destinationConfiguration": {
                "s3": {
                    "bucketName": "demo-recording-bucket-2"
                }
            },
            "state": "ACTIVE",
            "tags": {}
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的錄製至 Amazon S3。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRecordingConfigurations。

list-stream-keys

以下程式碼範例顯示如何使用 list-stream-keys。

AWS CLI

取得串流金鑰清單

下列list-stream-keys範例列出指定 ARN (Amazon Resource Name) 的所有串流金鑰。

```
aws ivs list-stream-keys \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

輸出：

```
{
    "streamKeys": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",
            "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
            "tags": {}
        }
    ]
}
```

FFor詳細資訊，請參閱 IVS 低延遲使用者指南中的建立頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListStreamKeys。

list-stream-sessions

以下程式碼範例顯示如何使用 list-stream-sessions。

AWS CLI

取得目前 AWS 區域中指定頻道的目前和先前串流摘要

下列list-stream-sessions範例會報告指定頻道 ARN (Amazon Resource Name) 之串流的摘要資訊。

```
aws ivs list-stream-sessions \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
    --max-results 25 \
    --next-token ""
```

輸出：

```
{
    "nextToken": "set-2",
    "streamSessions": [
        {
            "startTime": 1641578182,
            "endTime": 1641579982,
            "hasErrorEvent": false,
            "streamId": "mystream"
        }
        ...
    ]
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的建立頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListStreamSessions。

list-streams

以下程式碼範例顯示如何使用 list-streams。

AWS CLI

取得即時串流及其狀態的清單

下列list-streams範例列出您 AWS 帳戶的所有即時串流。

```
aws ivs list-streams
```

輸出：

```
{
   "streams": [
```

```
        {
            "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
            "state": "LIVE",
            "health": "HEALTHY",
            "streamId": "st-ABCDEfghij01234KLMN5678",
            "viewerCount": 1
        }
    ]
}
```

如需詳細資訊，請參閱《IVS 低延遲使用者指南》中的建立頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListStreams。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出 AWS 資源的所有標籤（例如：頻道、串流金鑰）

下列`list-tags-for-resource`範例列出指定資源 ARN (Amazon Resource Name) 的所有標籤。

```
aws ivs list-tags-for-resource \
    --resource-arn arn:aws:ivs:us-west-2:12345689012:channel/abcdABCDefgh
```

輸出：

```
{
    "tags":
    {
        "key1": "value1",
        "key2": "value2"
    }
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service API 參考》中的標記。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

put-metadata

以下程式碼範例顯示如何使用 put-metadata。

AWS CLI

將中繼資料插入指定頻道的作用中串流

下列put-metadata範例會將指定的中繼資料插入指定頻道的串流。

```
aws ivs put-metadata \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
    --metadata '{"my": "metadata"}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《IVS 低延遲使用者指南》中的建立頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutMetadata。

start-viewer-session-revocation

以下程式碼範例顯示如何使用 start-viewer-session-revocation。

AWS CLI

撤銷指定多個頻道 ARN 和檢視器 ID 對的檢視器工作階段

下列start-viewer-session-revocation範例會開始撤銷與指定頻道 ARN 和檢視器 ID 相關聯的檢視器工作階段，直到並包括指定的工作階段版本編號。如果未提供版本，則預設為 0。

```
aws ivs batch-start-viewer-session-revocation \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
    --viewer-id abcdefg \
    --viewer-session-versions-less-than-or-equal-to 1234567890
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon 互動式影片服務使用者指南》中的設定私有頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartViewerSessionRevocation。

stop-stream

以下程式碼範例顯示如何使用 stop-stream。

AWS CLI

停止指定的串流

下列stop-stream範例會停止指定頻道上的串流。

```
aws ivs stop-stream \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

此命令不會產生輸出。

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的建立頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopStream。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

新增或更新 AWS 資源的標籤 （ 例如：頻道、串流金鑰 ）

下列tag-resource範例會新增或更新指定資源 ARN (Amazon Resource Name) 的標籤。

```
aws ivs tag-resource \
    --resource-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
    --tags "tagkey1=tagvalue1, tagkey2=tagvalue2"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Interactive Video Service API 參考》中的標記。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

移除 AWS 資源的標籤（例如：頻道、串流金鑰）

下列untag-resource範例會移除指定資源 ARN (Amazon Resource Name) 的指定標籤。

```
aws ivs untag-resource \
    --resource-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
    --tag-keys "tagkey1, tagkey2"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Interactive Video Service API 參考》中的標記。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-channel

以下程式碼範例顯示如何使用 update-channel。

AWS CLI

範例 1：更新頻道的組態資訊

下列update-channel範例會更新指定頻道 ARN 的頻道組態，以變更頻道名稱。這不會影響此頻道的持續串流；您必須停止並重新啟動串流，變更才會生效。

```
aws ivs update-channel \
    --arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh' \
    --name 'channel-1' \
    --insecure-ingest
```

輸出：

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "name": "channel-1",
        "latencyMode": "LOW",
        "containerFormat": "TS",
        "multitrackInputConfiguration": {
```

```
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": true,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {}
}
```

如需詳細資訊，請參閱《IVS 低延遲使用者指南》中的建立頻道。

範例 2：更新頻道的組態以啟用錄製

下列update-channel範例會更新指定頻道 ARN 的頻道組態，以啟用錄製。這不會影響此頻道的持續串流；您必須停止並重新啟動串流，變更才會生效。

```
aws ivs update-channel \
    --arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh' \
    --no-insecure-ingest \
    --recording-configuration-arn 'arn:aws:ivs:us-west-2:123456789012:recording-
configuration/ABCD12cdEFgh'
```

輸出：

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "name": "test-channel-with-recording",
        "latencyMode": "LOW",
        "containerFormat": "TS",
        "multitrackInputConfiguration": {
```

```
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:recording-
configuration/ABCD12cdEFgh",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "BA1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的記錄到 Amazon S3。

範例 3：更新頻道的組態以停用錄製

下列update-channel範例會更新指定頻道 ARN 的頻道組態，以停用錄製。這不會影響此頻道的持續串流；您必須停止並重新啟動串流，變更才會生效。

```
aws ivs update-channel \
    --arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh' \
    --recording-configuration-arn ''
```

輸出：

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "name": "test-channel-with-recording",
        "latencyMode": "LOW",
        "containerFormat": "TS",
        "multitrackInputConfiguration": {
```

```
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2edfGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的記錄到 Amazon S3。

範例 4：更新頻道的組態以啟用播放限制

下列update-channel範例會更新指定頻道 ARN 的頻道組態，以套用播放限制政策。這不會影響此頻道的持續串流；您必須停止並重新啟動串流，變更才會生效。

```
aws ivs update-channel \
    --arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh' \
    --no-insecure-ingest \
    --playback-restriction-policy-arn 'arn:aws:ivs:us-west-2:123456789012:playback-
restriction-policy/ABcdef34ghIJ'
```

輸出：

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "name": "test-channel-with-playback-restriction-policy",
        "latencyMode": "LOW",
        "containerFormat": "TS",
```

```
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaCBDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的不需要的內容和檢視器。

範例 5：更新頻道的組態以停用播放限制

下列update-channel範例會更新指定頻道 ARN 的頻道組態，以停用播放限制。這不會影響此頻道的持續串流；您必須停止並重新啟動串流，變更才會生效。

```
aws ivs update-channel \
    --arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh' \
    --playback-restriction-policy-arn ''
```

輸出：

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "name": "test-channel-with-playback-restriction-policy",
        "latencyMode": "LOW",
        "containerFormat": "TS",
```

```
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDeFghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的不需要的內容和檢視器。

範例 6：更新頻道的組態以啟用多軌

下列update-channel範例會更新指定頻道 ARN 的頻道組態，以啟用多軌。這不會影響此頻道的持續串流；您必須停止並重新啟動串流，變更才會生效。

```
aws ivs update-channel \
    --arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh' \
    --container-format 'FRAGMENTED_MP4' \
    --multitrack-input-configuration '{"enabled": true,"maximumResolution":
 "FULL_HD","policy": "ALLOW"}'
```

輸出：

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "containerFormat": "FRAGMENTED_MP4",
        "name": "test-channel-with-multitrack",
```

```
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": true,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaCBDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的不需要的內容和檢視器。

範例 7：更新頻道的組態以停用播放限制

下列update-channel範例會更新指定頻道 ARN 的頻道組態，以停用多軌。這不會影響此頻道的持續串流；您必須停止並重新啟動串流，變更才會生效。

```
aws ivs update-channel \
    --arn 'arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh' \
    --container-format 'TS' \
    --multitrack-input-configuration '{"enabled": false}'
```

輸出：

```
{
    "channel": {
        ""arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "containerFormat": "TS",
        "name": "test-channel-with-multitrack",
```

```
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaCBDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {}
    }
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的不需要的內容和檢視器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateChannel。

update-playback-restriction-policy

以下程式碼範例顯示如何使用 update-playback-restriction-policy。

AWS CLI

更新播放限制政策

下列update-playback-restriction-policy範例會使用指定的政策 ARN 更新播放限制政策，以停用嚴格的原始伺服器強制執行。這不會影響相關聯頻道的持續串流；您必須停止並重新啟動串流，變更才會生效。

```
aws ivs update-playback-restriction-policy \
    --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ" \
```

```
    --no-enable-strict-origin-enforcement
```

輸出：

```
{
    "playbackRestrictionPolicy": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ",
        "allowedCountries": [
            "US",
            "MX"
        ],
        "allowedOrigins": [
            "https://www.website1.com",
            "https://www.website2.com"
        ],
        "enableStrictOriginEnforcement": false,
        "name": "test-playback-restriction-policy",
        "tags": {
            "key1": "value1",
            "key2": "value2"
        }
    }
}
```

如需詳細資訊，請參閱 IVS 低延遲使用者指南中的不需要的內容和檢視器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePlaybackRestrictionPolicy。

使用 的 Amazon IVS 聊天功能範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon IVS Chat 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-chat-token`

以下程式碼範例顯示如何使用 `create-chat-token`。

AWS CLI

建立聊天字符

下列`create-chat-token`範例會建立加密的聊天字符，用於建立與房間的個別 WebSocket 連線。字符有效期為一分鐘，而使用字符建立的連線（工作階段）在指定的持續時間內有效。

```
aws ivschat create-chat-token \
    --roomIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6", \
    --userId" "11231234" \
    --capabilities "SEND_MESSAGE", \
    --sessionDurationInMinutes" 30
```

輸出：

```
{
    "token": "ACEGmnoq#1rstu2...BDFH3vxwy!4hlm!#5",
    "sessionExpirationTime": "2022-03-16T04:44:09+00:00"
    "state": "CREATING",
    "tokenExpirationTime": "2022-03-16T03:45:09+00:00"
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的步驟 3：驗證和授權聊天用戶端。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateChatToken。

`create-logging-configuration`

以下程式碼範例顯示如何使用 `create-logging-configuration`。

AWS CLI

建立聊天 LoggingConfiguration 資源

下列`create-logging-configuration`範例會建立 LoggingConfiguration 資源，允許用戶端存放和記錄傳送的訊息。

```
aws ivschat create-logging-configuration \
    --destination-configuration s3={bucketName=demo-logging-bucket} \
    --name "test-logging-config" \
    --tags "key1=value1, key2=value2"
```

輸出：

```
{
    "arn": "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
ABcdef34ghIJ",
    "createTime": "2022-09-14T17:48:00.653000+00:00",
    "destinationConfiguration": {
        "s3": {
            "bucketName": "demo-logging-bucket"
        }
    },
    "id": "ABcdef34ghIJ",
    "name": "test-logging-config",
    "state": "ACTIVE",
    "tags": { "key1" : "value1", "key2" : "value2" },
    "updateTime": "2022-09-14T17:48:01.104000+00:00"
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 Amazon IVS 聊天功能入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLoggingConfiguration。

create-room

以下程式碼範例顯示如何使用 create-room。

AWS CLI

建立房間

下列create-room範例會建立新的房間。

```
aws ivschat create-room \
    --name "test-room-1" \
    --logging-configuration-identifiers "arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ" \
```

```
    --maximum-message-length 256 \
    --maximum-message-rate-per-second 5
```

輸出：

```
{
    "arn": "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6",
    "id": "g1H2I3j4k5L6",
    "createTime": "2022-03-16T04:44:09+00:00",
    "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ"],
    "maximumMessageLength": 256,
    "maximumMessageRatePerSecond": 5,
    "name": "test-room-1",
    "tags": {}
    "updateTime": "2022-03-16T07:22:09+00:00"
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的步驟 2：建立聊天室。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRoom。

`delete-logging-configuration`

以下程式碼範例顯示如何使用 `delete-logging-configuration`。

AWS CLI

刪除聊天 LoggingConfiguration 資源

下列`delete-logging-configuration`範例會刪除指定 ARN 的 LoggingConfiguration 資源。

```
aws ivschat delete-logging-configuration \
    --identifier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
ABcdef34ghIJ"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 Amazon IVS 聊天功能入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLoggingConfiguration。

delete-message

以下程式碼範例顯示如何使用 delete-message。

AWS CLI

從指定的房間刪除訊息

下列delete-message範例會將 甚至傳送至指定的房間，引導用戶端刪除指定的訊息：也就是從檢視中取消轉譯並從用戶端的聊天歷史記錄中刪除。

```
aws ivschat delete-message \
    --roomIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \
    --id "ABC123def456" \
    --reason "Message contains profanity"
```

輸出：

```
{
    "id": "12345689012"
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 Amazon IVS 聊天功能入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteMessage。

delete-room

以下程式碼範例顯示如何使用 delete-room。

AWS CLI

刪除房間

下列delete-room範例會刪除指定的房間。已連線的用戶端已中斷連線。成功時，它會傳回具有空白回應內文的 HTTP 204。

```
aws ivschat delete-room \
    --identifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 Amazon IVS 聊天功能入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRoom。

`disconnect-user`

以下程式碼範例顯示如何使用 `disconnect-user`。

AWS CLI

中斷使用者與房間的連線

下列`disconnect-user`範例會中斷指定使用者與指定房間的所有連線。成功時，它會傳回具有空白回應內文的 HTTP 200。

```
aws ivschat disconnect-user \
    --roomIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \
    --userId "ABC123def456" \
    --reason "Violated terms of service"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 Amazon IVS 聊天功能入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisconnectUser。

`get-logging-configuration`

以下程式碼範例顯示如何使用 `get-logging-configuration`。

AWS CLI

取得 LoggingConfiguration 資源的相關資訊

下列`get-logging-configuration`範例取得指定 ARN 的 LoggingConfiguration 資源相關資訊。

```
aws ivschat get-logging-configuration \
    --identifier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
ABcdef34ghIJ"
```

輸出：

```
{
    "arn": "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
ABcdef34ghIJ",
    "createTime": "2022-09-14T17:48:00.653000+00:00",
    "destinationConfiguration": {
        "s3": {
            "bucketName": "demo-logging-bucket"
        }
    },
    "id": "ABcdef34ghIJ",
    "name": "test-logging-config",
    "state": "ACTIVE",
    "tags": { "key1" : "value1", "key2" : "value2" },
    "updateTime": "2022-09-14T17:48:01.104000+00:00"
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 Amazon IVS 聊天功能入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLoggingConfiguration。

get-room

以下程式碼範例顯示如何使用 get-room。

AWS CLI

取得指定的房間

下列get-room範例會取得指定房間的相關資訊。

```
aws ivschat get-room \
    --identifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6"
```

輸出：

```
{
    "arn": "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6",
    "createTime": "2022-03-16T04:44:09+00:00",
```

```
    "id": "g1H2I3j4k5L6",
    "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ"],
    "maximumMessageLength": 256,
    "maximumMessageRatePerSecond": 5,
    "name": "test-room-1",
    "tags": {},
    "updateTime": "2022-03-16T07:22:09+00:00"
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 Amazon IVS 聊天功能入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRoom。

`list-logging-configurations`

以下程式碼範例顯示如何使用 `list-logging-configurations`。

AWS CLI

若要取得處理 API 請求之 AWS 區域中使用者所有記錄組態的摘要資訊

下列`list-logging-configurations`範例列出處理 API 請求之 AWS 區域中使用者的所有 LoggingConfiguration 資源的相關資訊。

```
aws ivschat list-logging-configurations \
    --max-results 2 \
    --next-token ""
```

輸出：

```
{
    "nextToken": "set-2",
    "loggingConfigurations": [
        {
            "arn": "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
ABcdef34ghIJ",
            "createTime": "2022-09-14T17:48:00.653000+00:00",
            "destinationConfiguration": {
                "s3": {
                    "bucketName": "demo-logging-bucket"
```

```
                }
            },
            "id": "ABcdef34ghIJ",
            "name": "test-logging-config",
            "state": "ACTIVE",
            "tags": { "key1" : "value1", "key2" : "value2" },
            "updateTime": "2022-09-14T17:48:01.104000+00:00"
        }
        ...
    ]
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 Amazon IVS 聊天功能入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListLoggingConfigurations。

list-rooms

以下程式碼範例顯示如何使用 list-rooms。

AWS CLI

若要取得目前區域中所有房間的摘要資訊

下列list-rooms範例會取得處理請求之 AWS 區域中所有房間的摘要資訊。結果會依 updateTime 的遞減順序排序。

```
aws ivschat list-rooms \
    --logging-configuration-identifier "arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ" \
    --max-results 10 \
    --next-token ""
```

輸出：

```
{
    "nextToken": "page3",
    "rooms": [
        {
            "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6",
            "createTime": "2022-03-16T04:44:09+00:00",
```

```
            "id": "g1H2I3j4k5L6",
            "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ"],
            "name": "test-room-1",
            "tags": {},
            "updateTime": "2022-03-16T07:22:09+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 Amazon IVS 聊天功能入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRooms。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出 AWS 資源的所有標籤 （ 例如：房間 ）

下列`list-tags-for-resource`範例列出指定資源 ARN (Amazon Resource Name) 的所有標籤。

```
aws ivschat list-tags-for-resource \
    --resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6
```

輸出：

```
{
    "tags":
    {
        "key1": "value1",
        "key2": "value2"
    }
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service API 參考》中的標記。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

send-event

以下程式碼範例顯示如何使用 send-event。

AWS CLI

將事件傳送至房間

下列send-event範例會將指定的事件傳送至指定的房間。

```
aws ivschat send-event \
    --roomIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \
    --eventName "SystemMessage" \
    --attributes \
        "msgType"="user-notification", \
        "msgText"="This chat room will close in 15 minutes."
```

輸出：

```
{
    "id": "12345689012"
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 Amazon IVS 聊天功能入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SendEvent。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

新增或更新 AWS 資源的標籤 （ 例如：房間 ）

下列tag-resource範例會新增或更新指定資源 ARN (Amazon Resource Name) 的標籤。成功時，它會傳回具有空白回應內文的 HTTP 200。

```
aws ivschat tag-resource \
    --resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6 \
    --tags "tagkey1=tagkeyvalue1, tagkey2=tagkeyvalue2"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Interactive Video Service API 參考》中的標記。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

移除 AWS 資源的標籤（例如：房間）

下列untag-resource範例會移除指定資源 ARN (Amazon Resource Name) 的指定標籤。成功時，它會傳回具有空白回應內文的 HTTP 200。

```
aws ivschat untag-resource \
    --resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6 \
    --tag-keys "tagkey1, tagkey2"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Interactive Video Service API 參考》中的標記。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-logging-configuration

以下程式碼範例顯示如何使用 update-logging-configuration。

AWS CLI

更新房間的記錄組態

下列update-logging-configuration範例會使用指定的資料更新 LoggingConfiguration 資源。

```
aws ivschat update-logging-configuration \
    --destination-configuration s3={bucketName=demo-logging-bucket} \
    --identifier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
ABcdef34ghIJ" \
    --name "test-logging-config"
```

輸出：

```
{
    "arn": "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
ABcdef34ghIJ",
    "createTime": "2022-09-14T17:48:00.653000+00:00",
    "destinationConfiguration": {
        "s3": {
            "bucketName": "demo-logging-bucket"
        }
    },
    "id": "ABcdef34ghIJ",
    "name": "test-logging-config",
    "state": "ACTIVE",
    "tags": { "key1" : "value1", "key2" : "value2" },
    "updateTime": "2022-09-14T17:48:01.104000+00:00"
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 Amazon IVS 聊天功能入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateLoggingConfiguration。

update-room

以下程式碼範例顯示如何使用 update-room。

AWS CLI

更新房間的組態

下列update-room範例會使用指定的資料更新指定房間的組態。

```
aws ivschat update-room \
    --identifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \
    --logging-configuration-identifiers "arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ" \
    --name "chat-room-a" \
    --maximum-message-length 256 \
    --maximum-message-rate-per-second 5
```

輸出：

```
{
    "arn": "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6",
    "createTime": "2022-03-16T04:44:09+00:00",
    "id": "g1H2I3j4k5L6",
    "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ"],
    "maximumMessageLength": 256,
    "maximumMessageRatePerSecond": 5,
    "name": "chat-room-a",
    "tags": {},
    "updateTime": "2022-03-16T07:22:09+00:00"
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 Amazon IVS 聊天功能入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRoom。

使用 的 Amazon IVS 即時串流範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon IVS 即時串流來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-encoder-configuration`

以下程式碼範例顯示如何使用 `create-encoder-configuration`。

AWS CLI

建立合成編碼器組態

下列create-encoder-configuration範例會建立具有指定屬性的合成編碼器組態。

```
aws ivs-realtime create-encoder-configuration \
    --name test-ec --video bitrate=3500000,framerate=30.0,height=1080,width=1920
```

輸出：

```
{
    "encoderConfiguration": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
ABabCDcdEFef",
        "name": "test-ec",
        "tags": {},
        "video": {
            "bitrate": 3500000,
            "framerate": 30,
            "height": 1080,
            "width": 1920
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateEncoderConfiguration。

create-ingest-configuration

以下程式碼範例顯示如何使用 create-ingest-configuration。

AWS CLI

建立擷取組態

下列create-ingest-configuration範例會使用 RTMPS 通訊協定建立擷取組態。

```
aws ivs-realtime create-ingest-configuration \
    --name ingest1 \
    --ingest-protocol rtmps
```

輸出：

```
{
    "ingestConfiguration": {
        "name": "ingest1",
        "arn": "arn:aws:ivs:us-west-2:123456789012:ingest-configuration/
AbCdEfGh1234",
        "ingestProtocol": "RTMPS",
        "streamKey": "rt_123456789012_us-
west-2_AbCdEfGh1234_abcd1234efgh5678ijkl9012MNOP34",
        "stageArn": "",
        "participantId": "xyZ654abC321",
        "state": "INACTIVE",
        "userId": "",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 IVS 串流擷取 | 即時串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateIngestConfiguration。

create-participant-token

以下程式碼範例顯示如何使用 create-participant-token。

AWS CLI

建立階段參與者字符

下列create-participant-token範例會為指定的階段建立參與者 Toke。

```
aws ivs-realtime create-participant-token \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --user-id bob
```

輸出：

```
{
    "participantToken": {
        "expirationTime": "2023-03-07T09:47:43+00:00",
        "participantId": "ABCDEfghij01234KLMN6789",
        "token": "abcd1234defg5678"
```

```
    }
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateParticipantToken。

`create-stage`

以下程式碼範例顯示如何使用 `create-stage`。

AWS CLI

範例 1：建立階段

下列`create-stage`範例會為指定的使用者建立階段和階段參與者字符。

```
aws ivs-realtime create-stage \
    --name stage1 \
    --participant-token-configurations userId=alice
```

輸出：

```
{
    "participantTokens": [
        {
            "participantId": "ABCDEfghij01234KLMN5678",
            "token": "a1b2c3d4567890ab",
            "userId": "alice"
        }
    ],
    "stage": {
        "activeSessionId": "st-a1b2c3d4e5f6g",
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "autoParticipantRecordingConfiguration": {
            "storageConfigurationArn": "",
            "mediaTypes": [
                "AUDIO_VIDEO"
            ],
            "thumbnailConfiguration": {
                "targetIntervalSeconds": 60,
```

```
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "DISABLED"
            },
            "recordingReconnectWindowSeconds": 0,
            "hlsConfiguration": {
                "targetSegmentDurationSeconds": 6
            },
            "recordParticipantReplicas": true
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://9x0y8z7s6t5u.global-bm.whip.live-video.net"
        },
        "name": "stage1",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon IVS 低延遲串流使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

範例 2：建立階段並設定個別參與者記錄

下列create-stage範例會建立階段並設定個別參與者記錄。

```
aws ivs-realtime create-stage \
    --name stage1 \
    --auto-participant-recording-configuration '{"mediaTypes":
 ["AUDIO_VIDEO"],"storageConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:storage-configuration/abcdABCDefgh",
 "recordingReconnectWindowSeconds": 100, \
        "hlsConfiguration": {"targetSegmentDurationSeconds": 5}}'
```

輸出：

```
{
    "stage": {
```

```
        "activeSessionId": "st-a1b2c3d4e5f6g",
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "autoParticipantRecordingConfiguration": {
            "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-
configuration/abcdABCDefgh"
            "mediaTypes": [
                "AUDIO_VIDEO"
            ],
            "thumbnailConfiguration": {
                "targetIntervalSeconds": 60,
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "DISABLED"
            },
            "recordingReconnectWindowSeconds": 100,
            "hlsConfiguration": {
                "targetSegmentDurationSeconds": 5
            },
            "recordParticipantReplicas": true
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://9x0y8z7s6t5u.global-bm.whip.live-video.net"
        },
        "name": "stage1",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon IVS 低延遲串流使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

範例 3：建立階段並設定啟用縮圖錄製的個別參與者錄製

下列create-stage範例會建立階段，並設定啟用縮圖錄製的個別參與者錄製。

```
aws ivs-realtime create-stage \
    --name stage1 \
```

```
    --auto-participant-recording-configuration '{"mediaTypes":
 ["AUDIO_VIDEO"],"storageConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:storage-configuration/abcdABCDefgh", \
        "thumbnailConfiguration": {"recordingMode": "INTERVAL","storage":
 ["SEQUENTIAL"],"targetIntervalSeconds": 60}}'
```

輸出：

```
{
    "stage": {
        "activeSessionId": "st-a1b2c3d4e5f6g",
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "autoParticipantRecordingConfiguration": {
            "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-
configuration/abcdABCDefgh",
            "mediaTypes": [
                "AUDIO_VIDEO"
            ],
            "thumbnailConfiguration": {
                "targetIntervalSeconds": 60,
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "INTERVAL"
            },
            "recordingReconnectWindowSeconds": 0,
            "hlsConfiguration": {
                "targetSegmentDurationSeconds": 6
            },
            "recordParticipantReplicas": true
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://9x0y8z7s6t5u.global-bm.whip.live-video.net"
        },
        "name": "stage1",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon IVS 低延遲串流使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateStage。

create-storage-configuration

以下程式碼範例顯示如何使用 create-storage-configuration。

AWS CLI

建立合成儲存組態

下列create-storage-configuration範例會建立具有指定屬性的合成儲存組態。

```
aws ivs-realtime create-storage-configuration \
    --name "test-sc" --s3 "bucketName=amzn-s3-demo-bucket"
```

輸出：

```
{
    "storageConfiguration": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
ABabCDcdEFef",
        "name": "test-sc",
        "s3": {
            "bucketName": "amzn-s3-demo-bucket"
        },
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateStorageConfiguration。

delete-encoder-configuration

以下程式碼範例顯示如何使用 delete-encoder-configuration。

AWS CLI

刪除合成編碼器組態

以下內容會delete-encoder-configuration刪除指定 ARN (Amazon Resource Name) 指定的合成編碼器組態。

```
aws ivs-realtime delete-encoder-configuration \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteEncoderConfiguration。

delete-ingest-configuration

以下程式碼範例顯示如何使用 delete-ingest-configuration。

AWS CLI

範例 1：刪除非作用中的擷取組態

下列delete-ingest-configuration範例會刪除指定擷取組態 ARN (Amazon Resource Name) 的非作用中擷取組態。

```
aws ivs-realtime delete-ingest-configuration \
    --arn arn:aws:ivs:us-west-2:123456789012:ingest-configuration/AbCdEfGh1234
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 IVS 串流擷取 | 即時串流。

範例 2：強制刪除作用中的擷取組態

下列delete-ingest-configuration範例會強制刪除指定擷取組態 ARN (Amazon Resource Name) 的作用中擷取組態。

```
aws ivs-realtime delete-ingest-configuration \
```

```
    --arn arn:aws:ivs:us-west-2:123456789012:ingest-configuration/AbCdEfGh1234 \
    --force
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 IVS 串流擷取 | 即時串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteIngestConfiguration。

delete-public-key

以下程式碼範例顯示如何使用 delete-public-key。

AWS CLI

刪除公有金鑰

以下內容會delete-public-key刪除指定的公有金鑰。

```
aws ivs-realtime delete-public-key \
    --arn arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon IVS 即時串流使用者指南》中的分發參與者字符。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePublicKey。

delete-stage

以下程式碼範例顯示如何使用 delete-stage。

AWS CLI

刪除階段

下列delete-stage範例會刪除指定的階段。

```
aws ivs-realtime delete-stage \
    --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteStage。

delete-storage-configuration

以下程式碼範例顯示如何使用 delete-storage-configuration。

AWS CLI

刪除合成儲存組態

以下內容會delete-storage-configuration刪除指定 ARN (Amazon Resource Name) 指定的合成儲存組態。

```
aws ivs-realtime delete-storage-configuration \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
ABabCDcdEFef"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteStorageConfiguration。

disconnect-participant

以下程式碼範例顯示如何使用 disconnect-participant。

AWS CLI

中斷連接階段參與者

下列disconnect-participant範例會中斷指定參與者與指定階段的連線。

```
aws ivs-realtime disconnect-participant \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --participant-id ABCDEfghij01234KLMN5678
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisconnectParticipant。

`get-composition`

以下程式碼範例顯示如何使用 `get-composition`。

AWS CLI

範例 1：使用預設配置設定取得合成

下列`get-composition`範例取得指定 ARN (Amazon Resource Name) 的合成。

```
aws ivs-realtime get-composition \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh"
```

輸出：

```
{
    "composition": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh",
        "destinations": [
            {
                "configuration": {
                    "channel": {
                        "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
                        "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                    },
                    "name": ""
                },
                "id": "AabBCcdDEefF",
                "startTime": "2023-10-16T23:26:00+00:00",
                "state": "ACTIVE"
            },
            {
                "configuration": {
                    "name": "",
                    "s3": {
```

```
                        "encoderConfigurationArns": [
                            "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                        ],
                        "recordingConfiguration": {
                            "format": "HLS",
                            "hlsConfiguration": {
                                "targetSegmentDurationSeconds": 2
                            }
                        },
                        "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE",
                    }
                },
                "detail": {
                    "s3": {
                        "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
                    }
                },
                "id": "GHFabcgefABC",
                "startTime": "2023-10-16T23:26:00+00:00",
                "state": "STARTING"
            }
        ],
        "layout": {
            "grid": {
                "featuredParticipantAttribute": ""
                "gridGap": 2,
                "omitStoppedVideo": false,
                "videoAspectRatio": "VIDEO",
                "videoFillMode": ""
            }
        },
        "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
        "startTime": "2023-10-16T23:24:00+00:00",
        "state": "ACTIVE",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon IVS 即時串流使用者指南》中的 IVS 複合錄製 | 即時串流。

範例 2：使用 PiP 配置取得合成

下列get-composition範例取得指定 ARN (Amazon Resource Name) 的合成，其使用 PiP 配置。

```
aws ivs-realtime get-composition \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/wxyzWXYZpqrs"
```

輸出：

```
{
    "composition": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/wxyzWXYZpqrs",
        "destinations": [
            {
                "configuration": {
                    "channel": {
                        "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
                        "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                    },
                    "name": ""
                },
                "id": "AabBCcdDEefF",
                "startTime": "2023-10-16T23:26:00+00:00",
                "state": "ACTIVE"
            },
            {
                "configuration": {
                    "name": "",
                    "s3": {
                        "encoderConfigurationArns": [
                            "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                        ],
                        "recordingConfiguration": {
                            "format": "HLS",
                            "hlsConfiguration": {
                                "targetSegmentDurationSeconds": 2
                            }
                        },
                        "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE"
                    }
```

```
                },
                "detail": {
                    "s3": {
                        "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
                    }
                },
                "id": "GHFabcgefABC",
                "startTime": "2023-10-16T23:26:00+00:00",
                "state": "STARTING"
            }
        ],
        "layout": {
            "pip": {
                "featuredParticipantAttribute": "abcdefg",
                "gridGap": 0,
                "omitStoppedVideo": false,
                "pipBehavior": "STATIC",
                "pipOffset": 0,
                "pipParticipantAttribute": "",
                "pipPosition": "BOTTOM_RIGHT",
                "videoFillMode": "COVER"
            }
        },
        "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
        "startTime": "2023-10-16T23:24:00+00:00",
        "state": "ACTIVE",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon IVS 即時串流使用者指南》中的 IVS 複合錄製 | 即時串流。

範例 3：取得已啟用縮圖記錄的合成

下列`get-composition`範例取得指定 ARN (Amazon Resource Name) 的合成，該 ARN 已使用預設設定啟用縮圖錄製。

```
aws ivs-realtime get-composition \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh"
```

輸出：

```
{
    "composition": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh",
        "destinations": [
            {
                "configuration": {
                    "channel": {
                        "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
                        "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                    },
                    "name": ""
                },
                "id": "AabBCcdDEefF",
                "startTime": "2023-10-16T23:26:00+00:00",
                "state": "ACTIVE"
            },
            {
                "configuration": {
                    "name": "",
                    "s3": {
                        "encoderConfigurationArns": [
                            "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                        ],
                        "recordingConfiguration": {
                            "format": "HLS",
                            "hlsConfiguration": {
                                "targetSegmentDurationSeconds": 2
                            }
                        },
                        "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE",
                        "thumbnailConfigurations": [
                           {
                              "targetIntervalSeconds": 60,
                              "storage": [
                                  "SEQUENTIAL"
                              ],
                           }
                        ]
                    }
```

```
                },
                "detail": {
                    "s3": {
                        "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
                    }
                },
                "id": "GHFabcgefABC",
                "startTime": "2023-10-16T23:26:00+00:00",
                "state": "STARTING"
            }
        ],
        "layout": {
            "grid": {
                "featuredParticipantAttribute": ""
                "gridGap": 2,
                "omitStoppedVideo": false,
                "videoAspectRatio": "VIDEO",
                "videoFillMode": ""                }
        },
        "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
        "startTime": "2023-10-16T23:24:00+00:00",
        "state": "ACTIVE",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon IVS 即時串流使用者指南》中的 IVS 複合錄製 | 即時串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetComposition。

get-encoder-configuration

以下程式碼範例顯示如何使用 get-encoder-configuration。

AWS CLI

取得合成編碼器組態

下列get-encoder-configuration範例會取得指定 ARN (Amazon Resource Name) 指定的合成編碼器組態。

```
aws ivs-realtime get-encoder-configuration \
```

```
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
abcdABCDefgh"
```

輸出：

```
{
    "encoderConfiguration": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
abcdABCDefgh",
        "name": "test-ec",
        "tags": {},
        "video": {
            "bitrate": 3500000,
            "framerate": 30,
            "height": 1080,
            "width": 1920
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetEncoderConfiguration。

get-ingest-configuration

以下程式碼範例顯示如何使用 get-ingest-configuration。

AWS CLI

取得擷取組態資訊

下列get-ingest-configuration範例取得指定擷取組態 ARN (Amazon Resource Name) 的擷取組態。

```
aws ivs-realtime get-ingest-configuration \
    --arn arn:aws:ivs:us-west-2:123456789012:ingest-configuration/AbCdEfGh1234
```

輸出：

```
{
```

```
    "ingestConfiguration": {
        "name": "ingest1",
        "arn": "arn:aws:ivs:us-west-2:123456789012:ingest-configuration/
AbCdEfGh1234",
        "ingestProtocol": "RTMPS",
        "streamKey": "rt_123456789012_us-
west-2_AbCdEfGh1234_abcd1234efgh5678ijkl9012MNOP34",
        "stageArn": "",
        "participantId": "xyZ654abC321",
        "state": "INACTIVE",
        "userId": "",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 IVS 串流擷取 | 即時串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIngestConfiguration。

get-participant

以下程式碼範例顯示如何使用 get-participant。

AWS CLI

範例 1：取得階段參與者

下列get-participant範例取得指定階段 ARN (Amazon Resource Name) 中指定參與者 ID 和工作階段 ID 的階段參與者。

```
aws ivs-realtime get-participant \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g \
    --participant-id abCDEf12GHIj
```

輸出：

```
{
    "participant": {
        "browserName": "Google Chrome",
        "browserVersion": "116",
```

```
        "firstJoinTime": "2023-04-26T20:30:34+00:00",
        "ispName": "Comcast",
        "osName": "Microsoft Windows 10 Pro",
        "osVersion": "10.0.19044",
        "participantId": "abCDEf12GHIj",
        "published": true,
        "recordingS3BucketName": "bucket-name",
        "recordingS3Prefix": "abcdABCDefgh/st-a1b2c3d4e5f6g/
abCDEf12GHIj/1234567890",
        "recordingState": "ACTIVE",
        "sdkVersion": "",
        "state": "CONNECTED",
        "userId": ""
    }
}
```

如需詳細資訊，請參閱《Amazon IVS 低延遲串流使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

範例 2：取得已複寫至另一個階段的階段參與者

當參與者也複寫到另一個階段時，以下get-participant範例會取得指定階段 ARN (Amazon Resource Name) 中指定參與者 ID 和工作階段 ID 的階段參與者。

```
aws ivs-realtime get-participant \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g \
    --participant-id abCDEf12GHIj
```

輸出：

```
{
    "participant": {
        "browserName": "Google Chrome",
        "browserVersion": "116",
        "firstJoinTime": "2023-04-26T20:30:34+00:00",
        "ispName": "Comcast",
        "osName": "Microsoft Windows 10 Pro",
        "osVersion": "10.0.19044",
        "participantId": "abCDEf12GHIj",
        "published": true,
        "recordingS3BucketName": "bucket-name",
```

```
        "recordingS3Prefix": "abcdABCDefgh/st-a1b2c3d4e5f6g/
abCDEf12GHIj/1234567890",
        "recordingState": "ACTIVE",
        "replicationState": "ACTIVE",
        "replicationType": "SOURCE",
        "sdkVersion": "",
        "state": "CONNECTED",
        "userId": ""
    }
}
```

如需詳細資訊，請參閱《Amazon IVS 低延遲串流使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

範例 3：取得已從另一個階段複寫的階段參與者

下列get-participant範例會從另一個階段複寫參與者時，取得指定階段 ARN (Amazon Resource Name) 中指定參與者 ID 和工作階段 ID 的階段參與者。

```
aws ivs-realtime get-participant \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g \
    --participant-id abCDEf12GHIj
```

輸出：

```
{
    "participant": {
        "browserName": "Google Chrome",
        "browserVersion": "116",
        "firstJoinTime": "2023-04-26T20:30:34+00:00",
        "ispName": "Comcast",
        "osName": "Microsoft Windows 10 Pro",
        "osVersion": "10.0.19044",
        "participantId": "abCDEf12GHIj",
        "published": true,
        "recordingS3BucketName": "bucket-name",
        "recordingS3Prefix": "abcdABCDefgh/st-a1b2c3d4e5f6g/
abCDEf12GHIj/1234567890",
        "recordingState": "ACTIVE",
        "replicationState": "ACTIVE",
        "replicationType": "REPLICA",
```

```
        "sdkVersion": "",
        "state": "CONNECTED",
        "userId": ""
    }
}
```

如需詳細資訊，請參閱《Amazon IVS 低延遲串流使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetParticipant。

get-public-key

以下程式碼範例顯示如何使用 get-public-key。

AWS CLI

取得用來簽署階段參與者字符的現有公有金鑰

下列get-public-key範例取得由提供的 ARN 指定的公有金鑰，用於簽署階段參與者字符。

```
aws ivs-realtime get-public-key \
    --arn arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2
```

輸出：

```
{
    "publicKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2",
        "name": "",
        "publicKeyMaterial": "-----BEGIN PUBLIC KEY-----
\nMHYwEAYHKoZIzj0CAQYFK4EEACIDYgAEqVWUtqs6EktQMR1sCYmEzGvRwtaycI16\n9pmzcpiWu/
uhNStGlteJ5odRfRwVkoQUMnSZXTCcbn9bBTTmiWo4mJcFOOAzsthH
\n0UAb8NdD4tUE0At4a9hYP9IETEXAMPLE\n-----END PUBLIC KEY-----",
        "fingerprint": "12:a3:44:56:bc:7d:e8:9f:10:2g:34:hi:56:78:90:12",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon IVS 即時串流使用者指南》中的分發參與者字符。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPublicKey。

get-stage-session

以下程式碼範例顯示如何使用 get-stage-session。

AWS CLI

取得階段工作階段

下列get-stage-session範例取得指定階段 ARN (Amazon Resource Name) 之指定工作階段 ID 的階段工作階段。

```
aws ivs-realtime get-stage-session \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g
```

輸出：

```
{
    "stageSession": {
        "endTime": "2023-04-26T20:36:29+00:00",
        "sessionId": "st-a1b2c3d4e5f6g",
        "startTime": "2023-04-26T20:30:29.602000+00:00"
    }
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetStageSession。

get-stage

以下程式碼範例顯示如何使用 get-stage。

AWS CLI

取得階段的組態資訊

下列get-stage範例取得指定階段 ARN (Amazon Resource Name) 的階段組態。

```
aws ivs-realtime get-stage \
```

```
    --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

輸出：

```
{
    "stage": {
        "activeSessionId": "st-a1b2c3d4e5f6g",
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "autoParticipantRecordingConfiguration": {
            "storageConfigurationArn": "",
            "mediaTypes": [
                "AUDIO_VIDEO"
            ],
            "thumbnailConfiguration": {
                "targetIntervalSeconds": 60,
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "DISABLED"
            },
            "recordingReconnectWindowSeconds": 0,
            "hlsConfiguration": {
                "targetSegmentDurationSeconds": 6
            },
            "recordParticipantReplicas": true
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://9x0y8z7s6t5u.global-bm.whip.live-video.net"
        },
        "name": "test",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon IVS 低延遲串流使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetStage。

get-storage-configuration

以下程式碼範例顯示如何使用 get-storage-configuration。

AWS CLI

取得合成儲存組態

下列get-storage-configuration範例會取得指定 ARN (Amazon Resource Name) 所指定的合成儲存組態。

```
aws ivs-realtime get-storage-configuration \
    --name arn "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
abcdABCDefgh"
```

輸出：

```
{
    "storageConfiguration": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
abcdABCDefgh",
        "name": "test-sc",
        "s3": {
            "bucketName": "amzn-s3-demo-bucket"
        },
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetStorageConfiguration。

import-public-key

以下程式碼範例顯示如何使用 import-public-key。

AWS CLI

匯入用於簽署階段參與者字符的現有公有金鑰

下列import-public-key範例會從物料檔案匯入公有金鑰，用於簽署階段參與者字符。

```
aws ivs-realtime import-public-key \
    --public-key-material="`cat public.pem`"
```

輸出：

```
{
    "publicKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2",
        "name": "",
        "publicKeyMaterial": "-----BEGIN PUBLIC KEY-----
\nMHYwEAYHKoZIzj0CAQYFK4EEACIDYgAEqVWUtqs6EktQMR1sCYmEzGvRwtaycI16\n9pmzcpiWu/
uhNStGlteJ5odRfRwVkoQUMnSZXTCcbn9bBTTmiWo4mJcFOOAzsthH
\n0UAb8NdD4tUE0At4a9hYP9IETEXAMPLE\n-----END PUBLIC KEY-----",
        "fingerprint": "12:a3:44:56:bc:7d:e8:9f:10:2g:34:hi:56:78:90:12",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon IVS 即時串流使用者指南》中的分發參與者字符。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ImportPublicKey。

`list-compositions`

以下程式碼範例顯示如何使用 `list-compositions`。

AWS CLI

取得合成的清單

以下`list-compositions`列出處理 API 請求的 AWS 區域中，您 AWS 帳戶的所有組成。

```
aws ivs-realtime list-compositions
```

輸出：

```
{
    "compositions": [
        {
            "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/
abcdABCDefgh",
```

```
            "destinations": [
                {
                    "id": "AabBCcdDEefF",
                    "startTime": "2023-10-16T23:25:23+00:00",
                    "state": "ACTIVE"
                }
            ],
            "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/
defgABCDabcd",
            "startTime": "2023-10-16T23:25:21+00:00",
            "state": "ACTIVE",
            "tags": {}
        },
        {
            "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/
ABcdabCDefgh",
            "destinations": [
                {
                    "endTime": "2023-10-16T23:25:00.786512+00:00",
                    "id": "aABbcCDdeEFf",
                    "startTime": "2023-10-16T23:24:01+00:00",
                    "state": "STOPPED"
                },
                {
                    "endTime": "2023-10-16T23:25:00.786512+00:00",
                    "id": "deEFfaABbcCD",
                    "startTime": "2023-10-16T23:24:01+00:00",
                    "state": "STOPPED"
                }
            ],
            "endTime": "2023-10-16T23:25:00+00:00",
            "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/
efghabcdABCD",
            "startTime": "2023-10-16T23:24:00+00:00",
            "state": "STOPPED",
            "tags": {}
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCompositions。

list-encoder-configurations

以下程式碼範例顯示如何使用 list-encoder-configurations。

AWS CLI

列出合成編碼器組態

以下list-encoder-configurations列出處理 API 請求的 AWS 區域中，您 AWS 帳戶的所有合成編碼器組態。

```
aws ivs-realtime list-encoder-configurations
```

輸出：

```
{
    "encoderConfigurations": [
        {
            "arn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
abcdABCDefgh",
            "name": "test-ec-1",
            "tags": {}
        },
        {
            "arn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
ABCefgEFGabc",
            "name": "test-ec-2",
            "tags": {}
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEncoderConfigurations。

list-ingest-configurations

以下程式碼範例顯示如何使用 list-ingest-configurations。

AWS CLI

取得所有擷取組態的摘要資訊

下列list-ingest-configurations範例列出處理 API 請求 AWS 的區域中，您 AWS 帳戶的所有擷取組態。

```
aws ivs-realtime list-ingest-configurations
```

輸出：

```
{
    "ingestConfigurations": [
        {
            "name": "",
            "arn": "arn:aws:ivs:us-west-2:123456789012:ingest-configuration/
XYZuvwSt4567",
            "ingestProtocol": "RTMPS",
            "stageArn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
            "participnatId": "abC789Xyz456",
            "state": "INACTIVE"
            "userId": "",
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 IVS 串流擷取 | 即時串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListIngestConfigurations。

list-participant-events

以下程式碼範例顯示如何使用 list-participant-events。

AWS CLI

範例 1：取得階段參與者事件的清單

下列list-participant-events範例列出指定參與者 ID 的所有參與者事件，以及指定階段 ARN (Amazon Resource Name) 的工作階段 ID。

```
aws ivs-realtime list-participant-events \
```

```
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g \
    --participant-id abCDEf12GHIj
```

輸出：

```
{
    "events": [
        {
            "eventTime": "2023-04-26T20:36:28+00:00",
            "name": "LEFT",
            "participantId": "abCDEf12GHIj"
        },
        {
            "eventTime": "2023-04-26T20:36:28+00:00",
            "name": "PUBLISH_STOPPED",
            "participantId": "abCDEf12GHIj"
        },
        {
            "eventTime": "2023-04-26T20:30:34+00:00",
            "name": "JOINED",
            "participantId": "abCDEf12GHIj"
        },
        {
            "eventTime": "2023-04-26T20:30:34+00:00",
            "name": "PUBLISH_STARTED",
            "participantId": "abCDEf12GHIj"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon IVS 低延遲串流使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

範例 2：取得階段參與者事件的清單，包括參與者複寫停止和開始

下列`list-participant-events`範例列出指定階段 ARN (Amazon Resource Name) 之指定工作階段 ID 的所有參與者事件，其中參與者會複寫至另一個階段。

```
aws ivs-realtime list-participant-events \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g \
```

```
    --participant-id abCDEf12GHIj
```

輸出：

```
{
    "events": [
        {
            "eventTime": "2025-04-26T20:36:28+00:00",
            "name": "LEFT",
            "participantId": "abCDEf12GHIj"
        },
        {
            "eventTime": "2025-04-26T20:36:28+00:00",
            "name": "PUBLISH_STOPPED",
            "participantId": "abCDEf12GHIj"
        },
        {
            "eventTime": "2025-04-26T20:30:34+00:00",
            "name": "JOINED",
            "participantId": "abCDEf12GHIj"
        },
        {
            "eventTime": "2025-04-26T20:30:34+00:00",
            "name": "PUBLISH_STARTED",
            "participantId": "abCDEf12GHIj"
        },
        {
            "name": "REPLICATION_STARTED",
            "participantId": "abCDEf12GHIj",
            "eventTime": "2025-04-26T20:30:34+00:00",
            "destinationStageArn": "arn:aws:ivs:us-west-2:12345678901:stage/
ABCDabcdefgh",
            "destinationSessionId": "st-b1c2d3e4f5g6a"
        },
        {
            "name": "REPLICATION_STOPPED",
            "participantId": "abCDEf12GHIj",
            "eventTime": "2025-04-26T20:32:34+00:00",
            "destinationStageArn": "arn:aws:ivs:us-west-2:12345678901:stage/
ABCDabcdefgh",
            "destinationSessionId": "st-b1c2d3e4f5g6a"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon IVS 低延遲串流使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListParticipantEvents。

`list-participant-replicas`

以下程式碼範例顯示如何使用 `list-participant-replicas`。

AWS CLI

取得階段參與者複本的清單

下列`list-participant-replicas`範例列出從指定的來源階段 ARN (Amazon Resource Name) 複寫到另一個階段的所有階段參與者。

```
aws ivs-realtime list-participant-replicas \
    --source-stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --participant-id abCDEf12GHIj
```

輸出：

```
{
    "replicas": [
        {
            "sourceStageArn": "arn:aws:ivs:us-west-2:123456789012:stage/
abcdABCDefgh",
            "participantId": "abCDEf12GHIj",
            "sourceSessionId": "st-a1b2c3d4e5f6g",
            "destinationStageArn": "arn:aws:ivs:us-west-2:012345678901:stage/
ABCDabcdefgh",
            "destinationSessionId": "st-b1c2d3e4f5g6a",
            "replicationState": "ACTIVE"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon IVS 即時串流使用者指南》中的 IVS 參與者複寫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListParticipantReplicas。

list-participants

以下程式碼範例顯示如何使用 list-participants。

AWS CLI

範例 1：取得階段參與者的清單

下列list-participants範例列出指定階段 ARN (Amazon Resource Name) 之指定工作階段 ID 的所有參與者。

```
aws ivs-realtime list-participants \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g
```

輸出：

```
{
    "participants": [
        {
            "firstJoinTime": "2023-04-26T20:30:34+00:00",
            "participantId": "abCDEf12GHIj",
            "published": true,
            "recordingState": "STOPPED",
            "state": "DISCONNECTED",
            "userId": ""
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon IVS 低延遲串流使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

範例 2：當參與者複寫至另一個階段時，取得階段參與者的清單

下列list-participants範例列出當參與者複寫到另一個階段時，指定階段 ARN (Amazon Resource Name) 之指定工作階段 ID 的所有參與者。

```
aws ivs-realtime list-participants \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g
```

輸出：

```
{
    "participants": [
        {
            "firstJoinTime": "2023-04-26T20:30:34+00:00",
            "participantId": "abCDEf12GHIj",
            "published": true,
            "recordingState": "STOPPED",
            "state": "DISCONNECTED",
            "userId": "",
            "replicationState": "ACTIVE",
            "replicationType": "SOURCE",
            "sourceStageArn": "",
            "sourceSessionId": ""
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon IVS 低延遲串流使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

範例 3：從另一個階段複寫參與者時，取得階段參與者的清單

下列`list-participants`範例列出從另一個階段複寫參與者時，指定階段 ARN (Amazon Resource Name) 之指定工作階段 ID 的所有參與者。

```
aws ivs-realtime list-participants \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g
```

輸出：

```
{
    "participants": [
        {
            "firstJoinTime": "2023-04-26T20:30:34+00:00",
            "participantId": "abCDEf12GHIj",
            "published": true,
            "recordingState": "STOPPED",
            "state": "DISCONNECTED",
            "userId": "",
```

```
            "replicationState": "ACTIVE",
            "replicationType": "REPLICA",
            "sourceStageArn": "arn:aws:ivs:us-west-2:123456789012:stage/
abcdABCDefgh",
            "sourceSessionId": "st-a1b2c3d4e5f6g"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon IVS 低延遲串流使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListParticipants。

`list-public-keys`

以下程式碼範例顯示如何使用 `list-public-keys`。

AWS CLI

列出可用於簽署階段參與者字符的現有公有金鑰

下列`list-public-keys`範例列出在處理 API 請求的 AWS 區域中，可用於簽署階段參與者字符的所有公有金鑰。

```
aws ivs-realtime list-public-keys
```

輸出：

```
{
    "publicKeys": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2",
            "name": "",
            "tags": {}
        },
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/3bcdABCDefg4",
            "name": "",
            "tags": {}
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon IVS 即時串流使用者指南》中的分發參與者字符。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPublicKeys。

`list-stage-sessions`

以下程式碼範例顯示如何使用 `list-stage-sessions`。

AWS CLI

取得階段工作階段的清單

下列`list-stage-sessions`範例列出指定階段 ARN (Amazon Resource Name) 的所有工作階段。

```
aws ivs-realtime list-stage-sessions \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

輸出：

```
{
    "stageSessions": [
        {
            "endTime": "2023-04-26T20:36:29+00:00",
            "sessionId": "st-a1b2c3d4e5f6g",
            "startTime": "2023-04-26T20:30:29.602000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListStageSessions。

`list-stages`

以下程式碼範例顯示如何使用 `list-stages`。

AWS CLI

取得所有階段的摘要資訊

下列list-stages範例列出處理 API 請求的 AWS 區域中，您 AWS 帳戶的所有階段。

```
aws ivs-realtime list-stages
```

輸出：

```
{
    "stages": [
        {
            "activeSessionId": "st-a1b2c3d4e5f6g",
            "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
            "name": "stage1",
            "tags": {}
        },
        {
            "activeSessionId": "st-a123bcd456efg",
            "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcd1234ABCD",
            "name": "stage2",
            "tags": {}
        },
        {
            "activeSessionId": "st-abcDEF1234ghi",
            "arn": "arn:aws:ivs:us-west-2:123456789012:stage/ABCD1234efgh",
            "name": "stage3",
            "tags": {}
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListStages。

list-storage-configurations

以下程式碼範例顯示如何使用 list-storage-configurations。

AWS CLI

列出合成儲存組態

以下`list-storage-configurations`列出處理 API 請求的 AWS 區域中，您 AWS 帳戶的所有合成儲存組態。

```
aws ivs-realtime list-storage-configurations
```

輸出：

```
{
    "storageConfigurations": [
        {
            "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
abcdABCDefgh",
            "name": "test-sc-1",
            "s3": {
                "bucketName": "amzn-s3-demo-bucket-1"
            },
            "tags": {}
        },
        {
            "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
ABCefgEFGabc",
            "name": "test-sc-2",
            "s3": {
                "bucketName": "amzn-s3-demo-bucket-2"
            },
            "tags": {}
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListStorageConfigurations。

`start-composition`

以下程式碼範例顯示如何使用 `start-composition`。

AWS CLI

範例 1：使用預設配置設定啟動合成

下列start-composition範例會啟動要串流至指定位置之指定階段的合成。

```
aws ivs-realtime start-composition \
    --stage-arn arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd \
    --destinations '[{"channel": {"channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg", \
        "encoderConfigurationArn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-
configuration/ABabCDcdEFef"}}, \
        {"s3":{"encoderConfigurationArns":["arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"], \
        "recordingConfiguration": {"hlsConfiguration":
 {"targetSegmentDurationSeconds": 5}}, \
        "storageConfigurationArn":"arn:aws:ivs:ap-northeast-1:123456789012:storage-
configuration/FefABabCDcdE"}}]'
```

輸出：

```
{
    "composition": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh",
        "destinations": [
            {
                "configuration": {
                    "channel": {
                        "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
                        "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                    },
                    "name": ""
                },
                "id": "AabBCcdDEefF",
                "state": "STARTING"
            },
            {
                "configuration": {
                    "name": "",
                    "s3": {
                        "encoderConfigurationArns": [
```

```
                            "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                        ],
                        "recordingConfiguration": {
                            "format": "HLS",
                            "hlsConfiguration": {
                                "targetSegmentDurationSeconds": 5
                            }
                        },
                        "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE"
                    }
                },
                "detail": {
                    "s3": {
                        "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
                    }
                },
                "id": "GHFabcgefABC",
                "state": "STARTING"
            }
        ],
        "layout": {
            "grid": {
                "featuredParticipantAttribute": ""
                "gridGap": 2,
                "omitStoppedVideo": false,
                "videoAspectRatio": "VIDEO",
                "videoFillMode": ""
            }
        },
        "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
        "startTime": "2023-10-16T23:24:00+00:00",
        "state": "STARTING",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon IVS 即時串流使用者指南》中的 IVS 複合錄製 | 即時串流。

範例 2：使用 PiP 配置啟動合成

下列`start-composition`範例會啟動指定階段的合成，以使用 PiP 配置串流至指定的位置。

```
aws ivs-realtime start-composition \
    --stage-arn arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd \
    --destinations '[{"channel": {"channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg", \
        "encoderConfigurationArn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-
configuration/ABabCDcdEFef"}}, \
        {"s3":{"encoderConfigurationArns":["arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"], \
        "storageConfigurationArn":"arn:aws:ivs:ap-northeast-1:123456789012:storage-
configuration/FefABabCDcdE"}}]' \
    --layout pip='{featuredParticipantAttribute="abcdefg"}'
```

輸出：

```
{
    "composition": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/wxyzWXYZpqrs",
        "destinations": [
            {
                "configuration": {
                    "channel": {
                        "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
                        "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                    },
                    "name": ""
                },
                "id": "AabBCcdDEefF",
                "state": "STARTING"
            },
            {
                "configuration": {
                    "name": "",
                    "s3": {
                        "encoderConfigurationArns": [
                            "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                        ],
                        "recordingConfiguration": {
                            "format": "HLS",
                            "hlsConfiguration": {
                                "targetSegmentDurationSeconds": 2
```

```
                            }
                        },
                        "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE"
                    }
                },
                "detail": {
                    "s3": {
                        "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
                    }
                },
                "id": "GHFabcgefABC",
                "state": "STARTING"
            }
        ],
        "layout": {
            "pip": {
                "featuredParticipantAttribute": "abcdefg",
                "gridGap": 0,
                "omitStoppedVideo": false,
                "pipBehavior": "STATIC",
                "pipOffset": 0,
                "pipParticipantAttribute": "",
                "pipPosition": "BOTTOM_RIGHT",
                "videoFillMode": "COVER"
            }
        },
        "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
        "startTime": "2023-10-16T23:24:00+00:00",
        "state": "STARTING",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon IVS 即時串流使用者指南》中的 IVS 複合錄製 | 即時串流。

範例 3：在啟用縮圖記錄的情況下啟動合成

下列`start-composition`範例會在啟用縮圖錄製的情況下，開始將指定階段的合成串流至指定的位置。

```
aws ivs-realtime start-composition \
```

```
    --stage-arn arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd \
    --destinations '[{"channel": {"channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg", \
        "encoderConfigurationArn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-
configuration/ABabCDcdEFef"}}, \
        {"s3": {"encoderConfigurationArns": ["arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"], \
        "storageConfigurationArn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-
configuration/FefABabCDcdE", \
        "thumbnailConfigurations": [{"storage":
 ["SEQUENTIAL"],"targetIntervalSeconds": 60}]}}]'
```

輸出：

```
{
    "composition": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh",
        "destinations": [
            {
                "configuration": {
                    "channel": {
                        "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
                        "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                    },
                    "name": ""
                },
                "id": "AabBCcdDEefF",
                "state": "STARTING"
            },
            {
                "configuration": {
                    "name": "",
                    "s3": {
                        "encoderConfigurationArns": [
                            "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                        ],
                        "recordingConfiguration": {
                            "format": "HLS",
                            "hlsConfiguration": {
                                "targetSegmentDurationSeconds": 2
```

```
                            }
                        },
                        "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE",
                        "thumbnailConfigurations": [
                           {
                              "targetIntervalSeconds": 60,
                              "storage": [
                                  "SEQUENTIAL"
                              ]
                           }
                        ]
                    }
                },
                "detail": {
                    "s3": {
                        "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
                    }
                },
                "id": "GHFabcgefABC",
                "state": "STARTING"
            }
        ],
        "layout": {
            "grid": {
                "featuredParticipantAttribute": ""
                "gridGap": 2,
                "omitStoppedVideo": false,
                "videoAspectRatio": "VIDEO",
                "videoFillMode": ""
            }
        },
        "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
        "startTime": "2023-10-16T23:24:00+00:00",
        "state": "STARTING",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon IVS 即時串流使用者指南》中的 IVS 複合錄製 | 即時串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartComposition。

start-participant-replication

以下程式碼範例顯示如何使用 start-participant-replication。

AWS CLI

開始將參與者從一個階段複寫到另一個階段

下列start-participant-replication範例會將參與者從來源階段複寫到目的地階段，每個階段由其 ARN (Amazon Resource Name) 指定。

```
aws ivs-realtime start-participant-replication \
    --source-stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --destination-stage-arn arn:aws:ivs:us-west-2:234567890123:stage/bcdABCDefghi \
    --participant-id abCDEf12GHIj
```

輸出：

```
{
    "accessControlAllowOrigin": "*",
    "accessControlExposeHeaders": "Access-Control-Allow-Origin,Access-Control-
Expose-Headers,Cache-Control,Content-Length, \
    Content-Security-Policy,Content-Type,date,Strict-Transport-Security,x-amz-apigw-
id,x-amzn-errormessage,x-amzn-errortype, \
    x-amzn-requestid,x-amzn-trace-id,X-Content-Type-Options,X-Frame-Options",
    "cacheControl": "no-store, no-cache",
    "contentSecurityPolicy": "default-src 'self'; upgrade-insecure-requests;",
    "strictTransportSecurity": "max-age:47304000; includeSubDomains",
    "xContentTypeOptions": "nosniff",
    "xFrameOptions": "DENY"
}
```

如需詳細資訊，請參閱《Amazon IVS 即時串流使用者指南》中的 IVS 參與者複寫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartParticipantReplication。

stop-composition

以下程式碼範例顯示如何使用 stop-composition。

AWS CLI

停止合成

以下內容會stop-composition停止指定 ARN (Amazon Resource Name) 指定的合成。

```
aws ivs-realtime stop-composition \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopComposition。

stop-participant-replication

以下程式碼範例顯示如何使用 stop-participant-replication。

AWS CLI

停止將參與者從一個階段複寫到另一個階段

下列stop-participant-replication範例會停止將參與者從來源階段複寫到目的地階段，每個階段由其 ARN (Amazon Resource Name) 指定。

```
aws ivs-realtime stop-participant-replication \
    --source-stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --destination-stage-arn arn:aws:ivs:us-west-2:234567890123:stage/bcdABCDefghi \
    --participant-id abCDEf12GHIj
```

輸出：

```
{
    "accessControlAllowOrigin": "*",
    "accessControlExposeHeaders": "Access-Control-Allow-Origin,Access-Control-
Expose-Headers,Cache-Control,Content-Length, \
    Content-Security-Policy,Content-Type,date,Strict-Transport-Security,x-amz-apigw-
id,x-amzn-errormessage,x-amzn-errortype, \
    x-amzn-requestid,x-amzn-trace-id,X-Content-Type-Options,X-Frame-Options",
    "cacheControl": "no-store, no-cache",
    "contentSecurityPolicy": "default-src 'self'; upgrade-insecure-requests;",
    "strictTransportSecurity": "max-age:47304000; includeSubDomains",
    "xContentTypeOptions": "nosniff",
    "xFrameOptions": "DENY"
```

```
}
```

如需詳細資訊，請參閱《Amazon IVS 即時串流使用者指南》中的 IVS 參與者複寫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopParticipantReplication。

update-ingest-configuration

以下程式碼範例顯示如何使用 update-ingest-configuration。

AWS CLI

更新擷取組態

下列update-inegst-configuration範例會更新擷取組態，以將其連接至階段。

```
aws ivs-realtime update-ingest-configuration \
    --arn arn:aws:ivs:us-west-2:123456789012:ingest-configuration/AbCdEfGh1234 \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

輸出：

```
{
    "ingestConfiguration": {
        "name": "ingest1",
        "arn": "arn:aws:ivs:us-west-2:123456789012:ingest-configuration/
AbCdEfGh1234",
        "ingestProtocol": "RTMPS",
        "streamKey": "rt_123456789012_us-
west-2_AbCdEfGh1234_abcd1234efgh5678ijkl9012MNOP34",
        "stageArn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "participantId": "xyZ654abC321",
        "state": "INACTIVE",
        "userId": "",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon Interactive Video Service 使用者指南》中的 IVS 串流擷取 | 即時串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateIngestConfiguration。

update-stage

以下程式碼範例顯示如何使用 update-stage。

AWS CLI

範例 1：更新階段的組態

下列update-stage範例會更新指定階段 ARN 的階段，以更新階段名稱，並在啟用縮圖錄製的情況下設定個別參與者錄製。

```
aws ivs-realtime update-stage \
    --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --auto-participant-recording-configuration '{"mediaTypes":
 ["AUDIO_VIDEO"],"storageConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:storage-configuration/abcdABCDefgh",
 "recordingReconnectWindowSeconds": 100, \
        "thumbnailConfiguration": {"recordingMode": "INTERVAL","storage":
 ["SEQUENTIAL"],"targetIntervalSeconds": 60}} \
        "hlsConfiguration": {"targetSegmentDurationSeconds": 5}}' \
    --name stage1a
```

輸出：

```
{
    "stage": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "autoParticipantRecordingConfiguration": {
            "mediaTypes": [
                "AUDIO_VIDEO"
            ],
            "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-
configuration/abcdABCDefgh",
            "thumbnailConfiguration": {
                "targetIntervalSeconds": 60,
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "INTERVAL"
            },
            "recordingReconnectWindowSeconds": 100,
            "hlsConfiguration": {
                "targetSegmentDurationSeconds": 5
```

```
            },
            "recordParticipantReplicas": true
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"
        },
        "name": "stage1a",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon IVS 低延遲串流使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

範例 2：更新階段的組態，包括停用參與者複本記錄

下列update-stage範例會更新指定階段 ARN 的階段，以更新階段名稱，並在啟用縮圖錄製和停用參與者複本錄製的情況下設定個別參與者錄製。

```
aws ivs-realtime update-stage \
    --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --auto-participant-recording-configuration '{"mediaTypes":
 ["AUDIO_VIDEO"],"storageConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:storage-configuration/abcdABCDefgh",
 "recordingReconnectWindowSeconds": 100, \
        "thumbnailConfiguration": {"recordingMode": "INTERVAL","storage":
 ["SEQUENTIAL"],"targetIntervalSeconds": 60}, "recordParticipantReplicas":false} \
        "hlsConfiguration": {"targetSegmentDurationSeconds": 5}}' \
    --name stage1a
```

輸出：

```
{
    "stage": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "autoParticipantRecordingConfiguration": {
            "mediaTypes": [
                "AUDIO_VIDEO"
```

```
            ],
            "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-
configuration/abcdABCDefgh",
            "thumbnailConfiguration": {
                "targetIntervalSeconds": 60,
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "INTERVAL"
            },
            "recordingReconnectWindowSeconds": 100,
            "hlsConfiguration": {
                "targetSegmentDurationSeconds": 5
            },
            "recordParticipantReplicas": false
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"
        },
        "name": "stage1a",
        "tags": {}
    }
}
```

如需詳細資訊，請參閱《Amazon IVS 低延遲串流使用者指南》中的在 Amazon IVS 串流上啟用多個主機。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateStage。

使用 的 Amazon Kendra 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon Kendra 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

create-data-source

以下程式碼範例顯示如何使用 create-data-source。

AWS CLI

建立 Amazon Kendra 資料來源連接器

以下內容會create-data-source建立和設定 Amazon Kendra 資料來源連接器。您可以使用 describe-data-source 檢視資料來源連接器的狀態，如果狀態顯示要完整建立的資料來源連接器「失敗」，則讀取任何錯誤訊息。

```
aws kendra create-data-source \
    --name "example data source 1" \
    --description "Example data source 1 for example index 1 contains the first set
 of example documents" \
    --tags '{"Key": "test resources", "Value": "kendra"}, {"Key": "test resources",
 "Value": "aws"}' \
    --role-arn "arn:aws:iam::my-account-id:role/
KendraRoleForS3TemplateConfigDataSource" \
    --index-id exampleindex1 \
    --language-code "es" \
    --schedule "0 0 18 ? * TUE,MON,WED,THU,FRI,SAT *" \
    --configuration '{"TemplateConfiguration": {"Template": file://
s3schemaconfig.json}}' \
    --type "TEMPLATE" \
    --custom-document-enrichment-configuration '{"PostExtractionHookConfiguration":
 {"LambdaArn": "arn:aws:iam::my-account-id:function/my-function-ocr-docs",
 "S3Bucket": "s3://amzn-s3-demo-bucket/scanned-image-text-example-docs"}, "RoleArn":
 "arn:aws:iam:my-account-id:role/KendraRoleForCDE"}' \
    --vpc-configuration '{"SecurityGroupIds": ["sg-1234567890abcdef0"], "SubnetIds":
 ["subnet-1c234","subnet-2b134"]}'
```

輸出：

```
{
```

```
    "Id": "exampledatasource1"
}
```

如需詳細資訊，請參閱《Amazon Kendra 開發人員指南》中的 Amazon Kendra 索引和資料來源連接器入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDataSource。

`create-index`

以下程式碼範例顯示如何使用 `create-index`。

AWS CLI

建立 Amazon Kendra 索引

以下內容會`create-index`建立和設定 Amazon Kendra 索引。您可以使用 `describe-index` 檢視索引的狀態，如果狀態顯示要完整建立的索引「失敗」，則可以讀取任何錯誤訊息。

```
aws kendra create-index \
    --name "example index 1" \
    --description "Example index 1 contains the first set of example documents" \
    --tags '{"Key": "test resources", "Value": "kendra"}, {"Key": "test resources",
 "Value": "aws"}' \
    --role-arn "arn:aws:iam::my-account-id:role/KendraRoleForExampleIndex" \
    --edition "DEVELOPER_EDITION" \
    --server-side-encryption-configuration '{"KmsKeyId": "my-kms-key-id"}' \
    --user-context-policy "USER_TOKEN" \
    --user-token-configurations '{"JsonTokenTypeConfiguration":
 {"GroupAttributeField": "groupNameField", "UserNameAttributeField":
 "userNameField"}}'
```

輸出：

```
{
   "Id": index1
}
```

如需詳細資訊，請參閱《Amazon Kendra 開發人員指南》中的 Amazon Kendra 索引和資料來源連接器入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateIndex。

describe-data-source

以下程式碼範例顯示如何使用 describe-data-source。

AWS CLI

取得 Amazon Kendra 資料來源連接器的相關資訊

以下內容describe-data-source會取得有關 Amazon Kendra Data Soource 連接器的資訊。您可以檢視資料來源連接器的組態，如果狀態顯示要完整建立的資料來源連接器「FAILED」，您可以讀取任何錯誤訊息。

```
aws kendra describe-data-source \
    --id exampledatasource1 \
    --index-id exampleindex1
```

輸出：

```
{
    "Configuration": {
        "TemplateConfiguration": {
            "Template": {
                "connectionConfiguration": {
                    "repositoryEndpointMetadata": {
                        "BucketName": "amzn-s3-demo-bucket"
                    }
                },
                "repositoryConfigurations": {
                    "document":{
                        "fieldMappings": [
                            {
                                "indexFieldName":"_document_title",
                                "indexFieldType":"STRING",
                                "dataSourceFieldName": "title"
                            },
                            {
                                "indexFieldName":"_last_updated_at",
                                "indexFieldType":"DATE",
                                "dataSourceFieldName": "modified_date"
                            }
                        ]
                    }
                },
```

```
                "additionalProperties": {
                    "inclusionPatterns": [
                        "*.txt",
                        "*.doc",
                        "*.docx"
                    ],
                    "exclusionPatterns": [
                        "*.json"
                    ],
                    "inclusionPrefixes": [
                        "PublicExampleDocsFolder"
                    ],
                    "exclusionPrefixes": [
                        "PrivateDocsFolder/private"
                    ],
                    "aclConfigurationFilePath": "ExampleDocsFolder/AclConfig.json",
                    "metadataFilesPrefix": "metadata"
                },
                "syncMode": "FULL_CRAWL",
                "type" : "S3",
                "version": "1.0.0"
            }
        }
    },
    "CreatedAt": 2024-02-25T13:30:10+00:00,
    "CustomDocumentEnrichmentConfiguration": {
        "PostExtractionHookConfiguration": {
            "LambdaArn": "arn:aws:iam::my-account-id:function/my-function-ocr-docs",
            "S3Bucket": "s3://amzn-s3-demo-bucket/scanned-image-text-example-docs/
function"
        },
        "RoleArn": "arn:aws:iam:my-account-id:role/KendraRoleForCDE"
    }
    "Description": "Example data source 1 for example index 1 contains the first set
 of example documents",
    "Id": exampledatasource1,
    "IndexId": exampleindex1,
    "LanguageCode": "en",
    "Name": "example data source 1",
    "RoleArn": "arn:aws:iam::my-account-id:role/
KendraRoleForS3TemplateConfigDataSource",
    "Schedule": "0 0 18 ? * TUE,MON,WED,THU,FRI,SAT *",
    "Status": "ACTIVE",
    "Type": "TEMPLATE",
```

```
    "UpdatedAt": 1709163615,
    "VpcConfiguration": {
        "SecurityGroupIds": ["sg-1234567890abcdef0"],
        "SubnetIds": ["subnet-1c234","subnet-2b134"]
    }
}
```

如需詳細資訊，請參閱《Amazon Kendra 開發人員指南》中的 Amazon Kendra 索引和資料來源連接器入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDataSource。

describe-index

以下程式碼範例顯示如何使用 describe-index。

AWS CLI

取得 Amazon Kendra 索引的相關資訊

以下內容describe-index會取得 Amazon Kendra 索引的相關資訊。您可以檢視索引的組態，如果狀態顯示要完全建立的索引「失敗」，您可以讀取任何錯誤訊息。

```
aws kendra describe-index \
    --id exampleindex1
```

輸出：

```
{
    "CapacityUnits": {
        "QueryCapacityUnits": 0,
        "StorageCapacityUnits": 0
    },
    "CreatedAt": 2024-02-25T12:30:10+00:00,
    "Description": "Example index 1 contains the first set of example documents",
    "DocumentMetadataConfigurations": [
        {
            "Name": "_document_title",
            "Relevance": {
                "Importance": 8
            },
            "Search": {
                "Displayable": true,
```

```
                "Facetable": false,
                "Searchable": true,
                "Sortable": false
            },
            "Type": "STRING_VALUE"
        },
        {
            "Name": "_document_body",
            "Relevance": {
                "Importance": 5
            },
            "Search": {
                "Displayable": true,
                "Facetable": false,
                "Searchable": true,
                "Sortable": false
            },
            "Type": "STRING_VALUE"
        },
        {
            "Name": "_last_updated_at",
            "Relevance": {
                "Importance": 6,
                "Duration": "2628000s",
                "Freshness": true
            },
            "Search": {
                "Displayable": true,
                "Facetable": false,
                "Searchable": true,
                "Sortable": true
            },
            "Type": "DATE_VALUE"
        },
        {
            "Name": "department_custom_field",
            "Relevance": {
                "Importance": 7,
                "ValueImportanceMap": {
                    "Human Resources" : 4,
                    "Marketing and Sales" : 2,
                    "Research and innvoation" : 3,
                    "Admin" : 1
                }
```

```
            },
            "Search": {
                "Displayable": true,
                "Facetable": true,
                "Searchable": true,
                "Sortable": true
            },
            "Type": "STRING_VALUE"
        }
    ],
    "Edition": "DEVELOPER_EDITION",
    "Id": "index1",
    "IndexStatistics": {
        "FaqStatistics": {
            "IndexedQuestionAnswersCount": 10
        },
        "TextDocumentStatistics": {
            "IndexedTextBytes": 1073741824,
            "IndexedTextDocumentsCount": 1200
        }
    },
    "Name": "example index 1",
    "RoleArn": "arn:aws:iam::my-account-id:role/KendraRoleForExampleIndex",
    "ServerSideEncryptionConfiguration": {
        "KmsKeyId": "my-kms-key-id"
    },
    "Status": "ACTIVE",
    "UpdatedAt": 1709163615,
    "UserContextPolicy": "USER_TOKEN",
    "UserTokenConfigurations": [
        {
            "JsonTokenTypeConfiguration": {
                "GroupAttributeField": "groupNameField",
                "UserNameAttributeField": "userNameField"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Kendra 開發人員指南》中的 Amazon Kendra 索引和資料來源連接器入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeIndex。

update-data-source

以下程式碼範例顯示如何使用 update-data-source。

AWS CLI

更新 Amazon Kendra 資料來源連接器

以下內容會update-data-source更新 Amazon Kendra 資料來源連接器的組態。如果動作成功，則服務不會傳回任何輸出、HTTP 狀態碼 200 或 AWS CLI 傳回碼 0。您可以使用 describe-data-source來檢視資料來源連接器的組態和狀態。

```
aws kendra update-data-source \
    --id exampledatasource1 \
    --index-id exampleindex1 \
    --name "new name for example data source 1" \
    --description "new description for example data source 1" \
    --role-arn arn:aws:iam::my-account-id:role/KendraNewRoleForExampleDataSource \
    --configuration '{"TemplateConfiguration": {"Template": file://
s3schemanewconfig.json}}' \
    --custom-document-enrichment-configuration '{"PostExtractionHookConfiguration":
 {"LambdaArn": "arn:aws:iam::my-account-id:function/my-function-ocr-docs",
 "S3Bucket": "s3://amzn-s3-demo-bucket/scanned-image-text-example-docs"}, "RoleArn":
 "arn:aws:iam:my-account-id:role/KendraNewRoleForCDE"}' \
    --language-code "es" \
    --schedule "0 0 18 ? * MON,WED,FRI *" \
    --vpc-configuration '{"SecurityGroupIds": ["sg-1234567890abcdef0"], "SubnetIds":
 ["subnet-1c234","subnet-2b134"]}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Kendra 開發人員指南》中的 Amazon Kendra 索引和資料來源連接器入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDataSource。

update-index

以下程式碼範例顯示如何使用 update-index。

AWS CLI

更新 Amazon Kendra 索引

以下內容會update-index更新 Amazon Kendra 索引的組態。如果動作成功，則服務不會傳回任何輸出、HTTP 狀態碼 200 或 AWS CLI 傳回碼 0。您可以使用 describe-index來檢視索引的組態和狀態。

```
aws kendra update-index \
    --id enterpriseindex1 \
    --name "new name for Enterprise Edition index 1" \
    --description "new description for Enterprise Edition index 1" \
    --role-arn arn:aws:iam::my-account-id:role/KendraNewRoleForEnterpriseIndex \
    --capacity-units '{"QueryCapacityUnits": 2, "StorageCapacityUnits": 1}' \
    --document-metadata-configuration-updates '{"Name": "_document_title",
 "Relevance": {"Importance": 6}}, {"Name": "_last_updated_at", "Relevance":
 {"Importance": 8}}' \
    --user-context-policy "USER_TOKEN" \
    --user-token-configurations '{"JsonTokenTypeConfiguration":
 {"GroupAttributeField": "groupNameField", "UserNameAttributeField":
 "userNameField"}}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Kendra 開發人員指南》中的 Amazon Kendra 索引和資料來源連接器入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateIndex。

使用 的 Kinesis 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Kinesis 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`add-tags-to-stream`

以下程式碼範例顯示如何使用 `add-tags-to-stream`。

AWS CLI

將標籤新增至資料串流

下列`add-tags-to-stream`範例會將具有金鑰`samplekey`和值`example`的標籤指派給指定的串流。

```
aws kinesis add-tags-to-stream \
    --stream-name samplestream \
    --tags samplekey=example
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的標記您的串流。Amazon Kinesis

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddTagsToStream。

`create-stream`

以下程式碼範例顯示如何使用 `create-stream`。

AWS CLI

建立資料串流

下列`create-stream`範例會建立名為 samplestream 的資料串流，其中包含 3 個碎片。

```
aws kinesis create-stream \
    --stream-name samplestream \
    --shard-count 3
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的建立串流。 Amazon Kinesis

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateStream。

decrease-stream-retention-period

以下程式碼範例顯示如何使用 decrease-stream-retention-period。

AWS CLI

減少資料串流保留期間

下列decrease-stream-retention-period範例會將名為 samplestream 之串流的保留期間（資料記錄新增至串流後可存取的時間長度）縮短為 48 小時。

```
aws kinesis decrease-stream-retention-period \
    --stream-name samplestream \
    --retention-period-hours 48
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的變更資料保留期間。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DecreaseStreamRetentionPeriod。

delete-stream

以下程式碼範例顯示如何使用 delete-stream。

AWS CLI

刪除資料串流

下列delete-stream範例會刪除指定的資料串流。

```
aws kinesis delete-stream \
    --stream-name samplestream
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的刪除串流。 Amazon Kinesis

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteStream。

deregister-stream-consumer

以下程式碼範例顯示如何使用 deregister-stream-consumer。

AWS CLI

取消註冊資料串流取用者

下列deregister-stream-consumer範例會從指定的資料串流取消註冊指定的取用者。

```
aws kinesis deregister-stream-consumer \
    --stream-arn arn:aws:kinesis:us-west-2:123456789012:stream/samplestream \
    --consumer-name KinesisConsumerApplication
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Kinesis Kinesis Data Streams 開發人員指南》中的使用 Kinesis Data Streams API 開發具有增強廣發功能的消費者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterStreamConsumer。

describe-limits

以下程式碼範例顯示如何使用 describe-limits。

AWS CLI

描述碎片限制

下列describe-limits範例顯示目前 AWS 帳戶的碎片限制和用量。

```
aws kinesis describe-limits
```

輸出：

```
{
    "ShardLimit": 500,
    "OpenShardCount": 29
```

```
}
```

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的重新分片串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLimits。

describe-stream-consumer

以下程式碼範例顯示如何使用 describe-stream-consumer。

AWS CLI

描述資料串流取用者

下列describe-stream-consumer範例會傳回使用指定資料串流註冊之指定取用者的描述。

```
aws kinesis describe-stream-consumer \
    --stream-arn arn:aws:kinesis:us-west-2:012345678912:stream/samplestream \
    --consumer-name KinesisConsumerApplication
```

輸出：

```
{
    "ConsumerDescription": {
        "ConsumerName": "KinesisConsumerApplication",
        "ConsumerARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream/
consumer/KinesisConsumerApplication:1572383852",
        "ConsumerStatus": "ACTIVE",
        "ConsumerCreationTimestamp": 1572383852.0,
        "StreamARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream"
    }
}
```

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的從 Amazon Kinesis Data Streams 讀取資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStreamConsumer。

describe-stream-summary

以下程式碼範例顯示如何使用 describe-stream-summary。

AWS CLI

描述資料串流摘要

下列describe-stream-summary範例提供指定資料串流的摘要描述（不含碎片清單）。

```
aws kinesis describe-stream-summary \
    --stream-name samplestream
```

輸出：

```
{
    "StreamDescriptionSummary": {
        "StreamName": "samplestream",
        "StreamARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream",
        "StreamStatus": "ACTIVE",
        "RetentionPeriodHours": 48,
        "StreamCreationTimestamp": 1572297168.0,
        "EnhancedMonitoring": [
            {
                "ShardLevelMetrics": []
            }
        ],
        "EncryptionType": "NONE",
        "OpenShardCount": 3,
        "ConsumerCount": 0
    }
}
```

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的建立和管理串流。Amazon Kinesis

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStreamSummary。

describe-stream

以下程式碼範例顯示如何使用 describe-stream。

AWS CLI

描述資料串流

下列describe-stream範例會傳回指定資料串流的詳細資訊。

```
aws kinesis describe-stream \
    --stream-name samplestream
```

輸出：

```
{
    "StreamDescription": {
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
                    "StartingHashKey": "0",
                    "EndingHashKey": "113427455640312821154458202477256070484"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
 "496008716829570364423650249261910734372510605801286533 14"
                }
            },
            {
                "ShardId": "shardId-000000000001",
                "HashKeyRange": {
                    "StartingHashKey": "113427455640312821154458202477256070485",
                    "EndingHashKey": "226854911280625642308916404954512140969"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
 "496008716829793371875635555493326091555237089416346337 46"
                }
            },
            {
                "ShardId": "shardId-000000000002",
                "HashKeyRange": {
                    "StartingHashKey": "226854911280625642308916404954512140970",
                    "EndingHashKey": "340282366920938463463374607431768211455"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
 "496008716830016379327620861724741448737963573031406141 78"
                }
            }
```

```
        ],
        "StreamARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream",
        "StreamName": "samplestream",
        "StreamStatus": "ACTIVE",
        "RetentionPeriodHours": 24,
        "EnhancedMonitoring": [
            {
                "ShardLevelMetrics": []
            }
        ],
        "EncryptionType": "NONE",
        "KeyId": null,
        "StreamCreationTimestamp": 1572297168.0
    }
}
```

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的建立和管理串流。Amazon Kinesis

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStream。

disable-enhanced-monitoring

以下程式碼範例顯示如何使用 disable-enhanced-monitoring。

AWS CLI

停用碎片層級指標的增強型監控

下列disable-enhanced-monitoring範例會停用碎片層級指標的增強型 Kinesis 資料串流監控。

```
aws kinesis disable-enhanced-monitoring \
    --stream-name samplestream --shard-level-metrics ALL
```

輸出：

```
{
    "StreamName": "samplestream",
    "CurrentShardLevelMetrics": [
        "IncomingBytes",
        "OutgoingRecords",
```

```
        "IteratorAgeMilliseconds",
        "IncomingRecords",
        "ReadProvisionedThroughputExceeded",
        "WriteProvisionedThroughputExceeded",
        "OutgoingBytes"
    ],
    "DesiredShardLevelMetrics": []
}
```

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的監控 Amazon Kinesis Data Streams 中的串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableEnhancedMonitoring。

enable-enhanced-monitoring

以下程式碼範例顯示如何使用 enable-enhanced-monitoring。

AWS CLI

啟用碎片層級指標的增強型監控

下列enable-enhanced-monitoring範例會啟用碎片層級指標的增強型 Kinesis 資料串流監控。

```
aws kinesis enable-enhanced-monitoring \
    --stream-name samplestream \
    --shard-level-metrics ALL
```

輸出：

```
{
    "StreamName": "samplestream",
    "CurrentShardLevelMetrics": [],
    "DesiredShardLevelMetrics": [
        "IncomingBytes",
        "OutgoingRecords",
        "IteratorAgeMilliseconds",
        "IncomingRecords",
        "ReadProvisionedThroughputExceeded",
        "WriteProvisionedThroughputExceeded",
        "OutgoingBytes"
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的監控 Amazon Kinesis Data Streams 中的串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableEnhancedMonitoring。

get-records

以下程式碼範例顯示如何使用 get-records。

AWS CLI

從碎片取得記錄

下列get-records範例使用指定的碎片迭代器，從 Kinesis 資料串流的碎片取得資料記錄。

```
aws kinesis get-records \
    --shard-iterator AAAAAAAAAAF7/0mWD7IuHj1yGv/
TKuNgx2ukD5xipCY4cy4gU96orWwZwcSXh3K9tAmGYeOZyLZrvzzeOFVf9iN99hUPw/w/
b0YWYeehfNvnf1DYt5XpDJghLKr3DzgznkTmMymDP3R+3wRKeuEw6/kdxY2yKJH0veaiekaVc4N2VwK/
GvaGP2Hh9Fg7N++q0Adg6fIDQPt4p8RpavDbk+A4sL9SWGE1
```

輸出：

```
{
    "Records": [],
    "MillisBehindLatest": 80742000
}
```

如需詳細資訊，請參閱《Amazon Kinesis Kinesis Data Streams 開發人員指南》中的使用 Kinesis Data Streams API 搭配適用於 Java 的 AWS SDK 開發消費者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRecords。

get-shard-iterator

以下程式碼範例顯示如何使用 get-shard-iterator。

AWS CLI

取得碎片迭代器

下列get-shard-iterator範例使用AT_SEQUENCE_NUMBER碎片疊代運算類型，並產生碎片疊代運算，從以指定序號表示的位置開始讀取資料記錄。

```
aws kinesis get-shard-iterator \
    --stream-name samplestream \
    --shard-id shardId-000000000001 \
    --shard-iterator-type LATEST
```

輸出：

```
{
    "ShardIterator": "AAAAAAAAAAFEvJjIYI+3jw/4aqgH9FifJ+n48XWTh/
IFIsbILP6o5eDueD39NXNBfpZ10WL5K6ADXk8w+5H+Qhd9cFA9k268CPXCz/kebq1TGYI7Vy
+lUkA9BuN3xvATxMBGxRY3zYK05gqgvaIRn94O8SqeEqwhigwZxNWxID3Ej7YYYcxQi8Q/fIrCjGAy/
n2r5Z9G864YpWDfN9upNNQAR/iiOWKs"
}
```

如需詳細資訊，請參閱《Amazon Kinesis Kinesis Data Streams 開發人員指南》中的使用 Kinesis Data Streams API 搭配適用於 Java 的 AWS SDK 開發消費者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetShardIterator。

increase-stream-retention-period

以下程式碼範例顯示如何使用 increase-stream-retention-period。

AWS CLI

增加資料串流保留期

下列increase-stream-retention-period範例會將指定串流的保留期間（資料記錄新增至串流後可存取的時間長度）增加至 168 小時。

```
aws kinesis increase-stream-retention-period \
    --stream-name samplestream \
    --retention-period-hours 168
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的變更資料保留期間。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 IncreaseStreamRetentionPeriod。

list-shards

以下程式碼範例顯示如何使用 list-shards。

AWS CLI

列出資料串流中的碎片

下列list-shards範例會列出指定串流中的所有碎片，從其 ID 緊接在指定的 之後exclusive-start-shard-id的碎片開始shardId-000000000000。

```
aws kinesis list-shards \
    --stream-name samplestream \
    --exclusive-start-shard-id shardId-000000000000
```

輸出：

```
{
    "Shards": [
        {
            "ShardId": "shardId-000000000001",
            "HashKeyRange": {
                "StartingHashKey": "113427455640312821154458202477256070485",
                "EndingHashKey": "226854911280625642308916404954512140969"
            },
            "SequenceNumberRange": {
                "StartingSequenceNumber":
 "496008716829793371875635555493326091555237089416346337 46"
            }
        },
        {
            "ShardId": "shardId-000000000002",
            "HashKeyRange": {
                "StartingHashKey": "226854911280625642308916404954512140970",
                "EndingHashKey": "340282366920938463463374607431768211455"
            },
            "SequenceNumberRange": {
                "StartingSequenceNumber":
 "496008716830016379327620861724741448737963573031406141 78"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的列出碎片。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListShards。

`list-streams`

以下程式碼範例顯示如何使用 `list-streams`。

AWS CLI

列出資料串流

下列`list-streams`範例列出目前帳戶和區域中所有作用中的資料串流。

```
aws kinesis list-streams
```

輸出：

```
{
    "StreamNames": [
        "samplestream",
        "samplestream1"
    ]
}
```

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的列出串流。 Amazon Kinesis

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListStreams。

`list-tags-for-stream`

以下程式碼範例顯示如何使用 `list-tags-for-stream`。

AWS CLI

列出資料串流的標籤

下列`list-tags-for-stream`範例列出連接至指定資料串流的標籤。

```
aws kinesis list-tags-for-stream \
    --stream-name samplestream
```

輸出：

```
{
    "Tags": [
        {
            "Key": "samplekey",
            "Value": "example"
        }
    ],
    "HasMoreTags": false
}
```

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的標記您的串流。Amazon Kinesis

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTagsForStream。

merge-shards

以下程式碼範例顯示如何使用 merge-shards。

AWS CLI

合併碎片

下列merge-shards範例會在指定的資料串流中合併兩個 ID IDs 為 shardId-0000000000000 和 shardId-000000000001 的相鄰碎片，並將其合併為單一碎片。

```
aws kinesis merge-shards \
    --stream-name samplestream \
    --shard-to-merge shardId-000000000000 \
    --adjacent-shard-to-merge shardId-000000000001
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的合併兩個碎片。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 MergeShards。

put-record

以下程式碼範例顯示如何使用 put-record。

AWS CLI

將記錄寫入資料串流

下列put-record範例會使用指定的分割區索引鍵，將單一資料記錄寫入指定的資料串流。

```
aws kinesis put-record \
    --stream-name samplestream \
    --data sampledatarecord \
    --partition-key samplepartitionkey
```

輸出：

```
{
    "ShardId": "shardId-000000000009",
    "SequenceNumber": "49600902273357540915989931256901506243878407835297513618",
    "EncryptionType": "KMS"
}
```

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的使用 Amazon Kinesis Data Streams API 搭配適用於 Java 的 AWS SDK 開發生產者。 Amazon Kinesis

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutRecord。

put-records

以下程式碼範例顯示如何使用 put-records。

AWS CLI

將多筆記錄寫入資料串流

下列put-records範例使用指定的分割區索引鍵寫入資料記錄，並在單一呼叫中使用不同的分割區索引鍵寫入另一個資料記錄。

```
aws kinesis put-records \
    --stream-name samplestream \
    --
records Data=blob1,PartitionKey=partitionkey1 Data=blob2,PartitionKey=partitionkey2
```

輸出：

```
{
    "FailedRecordCount": 0,
    "Records": [
        {
            "SequenceNumber":
 "49600883331171471519674795588238531498465399900093808706",
            "ShardId": "shardId-000000000004"
        },
        {
            "SequenceNumber":
 "49600902273357540915989931256902715169698037101720764562",
            "ShardId": "shardId-000000000009"
        }
    ],
    "EncryptionType": "KMS"
}
```

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的使用 Amazon Kinesis Data Streams API 搭配適用於 Java 的 AWS SDK 開發生產者。 Amazon Kinesis

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutRecords。

register-stream-consumer

以下程式碼範例顯示如何使用 register-stream-consumer。

AWS CLI

註冊資料串流取用者

下列register-stream-consumer範例會KinesisConsumerApplication向指定的資料串流註冊呼叫的取用者。

```
aws kinesis register-stream-consumer \
    --stream-arn arn:aws:kinesis:us-west-2:012345678912:stream/samplestream \
    --consumer-name KinesisConsumerApplication
```

輸出：

```
{
    "Consumer": {
```

```
        "ConsumerName": "KinesisConsumerApplication",
        "ConsumerARN": "arn:aws:kinesis:us-west-2: 123456789012:stream/samplestream/
consumer/KinesisConsumerApplication:1572383852",
        "ConsumerStatus": "CREATING",
        "ConsumerCreationTimestamp": 1572383852.0
    }
}
```

如需詳細資訊，請參閱《Amazon Kinesis Kinesis Data Streams 開發人員指南》中的使用 Kinesis Data Streams API 開發具有增強廣發功能的消費者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterStreamConsumer。

remove-tags-from-stream

以下程式碼範例顯示如何使用 remove-tags-from-stream。

AWS CLI

從資料串流移除標籤

下列remove-tags-from-stream範例會從指定的資料串流移除具有指定金鑰的標籤。

```
aws kinesis remove-tags-from-stream \
    --stream-name samplestream \
    --tag-keys samplekey
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的標記您的串流。Amazon Kinesis

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveTagsFromStream。

split-shard

以下程式碼範例顯示如何使用 split-shard。

AWS CLI

分割碎片

下列split-shard範例使用新的開始雜湊索引鍵 10，將指定的碎片分割成兩個新的碎片。

```
aws kinesis split-shard \
    --stream-name samplestream \
    --shard-to-split shardId-000000000000 \
    --new-starting-hash-key 10
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的分割碎片。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SplitShard。

start-stream-encryption

以下程式碼範例顯示如何使用 start-stream-encryption。

AWS CLI

啟用資料串流加密

下列start-stream-encryption範例使用指定的 AWS KMS 金鑰，為指定的串流啟用伺服器端加密。

```
aws kinesis start-stream-encryption \
    --encryption-type KMS \
    --key-id arn:aws:kms:us-west-2:012345678912:key/a3c4a7cd-728b-45dd-
b334-4d3eb496e452 \
    --stream-name samplestream
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的 Amazon Kinesis Data Streams 中的資料保護。 Amazon Kinesis

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartStreamEncryption。

stop-stream-encryption

以下程式碼範例顯示如何使用 stop-stream-encryption。

AWS CLI

停用資料串流加密

下列stop-stream-encryption範例使用指定的 AWS KMS 金鑰，停用指定串流的伺服器端加密。

```
aws kinesis start-stream-encryption \
    --encryption-type KMS \
    --key-id arn:aws:kms:us-west-2:012345678912:key/a3c4a7cd-728b-45dd-b334-4d3eb496e452 \
    --stream-name samplestream
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的 Amazon Kinesis Data Streams 中的資料保護。 Amazon Kinesis

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopStreamEncryption。

update-shard-count

以下程式碼範例顯示如何使用 update-shard-count。

AWS CLI

更新資料串流中的碎片計數

下列update-shard-count範例會將指定資料串流的碎片計數更新為 6。此範例使用統一擴展，這會建立大小相等的碎片。

```
aws kinesis update-shard-count \
    --stream-name samplestream \
    --scaling-type UNIFORM_SCALING \
    --target-shard-count 6
```

輸出：

```
{
    "StreamName": "samplestream",
    "CurrentShardCount": 3,
```

```
    "TargetShardCount": 6
}
```

如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的重新分片串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateShardCount。

AWS KMS 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS KMS。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`cancel-key-deletion`

以下程式碼範例顯示如何使用 `cancel-key-deletion`。

AWS CLI

取消客戶受管 KMS 金鑰的排程刪除

下列`cancel-key-deletion`範例會取消客戶受管 KMS 金鑰的排程刪除。

```
aws kms cancel-key-deletion \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

輸出：

```
{
    "KeyId": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
```

```
}
```

當cancel-key-deletion命令成功時，會取消排定的刪除。不過，KMS 金鑰的金鑰狀態為 Disabled，因此您無法在密碼編譯操作中使用 KMS 金鑰。若要還原其功能，請使用enable-key命令 。

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的排程和取消金鑰刪除。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelKeyDeletion。

connect-custom-key-store

以下程式碼範例顯示如何使用 connect-custom-key-store。

AWS CLI

連接自訂金鑰存放區

下列connect-custom-key-store範例會重新連接指定的自訂金鑰存放區。您可以使用像這樣的命令來第一次連接自訂金鑰存放區，或重新連接中斷連接的金鑰存放區。

您可以使用此命令來連接 AWS CloudHSM 金鑰存放區或外部金鑰存放區。

```
aws kms connect-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0
```

此命令不會傳回任何輸出。若要驗證命令是否有效，請使用 describe-custom-key-stores命令。

如需有關連接 AWS CloudHSM 金鑰存放區的資訊，請參閱 AWS Key Management Service 開發人員指南中的連接和中斷連接 an AWS CloudHSM 金鑰存放區。

如需有關連接外部金鑰存放區的資訊，請參閱 Key Management Service 開發人員指南中的連接和中斷連接外部金鑰存放區。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ConnectCustomKeyStore。

create-alias

以下程式碼範例顯示如何使用 create-alias。

AWS CLI

建立 KMS 金鑰的別名

下列create-alias命令會為金鑰 ID 所識別example-alias的 KMS 金鑰建立名為 的別名1234abcd-12ab-34cd-56ef-1234567890ab。

別名名稱必須以 開頭alias/。請勿使用以 開頭的別名；alias/aws這些名稱會保留供 使用 AWS。

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

此命令不會傳回任何輸出。若要查看新的別名，請使用 list-aliases命令。

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的使用別名。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateAlias。

create-custom-key-store

以下程式碼範例顯示如何使用 create-custom-key-store。

AWS CLI

範例 1：建立 an AWS CloudHSM 金鑰存放區

下列create-custom-key-store範例會使用所需的參數建立由 AWS CloudHSM 叢集支援的 AWS CloudHSM 金鑰存放區。您也可以新增 custom-key-store-type``parameter with the default value: ``AWS_CLOUDHSM。

若要在 CLI AWS 中指定trust-anchor-certificate命令的檔案輸入，需要 file://字首。

```
aws kms create-custom-key-store \
    --custom-key-store-name ExampleCloudHSMKeyStore \
    --cloud-hsm-cluster-id cluster-1a23b4cdefg \
    --key-store-password kmsPswd \
    --trust-anchor-certificate file://customerCA.crt
```

輸出：

```
{
    "CustomKeyStoreId": cks-1234567890abcdef0
}
```

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的建立 an AWS CloudHSM 金鑰存放區。 AWS

範例 2：建立具有公有端點連線能力的外部金鑰存放區

下列create-custom-key-store範例會建立透過網際網路與 AWS KMS 通訊的外部金鑰存放區 (XKS)。

在此範例中， XksProxyUriPath使用選用的 字首example-prefix。

注意：如果您使用 AWS CLI 1.0 版，請在指定具有 HTTP 或 HTTPS 值的參數之前執行下列命令，例如 XksProxyUriEndpoint 參數。

```
aws configure set cli_follow_urlparam false
```

否則， AWS CLI 1.0 版會將 參數值取代為在該 URI 地址找到的內容。

```
aws kms create-custom-key-store \
    --custom-key-store-name ExamplePublicEndpointXKS \
    --custom-key-store-type EXTERNAL_KEY_STORE \
    --xks-proxy-connectivity PUBLIC_ENDPOINT \
    --xks-proxy-uri-endpoint "https://myproxy.xks.example.com" \
    --xks-proxy-uri-path "/example-prefix/kms/xks/v1" \
    --xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,
 RawSecretAccessKey=DXjSUawnel2fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

輸出：

```
{
    "CustomKeyStoreId": cks-2234567890abcdef0
}
```

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的建立外部金鑰存放區。 AWS

範例 3：建立具有 VPC 端點服務連線的外部金鑰存放區

下列create-custom-key-store範例會建立使用 Amazon VPC 端點服務與 AWS KMS 通訊的外部金鑰存放區 (XKS)。

注意：如果您使用 AWS CLI 1.0 版，請在指定具有 HTTP 或 HTTPS 值的參數之前執行下列命令，例如 XksProxyUriEndpoint 參數。

```
aws configure set cli_follow_urlparam false
```

否則， AWS CLI 1.0 版會將 參數值取代為在該 URI 地址找到的內容。

```
aws kms create-custom-key-store \
    --custom-key-store-name ExampleVPCEndpointXKS \
    --custom-key-store-type EXTERNAL_KEY_STORE \
    --xks-proxy-connectivity VPC_ENDPOINT_SERVICE \
    --xks-proxy-uri-endpoint "https://myproxy-private.xks.example.com" \
    --xks-proxy-uri-path "/kms/xks/v1" \
    --xks-proxy-vpc-endpoint-service-name "com.amazonaws.vpce.us-east-1.vpce-svc-example1" \
    --xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,RawSecretAccessKey=DXjSUawnel2fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

輸出：

```
{
    "CustomKeyStoreId": cks-3234567890abcdef0
}
```

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的建立外部金鑰存放區。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCustomKeyStore。

create-grant

以下程式碼範例顯示如何使用 create-grant。

AWS CLI

建立授予

下列create-grant範例會建立授予，允許使用者exampleUser在1234abcd-12ab-34cd-56ef-1234567890ab範例 KMS 金鑰上使用 decrypt命令。淘汰主體是 adminRole角色。只有在decrypt請求中的加密內容包

含"Department": "IT"金鑰/值對時，授予才會使用EncryptionContextSubset授予限制來允許此許可。

```
aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::123456789012:user/exampleUser \
    --operations Decrypt \
    --constraints EncryptionContextSubset={Department=IT} \
    --retiring-principal arn:aws:iam::123456789012:role/adminRole
```

輸出：

```
{
    "GrantId": "1a2b3c4d2f5e69f440bae30eaec9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",
    "GrantToken": "<grant token here>"
}
```

若要檢視授予的詳細資訊，請使用 list-grants命令。

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 AWS KMS 中的授權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateGrant。

create-key

以下程式碼範例顯示如何使用 create-key。

AWS CLI

範例 1：在 KMS 中建立客戶受管 AWS KMS 金鑰

下列create-key範例會建立對稱加密 KMS 金鑰。

若要建立基本 KMS 金鑰，即對稱加密金鑰，您不需要指定任何參數。這些參數的預設值會建立對稱加密金鑰。

由於此命令未指定金鑰政策，KMS 金鑰會取得以程式設計方式建立之 KMS 金鑰的預設金鑰政策。若要檢視金鑰政策，請使用 get-key-policy命令。若要變更金鑰政策，請使用 put-key-policy命令。

```
aws kms create-key
```

create-key 命令會傳回金鑰中繼資料，包括新 KMS 金鑰的金鑰 ID 和 ARN。您可以使用這些值來識別其他 KMS 操作中的 AWS KMS 金鑰。輸出不包含標籤。若要檢視 KMS 金鑰的標籤，請使用 list-resource-tags command。

輸出：

```
{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": "2017-07-05T14:04:55-07:00",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "Description": "",
        "Enabled": true,
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "AWS_KMS"
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

注意：create-key命令不允許您指定別名。若要為新的 KMS 金鑰建立別名，請使用 create-alias命令。

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的建立AWS 金鑰。

範例 2：建立用於加密和解密的非對稱 RSA KMS 金鑰

下列create-key範例會建立 KMS 金鑰，其中包含用於加密和解密的非對稱 RSA 金鑰對。建立金鑰之後，就無法變更金鑰規格和金鑰用量。：

```
aws kms create-key \
   --key-spec RSA_4096 \
```

```
    --key-usage ENCRYPT_DECRYPT
```

輸出：

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2021-04-05T14:04:55-07:00",
        "CustomerMasterKeySpec": "RSA_4096",
        "Description": "",
        "Enabled": true,
        "EncryptionAlgorithms": [
            "RSAES_OAEP_SHA_1",
            "RSAES_OAEP_SHA_256"
        ],
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "RSA_4096",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "AWS_KMS"
    }
}
```

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的 AWS KMS 中的非對稱金鑰。AWS

範例 3：建立用於簽署和驗證的非對稱橢圓曲線 KMS 金鑰

建立非對稱 KMS 金鑰，其中包含用於簽署和驗證的非對稱橢圓曲線 (ECC) 金鑰對。即使 SIGN_VERIFY是 ECC KMS 金鑰的唯一有效值，仍需要 --key-usage 參數。建立金鑰之後，就無法變更金鑰規格和金鑰用量。：

```
aws kms create-key \
    --key-spec ECC_NIST_P521 \
    --key-usage SIGN_VERIFY
```

輸出：

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2019-12-02T07:48:55-07:00",
        "CustomerMasterKeySpec": "ECC_NIST_P521",
        "Description": "",
        "Enabled": true,
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "ECC_NIST_P521",
        "KeyState": "Enabled",
        "KeyUsage": "SIGN_VERIFY",
        "MultiRegion": false,
        "Origin": "AWS_KMS",
        "SigningAlgorithms": [
            "ECDSA_SHA_512"
        ]
    }
}
```

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的 AWS KMS 中的非對稱金鑰。AWS

範例 4：建立用於簽署和驗證的非對稱 ML-DSA KMS 金鑰

此範例會建立用於簽署和驗證的模組格線數位簽章演算法 (ML-DSA) 金鑰。即使 SIGN_VERIFY是 ML-DSA 金鑰的唯一有效值，仍需要 key-usage 參數。

```
aws kms create-key \
    --key-spec ML_DSA_65 \
    --key-usage SIGN_VERIFY
```

輸出：

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2019-12-02T07:48:55-07:00",
```

```
        "Description": "",
        "Enabled": true,
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "ML_DSA_65",
        "KeyState": "Enabled",
        "KeyUsage": "SIGN_VERIFY",
        "MultiRegion": false,
        "Origin": "AWS_KMS",
        "SigningAlgorithms": [
            "ML_DSA_SHAKE_256"
        ]
    }
}
```

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的 AWS KMS 中的非對稱金鑰。AWS

範例 5：建立 HMAC KMS 金鑰

下列create-key範例會建立 384 位元 HMAC KMS 金鑰。參數GENERATE_VERIFY_MAC的值--key-usage是必要的，即使它是 HMAC KMS 金鑰的唯一有效值。

```
aws kms create-key \
    --key-spec HMAC_384 \
    --key-usage GENERATE_VERIFY_MAC
```

輸出：

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2022-04-05T14:04:55-07:00",
        "CustomerMasterKeySpec": "HMAC_384",
        "Description": "",
        "Enabled": true,
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "HMAC_384",
        "KeyState": "Enabled",
        "KeyUsage": "GENERATE_VERIFY_MAC",
```

```
        "MacAlgorithms": [
            "HMAC_SHA_384"
        ],
        "MultiRegion": false,
        "Origin": "AWS_KMS"
    }
}
```

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的 AWS KMS 中的 HMAC 金鑰。 AWS

範例 6：建立多區域主要 KMS 金鑰

下列create-key範例會建立多區域主要對稱加密金鑰。由於所有參數的預設值都會建立對稱加密金鑰，因此此 KMS 金鑰只需要 --multi-region 參數。在 AWS CLI 中，若要指出布林值參數為 true，只要指定參數名稱即可。

```
aws kms create-key \
    --multi-region
```

輸出：

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef12345678990ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2021-09-02T016:15:21-09:00",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "Description": "",
        "Enabled": true,
        "EncryptionAlgorithms": [
          "SYMMETRIC_DEFAULT"
        ],
        "KeyId": "mrk-1234abcd12ab34cd56ef12345678990ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": true,
```

```
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "PRIMARY",
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef12345678990ab",
                "Region": "us-west-2"
            },
            "ReplicaKeys": []
        },
        "Origin": "AWS_KMS"
    }
}
```

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的 AWS KMS 中的非對稱金鑰。AWS

範例 7：為匯入的金鑰材料建立 KMS 金鑰

下列create-key範例會建立不含金鑰材料的 KMS 金鑰。操作完成後，您可以將自己的金鑰材料匯入 KMS 金鑰。若要建立此 KMS 金鑰，請將 --origin 參數設定為 EXTERNAL。

```
aws kms create-key \
    --origin EXTERNAL
```

輸出：

```
{
     "KeyMetadata": {
         "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
         "AWSAccountId": "111122223333",
         "CreationDate": "2019-12-02T07:48:55-07:00",
         "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
         "Description": "",
         "Enabled": false,
         "EncryptionAlgorithms": [
             "SYMMETRIC_DEFAULT"
         ],
         "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
         "KeyManager": "CUSTOMER",
         "KeySpec": "SYMMETRIC_DEFAULT",
         "KeyState": "PendingImport",
         "KeyUsage": "ENCRYPT_DECRYPT",
```

```
        "MultiRegion": false,
        "Origin": "EXTERNAL"
    }
}
```

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的在 AWS KMS 金鑰中匯入金鑰材料。 AWS

範例 6：在 an AWS CloudHSM 金鑰存放區中建立 KMS 金鑰

下列create-key範例會在指定的 AWS CloudHSM 金鑰存放區中建立 KMS 金鑰。操作會在 KMS 中建立 AWS KMS 金鑰及其中繼資料，並在與自訂金鑰存放區相關聯的 AWS CloudHSM 叢集中建立金鑰材料。--custom-key-store-id 和 --origin 是必要參數。

```
aws kms create-key \
    --origin AWS_CLOUDHSM \
    --custom-key-store-id cks-1234567890abcdef0
```

輸出：

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CloudHsmClusterId": "cluster-1a23b4cdefg",
        "CreationDate": "2019-12-02T07:48:55-07:00",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "CustomKeyStoreId": "cks-1234567890abcdef0",
        "Description": "",
        "Enabled": true,
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "AWS_CLOUDHSM"
    }
```

```
}
```

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 CloudHSM 金鑰存放區。 AWS

範例 8：在外部金鑰存放區中建立 KMS 金鑰

下列create-key範例會在指定的外部金鑰存放區中建立 KMS 金鑰。此命令需要 --custom-key-store-id、 --origin和 --xks-key-id 參數。

--xks-key-id 參數指定外部金鑰管理器中現有對稱加密金鑰的 ID。此金鑰做為 KMS 金鑰的外部金鑰材料。 --origin 參數的值必須是 EXTERNAL_KEY_STORE。 custom-key-store-id 參數必須識別連接到其外部金鑰存放區代理的外部金鑰存放區。

```
aws kms create-key \
    --origin EXTERNAL_KEY_STORE \
    --custom-key-store-id cks-9876543210fedcba9 \
    --xks-key-id bb8562717f809024
```

輸出：

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2022-12-02T07:48:55-07:00",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "CustomKeyStoreId": "cks-9876543210fedcba9",
        "Description": "",
        "Enabled": true,
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "EXTERNAL_KEY_STORE",
        "XksKeyConfiguration": {
```

```
            "Id": "bb8562717f809024"
        }
    }
}
```

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的外部金鑰存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateKey。

`decrypt`

以下程式碼範例顯示如何使用 `decrypt`。

AWS CLI

範例 1：使用對稱 KMS 金鑰 (Linux 和 macOS) 解密加密的訊息

下列`decrypt`命令範例示範使用 CLI AWS 解密資料的建議方法。此版本示範如何在對稱 KMS 金鑰下解密資料。

在 file.In 中提供 `--ciphertext-blob` 參數值的加密文字，使用 `fileb://`字首，告知 CLI 從二進位檔案讀取資料。如果檔案不在目前的目錄中，請輸入檔案的完整路徑。如需從檔案讀取 AWS CLI 參數值的詳細資訊， 請參閱 Command Line Interface AWS 使用者指南中的從檔案 <https://docs.aws.amazon.com/cli/latest/userguide/cli-usage-parameters-file.html> 載入 CLI 參數，以及 AWS Command Line Tool 部落格中的本機檔案參數最佳實務 <https://aws.amazon.com/blogs/developer/best-practices-for-local-file-parameters/>。指定 KMS 金鑰以解密加密文字。使用對稱 KMS 金鑰解密時不需要 `--key-id` 參數。 AWS KMS 可以取得用於加密加密加密加密文字中中繼資料之 KMS 金鑰的金鑰 ID。 AWS 但是指定您正在使用的 KMS 金鑰永遠是最佳實務。此實務可確保您使用您想要的 KMS 金鑰，並防止您不小心使用您不信任的 KMS 金鑰解密加密文字。請求純文字輸出做為文字值。 `--query` 參數會告知 CLI 從輸出取得 `Plaintext` 欄位的值。`--output` 參數會以 text.Base64 形式傳回輸出。Base64 會將純文字解碼，並將其儲存在檔案中。下列範例會將 `Plaintext` 參數的值輸送 (|) 至 Base64 公用程式，以將其解碼。然後，它會將解碼的輸出重新導向 (`>`) 至 `ExamplePlaintext` 檔案。

執行此命令之前，請將範例金鑰 ID 取代為來自您 AWS 帳戶的有效金鑰 ID。

```
aws kms decrypt \
    --ciphertext-blob fileb://ExampleEncryptedFile \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --output text \
    --query Plaintext | base64 \
```

```
    --decode > ExamplePlaintextFile
```

此命令不會產生輸出。來自 decrypt命令的輸出是 base64 解碼並儲存在 檔案中。

如需詳細資訊，請參閱 AWS Key Management Service API 參考中的解密。

範例 2：使用對稱 KMS 金鑰解密加密的訊息 (Windows 命令提示)

下列範例與上一個範例相同，但使用 certutil公用程式對純文字資料進行 Base64-decode。此程序需要兩個命令，如下列範例所示。

執行此命令之前，請將範例金鑰 ID 取代為來自您 AWS 帳戶的有效金鑰 ID。

```
aws kms decrypt ^
    --ciphertext-blob fileb://ExampleEncryptedFile ^
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^
    --output text ^
    --query Plaintext > ExamplePlaintextFile.base64
```

執行 certutil 命令。

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

輸出：

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

如需詳細資訊，請參閱 AWS Key Management Service API 參考中的解密。

範例 3：使用非對稱 KMS 金鑰 (Linux 和 macOS) 解密加密的訊息

下列decrypt命令範例示範如何解密以 RSA 非對稱 KMS 金鑰加密的資料。

使用非對稱 KMS 金鑰時，需要指定用於加密純文字的演算法的 encryption-algorithm 參數。

執行此命令之前，請將範例金鑰 ID 取代為來自您 AWS 帳戶的有效金鑰 ID。

```
aws kms decrypt \
    --ciphertext-blob fileb://ExampleEncryptedFile \
    --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
```

```
    --encryption-algorithm RSAES_OAEP_SHA_256 \
    --output text \
    --query Plaintext | base64 \
    --decode > ExamplePlaintextFile
```

此命令不會產生輸出。來自 decrypt命令的輸出是 base64 解碼並儲存在 檔案中。

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的 AWS KMS 中的非對稱金鑰。AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的解密。

delete-alias

以下程式碼範例顯示如何使用 delete-alias。

AWS CLI

刪除 AWS KMS 別名

下列delete-alias範例會刪除別名 alias/example-alias。別名名稱必須以別名/ 開頭。

```
aws kms delete-alias \
    --alias-name alias/example-alias
```

此命令不會產生輸出。若要尋找別名，請使用 list-aliases命令。

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的刪除別名。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteAlias。

delete-custom-key-store

以下程式碼範例顯示如何使用 delete-custom-key-store。

AWS CLI

刪除自訂金鑰存放區

下列delete-custom-key-store範例會刪除指定的自訂金鑰存放區。

刪除 an AWS CloudHSM 金鑰存放區不會影響相關聯的 CloudHSM 叢集。刪除外部金鑰存放區不會影響相關聯的外部金鑰存放區代理、外部金鑰管理器或外部金鑰。

注意：您必須先排程刪除自訂金鑰存放區中的所有 KMS 金鑰，然後等待刪除這些 KMS 金鑰。然後，您必須中斷連接自訂金鑰存放區。如需在您的自訂金鑰存放區中尋找 KMS 金鑰的說明，請參閱 Key AWS Management Service 開發人員指南中的刪除 an AWS CloudHSM 金鑰存放區 (API)。

```
delete-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0
```

此命令不會傳回任何輸出。若要確認自訂金鑰存放區已刪除，請使用 `describe-custom-key-stores`命令。

如需有關刪除 an AWS CloudHSM 金鑰存放區的資訊，請參閱 Key AWS Management Service 開發人員指南中的刪除 an AWS CloudHSM 金鑰存放區。

如需有關刪除外部金鑰存放區的資訊，請參閱 Key Management Service 開發人員指南中的刪除外部金鑰存放區。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCustomKeyStore。

`delete-imported-key-material`

以下程式碼範例顯示如何使用 `delete-imported-key-material`。

AWS CLI

從 KMS 金鑰刪除匯入的金鑰材料

下列`delete-imported-key-material`範例會刪除已匯入 KMS 金鑰的金鑰材料。

```
aws kms delete-imported-key-material \
   --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

輸出：

```
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的刪除匯入的金鑰材料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteImportedKeyMaterial。

derive-shared-secret

以下程式碼範例顯示如何使用 derive-shared-secret。

AWS CLI

若要衍生共用秘密

下列derive-shared-secret範例使用金鑰協議演算法衍生共用秘密。

您必須使用KeyUsage值為 的非對稱 NIST 建議的橢圓曲線 (ECC) 或 SM2 （僅限中國區域） KMS 金鑰對KEY_AGREEMENT來呼叫 DeriveSharedSecret。

```
aws kms derive-shared-secret \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-agreement-algorithm ECDH \
    --public-
key "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvH3Yj0wbkLEpUl95Cv1cJVjsVNSjwGq3tCLnzXfhVwV
```

輸出：

```
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "SharedSecret": "MEYCIQCKZLWyTk5runarx6XiAkU9gv3lbwPO/pHa
+DXFehzdDwIhANwpsIV2g/9SPWLLsF6p/hiSskuIXMTRwqrMdVKWTMHG",
    "KeyAgreementAlgorithm": "ECDH",
    "KeyOrigin": "AWS_KMS"
}
```

如需詳細資訊，請參閱 AWS Key Management Service API 參考中的 DeriveSharedSecret。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeriveSharedSecret。

describe-custom-key-stores

以下程式碼範例顯示如何使用 describe-custom-key-stores。

AWS CLI

範例 1：取得有關 an AWS CloudHSM 金鑰存放區的詳細資訊

下列describe-custom-key-store範例顯示指定 AWS CloudHSM 金鑰存放區的詳細資訊。命令對於所有類型的自訂金鑰存放區都是相同的，但輸出會因金鑰存放區類型而有所不同，對於外部金鑰存放區，則會有其連線選項。

根據預設，此命令會顯示帳戶和區域中所有自訂金鑰存放區的相關資訊。若要顯示特定自訂金鑰存放區的相關資訊，請使用 custom-key-store-name或 custom-key-store-id 參數。

```
aws kms describe-custom-key-stores \
    --custom-key-store-name ExampleCloudHSMKeyStore
```

此命令的輸出包含有關 AWS CloudHSM 金鑰存放區的實用詳細資訊，包括其連線狀態 (ConnectionState)。如果連線狀態為 FAILED，則輸出會包含描述問題ConnectionErrorCode的欄位。

輸出：

```
{
    "CustomKeyStores": [
        {
            "CloudHsmClusterId": "cluster-1a23b4cdefg",
            "ConnectionState": "CONNECTED",
            "CreationDate": "2022-04-05T14:04:55-07:00",
            "CustomKeyStoreId": "cks-1234567890abcdef0",
            "CustomKeyStoreName": "ExampleExternalKeyStore",
            "TrustAnchorCertificate": "<certificate appears here>"
        }
    ]
}
```

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的檢視 an AWS CloudHSM 金鑰存放區。 AWS

範例 2：取得具有公有端點連線的外部金鑰存放區詳細資訊

下列describe-custom-key-store範例顯示指定外部金鑰存放區的詳細資訊。命令對於所有類型的自訂金鑰存放區都是相同的，但輸出會因金鑰存放區類型而有所不同，對於外部金鑰存放區，則會有其連線選項。

根據預設，此命令會顯示帳戶和區域中所有自訂金鑰存放區的相關資訊。若要顯示特定自訂金鑰存放區的相關資訊，請使用 custom-key-store-name或 custom-key-store-id 參數。

```
aws kms describe-custom-key-stores \
```

```
    --custom-key-store-id cks-9876543210fedcba9
```

此命令的輸出包含有關外部金鑰存放區的實用詳細資訊，包括其連線狀態 (ConnectionState)。如果連線狀態為 FAILED，則輸出會包含描述問題ConnectionErrorCode的欄位。

輸出：

```
{
    "CustomKeyStores": [
        {
            "CustomKeyStoreId": "cks-9876543210fedcba9",
            "CustomKeyStoreName": "ExampleXKS",
            "ConnectionState": "CONNECTED",
            "CreationDate": "2022-12-02T07:48:55-07:00",
            "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
            "XksProxyConfiguration": {
                "AccessKeyId": "ABCDE12345670EXAMPLE",
                "Connectivity": "PUBLIC_ENDPOINT",
                "UriEndpoint": "https://myproxy.xks.example.com",
                "UriPath": "/example-prefix/kms/xks/v1"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的檢視外部金鑰存放區。 AWS

範例 3：取得具有 VPC 端點服務連線之外部金鑰存放區的詳細資訊

下列describe-custom-key-store範例顯示指定外部金鑰存放區的詳細資訊。命令對於所有類型的自訂金鑰存放區都是相同的，但輸出會因金鑰存放區類型而有所不同，對於外部金鑰存放區，則會有其連線選項。

根據預設，此命令會顯示帳戶和區域中所有自訂金鑰存放區的相關資訊。若要顯示特定自訂金鑰存放區的相關資訊，請使用 custom-key-store-name或 custom-key-store-id 參數。

```
aws kms describe-custom-key-stores \
    --custom-key-store-id cks-2234567890abcdef0
```

此命令的輸出包含有關外部金鑰存放區的實用詳細資訊，包括其連線狀態 (ConnectionState)。如果連線狀態為 FAILED，則輸出會包含描述問題ConnectionErrorCode的欄位。

輸出：

```
{
    "CustomKeyStores": [
        {
            "CustomKeyStoreId": "cks-3234567890abcdef0",
            "CustomKeyStoreName": "ExampleVPCExternalKeyStore",
            "ConnectionState": "CONNECTED",
            "CreationDate": "2022-12-22T07:48:55-07:00",
            "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
            "XksProxyConfiguration": {
                "AccessKeyId": "ABCDE12345670EXAMPLE",
                "Connectivity": "VPC_ENDPOINT_SERVICE",
                "UriEndpoint": "https://myproxy-private.xks.example.com",
                "UriPath": "/kms/xks/v1",
                "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-
example1"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的檢視外部金鑰存放區。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCustomKeyStores。

describe-key

以下程式碼範例顯示如何使用 describe-key。

AWS CLI

範例 1：尋找 KMS 金鑰的詳細資訊

下列describe-key範例會取得範例帳戶和區域中 Amazon S3 AWS 受管金鑰的詳細資訊。您可以使用此命令來尋找 AWS 受管金鑰和客戶受管金鑰的詳細資訊。

若要指定 KMS 金鑰，請使用 key-id 參數。此範例使用別名名稱值，但您可以在此命令中使用金鑰 ID、金鑰 ARN、別名名稱或別名 ARN。

```
aws kms describe-key \
    --key-id alias/aws/s3
```

輸出：

```
{
    "KeyMetadata": {
        "AWSAccountId": "846764612917",
        "KeyId": "b8a9477d-836c-491f-857e-07937918959b",
        "Arn": "arn:aws:kms:us-west-2:846764612917:key/
b8a9477d-836c-491f-857e-07937918959b",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "CreationDate": 2017-06-30T21:44:32.140000+00:00,
        "Enabled": true,
        "Description": "Default KMS key that protects my S3 objects when no other
 key is defined",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "AWS",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的檢視AWS 金鑰。

範例 2：取得 RSA 非對稱 KMS 金鑰的詳細資訊

下列`describe-key`範例會取得用於簽署和驗證的非對稱 RSA KMS 金鑰的詳細資訊。

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

輸出：

```
{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
```

```
        "CreationDate": "2019-12-02T19:47:14.861000+00:00",
        "CustomerMasterKeySpec": "RSA_2048",
        "Enabled": false,
        "Description": "",
        "KeyState": "Disabled",
        "Origin": "AWS_KMS",
        "MultiRegion": false,
        "KeyManager": "CUSTOMER",
        "KeySpec": "RSA_2048",
        "KeyUsage": "SIGN_VERIFY",
        "SigningAlgorithms": [
            "RSASSA_PKCS1_V1_5_SHA_256",
            "RSASSA_PKCS1_V1_5_SHA_384",
            "RSASSA_PKCS1_V1_5_SHA_512",
            "RSASSA_PSS_SHA_256",
            "RSASSA_PSS_SHA_384",
            "RSASSA_PSS_SHA_512"
        ]
    }
}
```

範例 3：取得多區域複本金鑰的詳細資訊

下列describe-key範例會取得多區域複本金鑰的中繼資料。此多區域金鑰是對稱加密金鑰。任何多區域金鑰的describe-key命令輸出都會傳回主金鑰及其所有複本的相關資訊。

```
aws kms describe-key \
    --key-id arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab
```

輸出：

```
{
    "KeyMetadata": {
        "MultiRegion": true,
        "AWSAccountId": "111122223333",
        "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "CreationDate": "2021-06-28T21:09:16.114000+00:00",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "Description": "",
        "Enabled": true,
```

```
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "Origin": "AWS_KMS",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "PRIMARY",
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "us-west-2"
            },
            "ReplicaKeys": [
                {
                    "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "eu-west-1"
                },
                {
                    "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "ap-northeast-1"
                },
                {
                    "Arn": "arn:aws:kms:sa-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "sa-east-1"
                }
            ]
        }
    }
}
```

範例 4：取得 HMAC KMS 金鑰的詳細資訊

下列`describe-key`範例會取得 HMAC KMS 金鑰的詳細資訊。

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

輸出：

```
{
    "KeyMetadata": {
        "AWSAccountId": "123456789012",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Arn": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": "2022-04-03T22:23:10.194000+00:00",
        "Enabled": true,
        "Description": "Test key",
        "KeyUsage": "GENERATE_VERIFY_MAC",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "HMAC_256",
        "MacAlgorithms": [
            "HMAC_SHA_256"
        ],
        "MultiRegion": false
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeKey。

disable-key-rotation

以下程式碼範例顯示如何使用 disable-key-rotation。

AWS CLI

停用 KMS 金鑰的自動輪換

下列disable-key-rotation範例會停用客戶受管 KMS 金鑰的自動輪換。若要重新啟用自動輪換，請使用 enable-key-rotation命令。

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

此命令不會產生輸出。若要確認 KMS 金鑰的自動輪換已停用，請使用 get-key-rotation-status命令。

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的輪換AWS 金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableKeyRotation。

disable-key

以下程式碼範例顯示如何使用 disable-key。

AWS CLI

暫時停用 KMS 金鑰

下列disable-key命令會停用客戶受管 KMS 金鑰。若要重新啟用 KMS 金鑰，請使用 enable-key命令。

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

此命令不會產生輸出。

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的啟用和停用AWS 金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableKey。

disconnect-custom-key-store

以下程式碼範例顯示如何使用 disconnect-custom-key-store。

AWS CLI

中斷連接自訂金鑰存放區

下列disconnect-custom-key-store範例會中斷自訂金鑰存放區與其 AWS CloudHSM 叢集的連線。您可以中斷連接金鑰存放區以對問題進行故障診斷、更新其設定，或防止金鑰存放區中的 KMS 金鑰用於密碼編譯操作。

此命令對所有自訂金鑰存放區都相同，包括 AWS CloudHSM 金鑰存放區和外部金鑰存放區。

執行此命令之前，請將範例自訂金鑰存放區 ID 以有效的 ID 取代。

```
$ aws kms disconnect-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0
```

此命令不會產生輸出。請確認命令是否有效，請使用 `describe-custom-key-stores`命令。

如需中斷連接 an AWS CloudHSM 金鑰存放區的詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的連接和中斷連接 an AWS CloudHSM 金鑰存放區。

如需中斷連接外部金鑰存放區的詳細資訊，請參閱 Key AWS Management Service 開發人員指南中的連接和中斷連接外部金鑰存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisconnectCustomKeyStore。

`enable-key-rotation`

以下程式碼範例顯示如何使用 `enable-key-rotation`。

AWS CLI

啟用 KMS 金鑰的自動輪換

下列`enable-key-rotation`範例會啟用客戶受管 KMS 金鑰的自動輪換，輪換期間為 180 天。KMS 金鑰將從此命令完成之日起一年 （ 大約 365 天 ） 輪換，之後每年輪換一次。

`--key-id` 參數可識別 KMS 金鑰。此範例使用金鑰 ARN 值，但您可以使用金鑰 ID 或 KMS 金鑰的 ARN。 `--rotation-period-in-days` 參數會指定每個輪換日期之間的天數。指定介於 90 到 2560 天之間的值。如果未指定任何值，預設值為 365 天。

```
aws kms enable-key-rotation \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --rotation-period-in-days 180
```

此命令不會產生輸出。若要確認 KMS 金鑰已啟用，請使用 `get-key-rotation-status`命令。

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的輪換金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableKeyRotation。

`enable-key`

以下程式碼範例顯示如何使用 `enable-key`。

AWS CLI

啟用 KMS 金鑰

下列enable-key範例會啟用客戶受管金鑰。您可以使用像這樣的命令，透過使用 disable-key命令來啟用暫時停用的 KMS 金鑰。您也可以使用它來啟用已停用的 KMS 金鑰，因為其已排定刪除，且刪除已取消。

若要指定 KMS 金鑰，請使用 key-id 參數。此範例使用金鑰 ID 值，但您可以在此命令中使用金鑰 ID 或金鑰 ARN 值。

執行此命令之前，請將範例金鑰 ID 取代為有效的金鑰 ID。

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

此命令不會產生輸出。若要確認 KMS 金鑰已啟用，請使用 describe-key命令。查看describe-key輸出中 KeyState和 Enabled 欄位的值。

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的啟用和停用AWS 金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableKey。

encrypt

以下程式碼範例顯示如何使用 encrypt。

AWS CLI

範例 1：在 Linux 或 MacOS 上加密檔案的內容

下列encrypt命令示範使用 CLI AWS 加密資料的建議方式。

```
aws kms encrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --plaintext fileb://ExamplePlaintextFile \
    --output text \
    --query CiphertextBlob | base64 \
    --decode > ExampleEncryptedFile
```

命令會執行數個動作：

使用 --plaintext 參數來指示要加密的資料。此參數值必須是 base64 編碼。 plaintext 參數的值必須是 base64 編碼，或者您必須使用 fileb://字首，指示 AWS CLI 從 檔案讀取二進

位資料。如果檔案不在目前的目錄中，請輸入檔案的完整路徑。例如：fileb:///var/tmp/ExamplePlaintextFile 或 fileb://C:\Temp\ExamplePlaintextFile。如需從檔案讀取 AWS CLI 參數值的詳細資訊， 請參閱《 AWS 命令列界面使用者指南》中的從檔案載入參數，以及《 AWS 命令列工具部落格》上的本機檔案參數的最佳實務。 使用 --output和 --query 參數來控制命令的輸出。這些參數會擷取加密的資料， 稱為加密文字 從命令的輸出。如需控制輸出的詳細資訊， 請參閱《 命令列界面使用者指南》中的控制命令輸出。使用 base64公用程式將擷取的輸出解碼為二進位資料。成功encrypt命令傳回的加密文字是 base64 編碼的文字。AWS 您必須先解碼此文字，才能使用 AWS CLI 將其解密。將二進位加密文字儲存到 檔案。命令 (> ExampleEncryptedFile) 的最終部分會將二進位加密文字儲存到 檔案，讓解密更容易。如需使用 CLI AWS 解密資料的範例命令，請參閱解密範例。

範例 2：使用 AWS CLI 加密 Windows 上的資料

此範例與上一個範例相同，但它使用 certutil工具而非 base64。此程序需要兩個命令，如下列範例所示。

```
aws kms encrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --plaintext fileb://ExamplePlaintextFile \
    --output text \
    --query CiphertextBlob > C:\Temp\ExampleEncryptedFile.base64

certutil -decode C:\Temp\ExampleEncryptedFile.base64 C:\Temp\ExampleEncryptedFile
```

範例 3：使用非對稱 KMS 金鑰加密

下列encrypt命令顯示如何使用非對稱 KMS 金鑰加密純文字。--encryption-algorithm 參數是必要參數。如同所有 encrypt CLI 命令， plaintext 參數必須是 base64 編碼，或者您必須使用 fileb://字首，告知 AWS CLI 從 檔案讀取二進位資料。

```
aws kms encrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --encryption-algorithm RSAES_OAEP_SHA_256 \
    --plaintext fileb://ExamplePlaintextFile \
    --output text \
    --query CiphertextBlob | base64 \
    --decode > ExampleEncryptedFile
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的加密。

generate-data-key-pair-without-plaintext

以下程式碼範例顯示如何使用 generate-data-key-pair-without-plaintext。

AWS CLI

產生 ECC NIST P384 非對稱資料金鑰對

下列generate-data-key-pair-without-plaintext範例會請求在 外部使用的 ECC NIST P384 金鑰對 AWS。

命令會傳回純文字公有金鑰和以指定 KMS 金鑰加密的私有金鑰複本。它不會傳回純文字私有金鑰。您可以安全地將加密的私有金鑰與加密的資料一起存放，並在您需要使用私有金鑰時呼叫 AWS KMS 來解密私有金鑰。

若要請求 ECC NIST P384 非對稱資料金鑰對，請使用 值為 的 key-pair-spec 參數ECC_NIST_P384。

您指定的 KMS 金鑰必須是對稱加密 KMS 金鑰，也就是KeySpec值為 的 KMS 金鑰SYMMETRIC_DEFAULT。

注意：此範例輸出中的值會截斷以顯示。

```
aws kms generate-data-key-pair-without-plaintext \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-pair-spec ECC_NIST_P384
```

輸出：

```
{
    "PrivateKeyCiphertextBlob": "AQIDAHi6LtupRpdKl2aJTzkK6FbhOtQkMlQJJH3PdtHvS/y
+hAFFxmiD134doUDzMGmfCEtcAAAHaTCCB2UGCSqGSIb3DQEHBqCCB1...",
    "PublicKey":
 "MIIBojANBgkqhkiG9w0BAQEFAAOCAY8AMIIBigKCAYEA3A3eGMyPrvSn7+LdlJE1oUoQV5HpEuHAVbdOyND
+NmYDH/mL1OSIEuLrcdZ5hrMH4pk83r40l...",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
    "KeyPairSpec": "ECC_NIST_P384"
}
```

PublicKey 和 PrivateKeyCiphertextBlob會以 base64 編碼格式傳回。

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的資料金鑰對。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GenerateDataKeyPairWithoutPlaintext。

generate-data-key-pair

以下程式碼範例顯示如何使用 generate-data-key-pair。

AWS CLI

產生 2048 位元 RSA 非對稱資料金鑰對

下列generate-data-key-pair範例會請求在 外部使用的 2048 位元 RSA 非對稱資料金鑰對 AWS。命令會傳回純文字公有金鑰和純文字私有金鑰以供立即使用和刪除，以及使用指定的 KMS 金鑰加密的私有金鑰複本。您可以使用加密的資料安全地存放加密的私有金鑰。

若要請求 2048 位元 RSA 非對稱資料金鑰對，請使用 值為 的 key-pair-spec 參數RSA_2048。

您指定的 KMS 金鑰必須是對稱加密 KMS 金鑰，也就是KeySpec值為 的 KMS 金鑰SYMMETRIC_DEFAULT。

注意：此範例輸出中的值會截斷以顯示。

```
aws kms generate-data-key-pair \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-pair-spec RSA_2048
```

輸出：

```
{
    "PrivateKeyCiphertextBlob": "AQIDAHi6LtupRpdKl2aJTzkK6FbhOtQkMlQJJH3PdtHvS/y
+hAFFxmiD134doUDzMGmfCEtcAAAHaTCCB2UGCSqGSIb3DQEHBqCCB1...",
    "PrivateKeyPlaintext": "MIIG/
QIBADANBgkqhkiG9w0BAQEFAASCBucwggbjAgEAAoIBgQDcDd4YzI
+u9Kfv4t2UkTWhShBXkekS4cBVt07I0P42ZgMf+YvU5IgS4ut...",
    "PublicKey":
 "MIIBojANBgkqhkiG9w0BAQEFAAOCAY8AMIIBigKCAYEA3A3eGMyPrvSn7+LdlJE1oUoQV5HpEuHAVbdOyND
+NmYDH/mL1OSIEuLrcdZ5hrMH4pk83r40l...",
```

```
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
    "KeyPairSpec": "RSA_2048"
}
```

PublicKey、 PrivateKeyPlaintext和 PrivateKeyCiphertextBlob會以 base64 編碼格式傳回。

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的資料金鑰對。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GenerateDataKeyPair。

generate-data-key-without-plaintext

以下程式碼範例顯示如何使用 generate-data-key-without-plaintext。

AWS CLI

產生不含純文字金鑰的 256 位元對稱資料金鑰

下列generate-data-key-without-plaintext範例會請求 256 位元對稱資料金鑰的加密副本，以供 外部使用 AWS。當您準備好要使用資料金鑰時，您可以呼叫 AWS KMS 來解密資料金鑰。

若要請求 256 位元資料金鑰，請使用 值為 的 key-spec 參數AES_256。若要請求 128 位元資料金鑰，請使用 值為 的 key-spec 參數AES_128。對於所有其他資料金鑰長度，請使用 number-of-bytes 參數。

您指定的 KMS 金鑰必須是對稱加密 KMS 金鑰，也就是金鑰規格值為 SYMMETRIC_DEFAULT 的 KMS 金鑰。

```
aws kms generate-data-key-without-plaintext \
    --key-id "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab" \
    --key-spec AES_256
```

輸出：

```
{
```

```
    "CiphertextBlob":
 "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgkqhk
    "KeyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

CiphertextBlob （加密的資料金鑰） 會以 base64 編碼格式傳回。

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的資料金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GenerateDataKeyWithoutPlaintext。

generate-data-key

以下程式碼範例顯示如何使用 generate-data-key。

AWS CLI

範例 1：產生 256 位元對稱資料金鑰

下列generate-data-key範例會請求在 外部使用的 256 位元對稱資料金鑰 AWS。命令會傳回純文字資料金鑰以供立即使用和刪除，以及在指定的 KMS 金鑰下加密的資料金鑰複本。加密的資料金鑰可以安全地跟加密資料一起存放。

若要請求 256 位元資料金鑰，請使用 值為 的 key-spec 參數AES_256。若要請求 128 位元資料金鑰，請使用 值為 的 key-spec 參數AES_128。對於所有其他資料金鑰長度，請使用 number-of-bytes 參數。

您指定的 KMS 金鑰必須是對稱加密 KMS 金鑰，也就是金鑰規格值為 SYMMETRIC_DEFAULT 的 KMS 金鑰。

```
aws kms generate-data-key \
    --key-id alias/ExampleAlias \
    --key-spec AES_256
```

輸出：

```
{
    "Plaintext": "VdzKNHGzUAzJeRBVY+uUmofUGGiDzyB3+i9fVkh3piw=",
```

```
    "KeyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
    "CiphertextBlob":
 "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgkqhk
+YdhV8MrkBQPeac0ReRVNDt9qleAt+SHgIRF8P0H+7U="
}
```

Plaintext（純文字資料金鑰）和 CiphertextBlob（加密的資料金鑰）會以 base64 編碼格式傳回。

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的資料金鑰。範例 2：產生 512 位元對稱資料金鑰

下列generate-data-key範例會請求 512 位元對稱資料金鑰以進行加密和解密。命令會傳回純文字資料金鑰以供立即使用和刪除，以及在指定的 KMS 金鑰下加密的資料金鑰複本。加密的資料金鑰可以安全地跟加密資料一起存放。

若要請求 128 或 256 位元以外的金鑰長度，請使用 number-of-bytes 參數。若要請求 512 位元資料金鑰，下列範例會使用 值為 64 （位元組）的 number-of-bytes 參數。

您指定的 KMS 金鑰必須是對稱加密 KMS 金鑰，也就是金鑰規格值為 SYMMETRIC_DEFAULT 的 KMS 金鑰。

注意：此範例輸出中的值會截斷以顯示。

```
aws kms generate-data-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --number-of-bytes 64
```

輸出：

```
{
    "CiphertextBlob": "AQIBAHi6LtupRpdKl2aJTzkK6FbhOtQkMlQJJH3PdtHvS/y+hAEnX/
QQNmMwDfg2korNMEc8AAACaDCCAmQGCSqGSIb3DQEHBqCCAlUwggJRAgEAMIICSgYJKoZ...",
    "Plaintext": "ty8Lr0Bk6OF07M2BWt6qbFdNB
+G00ZLtf5MSEb4al3R2UKWGOp06njAwy2n72VRm2m7z/
Pm9Wpbvttz6a4lSo9hgPvKhZ5y6RTm4OovEXiVfBveyX3DQxDzRSwbKDPk/...",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
```

```
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

Plaintext（純文字資料金鑰）和 CiphertextBlob（加密的資料金鑰）會以 base64 編碼格式傳回。

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的資料金鑰。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GenerateDataKey。

generate-mac

以下程式碼範例顯示如何使用 generate-mac。

AWS CLI

範例 1：產生訊息的 HMAC

下列generate-mac命令會產生訊息的 HMAC、HMAC KMS 金鑰和 MAC 演算法。演算法必須由指定的 HMAC KMS 金鑰支援。

在 AWS CLI v2 中， message 參數的值必須是 Base64-encoded。或者，您可以將訊息儲存在檔案中，並使用 fileb://字首，告知 AWS CLI 從檔案讀取二進位資料。

執行此命令之前，請將範例金鑰 ID 取代為來自您 AWS 帳戶的有效金鑰 ID。金鑰 ID 必須代表金鑰使用量為 的 HMAC KMS 金鑰GENERATE_VERIFY_MAC。

```
msg=(echo 'Hello World' | base64)

aws kms generate-mac \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://Message \
    --mac-algorithm HMAC_SHA_384
```

輸出：

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "Mac": "<HMAC_TAG>",
```

```
    "MacAlgorithm": "HMAC_SHA_384"
}
```

如需在 KMS AWS 中使用 HMAC KMS 金鑰的詳細資訊，請參閱《金鑰管理服務開發人員指南》中的 AWS KMS 中的 HMAC 金鑰。 AWS

範例 2：將 HMAC 儲存在檔案中 (Linux 和 macOs)

下列generate-mac範例會為儲存在本機檔案中的短訊息產生 HMAC。命令也會從回應取得 Mac 屬性，Base64-decodes，並將其儲存在 ExampleMac 檔案中。您可以在驗證 MAC 的verify-mac命令中使用 MAC 檔案。

generate-mac 命令需要 Base64-encoded的訊息和 HMAC KMS 金鑰支援的 MAC 演算法。若要取得 KMS 金鑰支援的 MAC 演算法，請使用 describe-key命令。

執行此命令之前，請將範例金鑰 ID 取代為來自您 AWS 帳戶的有效金鑰 ID。金鑰 ID 必須代表金鑰使用量為 GENERATE_VERIFY_MAC 的非對稱 KMS 金鑰。

```
echo 'hello world' | base64 > EncodedMessage

aws kms generate-mac \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://EncodedMessage \
    --mac-algorithm HMAC_SHA_384 \
    --output text \
    --query Mac | base64 --decode > ExampleMac
```

此命令不會產生輸出。此範例會擷取輸出的 Mac 屬性，並將其儲存在檔案中。

如需在 KMS AWS 中使用 HMAC KMS 金鑰的詳細資訊，請參閱《金鑰管理服務開發人員指南》中的 AWS KMS 中的 HMAC 金鑰。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GenerateMac。

generate-random

以下程式碼範例顯示如何使用 generate-random。

AWS CLI

範例 1：產生 256 位元隨機位元組字串 (Linux 或 macOs)

下列generate-random範例會產生 256 位元 (32 位元組）、Base64 編碼的隨機位元組字串。此範例會解碼位元組字串，並將其儲存在隨機檔案中。

執行此命令時，您必須使用 number-of-bytes 參數以位元組為單位指定隨機值的長度。

當您執行此命令時，不會指定 KMS 金鑰。隨機位元組字串與任何 KMS 金鑰無關。

根據預設， AWS KMS 會產生隨機數字。不過，如果您指定自訂金鑰存放區，隨機位元組字串會在與自訂金鑰存放區相關聯的 AWS CloudHSM 叢集中產生。

此範例使用下列參數和值：

它使用值為 的必要--number-of-bytes參數32來請求 32 位元組 (256 位元） string.It 使用值為 的 --output 參數text來指示 AWS CLI 以文字形式傳回輸出。 而不是 JSON.It 使用 從 response.It 管道 (|) 將命令的輸出--query parameter擷取到 base64公用程式的 Plaintext 屬性值， 會解碼擷取的 output.It 使用重新導向運算子 (>) 將解碼的位元組字串儲存到 ExampleRandom file.It 使用重新導向運算子 (>) 將二進位加密文字儲存到檔案。

```
aws kms generate-random \
    --number-of-bytes 32 \
    --output text \
    --query Plaintext | base64 --decode > ExampleRandom
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Key Management Service API 參考中的 GenerateRandom。

範例 2：產生 256 位元隨機數字 (Windows 命令提示）

下列範例使用 generate-random命令來產生 256 位元 (32 位元組）、Base64 編碼的隨機位元組字串。此範例會解碼位元組字串，並將其儲存在隨機檔案中。此範例與上一個範例相同，只是在將隨機位元組字串儲存在檔案中之前，它在 Windows 中使用 certutil公用程式來對隨機位元組字串進行 base64 解碼。

首先，產生 base64 編碼的隨機位元組字串，並將其儲存在暫存檔案中ExampleRandom.base64。

```
aws kms generate-random \
    --number-of-bytes 32 \
    --output text \
```

```
    --query Plaintext > ExampleRandom.base64
```

由於generate-random命令的輸出會儲存在檔案中，因此此範例不會產生輸出。

現在請使用 certutil -decode命令來解碼 ExampleRandom.base64 檔案中的 base64 編碼位元組字串。然後，它會將解碼後的位元組字串儲存在 ExampleRandom 檔案中。

```
certutil -decode ExampleRandom.base64 ExampleRandom
```

輸出：

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

如需詳細資訊，請參閱 AWS Key Management Service API 參考中的 GenerateRandom。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GenerateRandom。

get-key-policy

以下程式碼範例顯示如何使用 get-key-policy。

AWS CLI

將金鑰政策從一個 KMS 金鑰複製到另一個 KMS 金鑰

下列get-key-policy範例會從一個 KMS 金鑰取得金鑰政策，並將其儲存在文字檔案中。然後，它會使用文字檔案取代不同 KMS 金鑰的政策做為政策輸入。

由於 的 --policy 參數put-key-policy需要字串，您必須使用 --output text選項，將輸出傳回為文字字串，而非 JSON。

```
aws kms get-key-policy \
    --policy-name default \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --query Policy \
    --output text > policy.txt

aws kms put-key-policy \
```

```
    --policy-name default \
    --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
    --policy file://policy.txt
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS KMS API 參考》中的 PutKeyPolicy。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetKeyPolicy。

get-key-rotation-status

以下程式碼範例顯示如何使用 get-key-rotation-status。

AWS CLI

擷取 KMS 金鑰的輪換狀態。

下列get-key-rotation-status範例會傳回指定 KMS 金鑰輪換狀態的相關資訊，包括是否啟用自動輪換、輪換期間和下一個排定的輪換日期。您可以在客戶受管 KMS 金鑰和 AWS 受管 KMS 金鑰上使用此命令。不過，所有 AWS 受管 KMS 金鑰都會每年自動輪換。

```
aws kms get-key-rotation-status \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

輸出：

```
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyRotationEnabled": true,
    "NextRotationDate": "2024-02-14T18:14:33.587000+00:00",
    "RotationPeriodInDays": 365
}
```

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的輪換金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetKeyRotationStatus。

get-parameters-for-import

以下程式碼範例顯示如何使用 get-parameters-for-import。

AWS CLI

若要取得將金鑰材料匯入 KMS 金鑰所需的項目

下列get-parameters-for-import範例會取得將金鑰材料匯入 KMS 金鑰所需的公有金鑰和匯入字符。當您使用 import-key-material命令時，請務必使用由相同get-parameters-for-import命令中傳回的公有金鑰加密的匯入字符和金鑰材料。此外，您在此命令中指定的包裝演算法必須是您使用公有金鑰加密金鑰材料的演算法。

若要指定 KMS 金鑰，請使用 key-id 參數。此範例使用金鑰 ID，但您可以在此命令中使用金鑰 ID 或金鑰 ARN。

```
aws kms get-parameters-for-import \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --wrapping-algorithm RSAES_OAEP_SHA_256 \
    --wrapping-key-spec RSA_2048
```

輸出：

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "PublicKey": "<public key base64 encoded data>",
    "ImportToken": "<import token base64 encoded data>",
    "ParametersValidTo": 1593893322.32
}
```

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的下載公有金鑰和匯入字符。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetParametersForImport。

get-public-key

以下程式碼範例顯示如何使用 get-public-key。

AWS CLI

範例 1：下載非對稱 KMS 金鑰的公有金鑰

下列get-public-key範例會下載非對稱 KMS 金鑰的公有金鑰。

除了傳回公有金鑰之外，輸出還包含您在 AWS KMS 外安全地使用公有金鑰所需的資訊，包括金鑰用量和支援的加密演算法。

```
aws kms get-public-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

輸出：

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "PublicKey": "jANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAl5epvg1/
QtJhxSi2g9SDEVg8QV/...",
    "CustomerMasterKeySpec": "RSA_4096",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "EncryptionAlgorithms": [
        "RSAES_OAEP_SHA_1",
        "RSAES_OAEP_SHA_256"
    ]
}
```

如需在 AWS KMS 中使用非對稱 KMS 金鑰的詳細資訊，請參閱 Key Management Service 開發人員指南中的 AWS KMS 中的非對稱金鑰。 AWS 範例 2：將公有金鑰轉換為 DER 格式 (Linux 和 macOS)

下列get-public-key範例會下載非對稱 KMS 金鑰的公有金鑰，並將其儲存在 DER 檔案中。

當您在 CLI AWS 中使用 get-public-key命令時，它會傳回 BaseBase64-encoded X.509 公有金鑰。此範例會以文字的形式取得 PublicKey 屬性的值。它會Base64-decodes，PublicKey並將其儲存在 public_key.der 檔案中。output 參數會以文字形式傳回輸出，而非 JSON。--query 參數只會取得 PublicKey 屬性，而不是您在 AWS KMS 外安全地使用公有金鑰所需的屬性。

執行此命令之前，請將範例金鑰 ID 取代為來自您 AWS 帳戶的有效金鑰 ID。

```
aws kms get-public-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --output text \
    --query PublicKey | base64 --decode > public_key.der
```

此命令不會產生輸出。

如需在 AWS KMS 中使用非對稱 KMS 金鑰的詳細資訊，請參閱 Key Management Service 開發人員指南中的 AWS KMS 中的非對稱金鑰。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPublicKey。

`import-key-material`

以下程式碼範例顯示如何使用 `import-key-material`。

AWS CLI

將金鑰材料匯入 KMS 金鑰

下列`import-key-material`範例會將金鑰材料上傳至不使用金鑰材料建立的 KMS 金鑰。KMS 金鑰的金鑰狀態必須為 `PendingImport`。

此命令使用您使用`get-parameters-for-import`命令傳回的公有金鑰加密的金鑰材料。它也會使用來自相同`get-parameters-for-import`命令的匯入字符。

`expiration-model` 參數指出金鑰材料會在 `valid-to` 參數指定的日期和時間自動過期。當金鑰材料過期時， AWS KMS 會刪除金鑰材料，KMS 金鑰的金鑰狀態會變更為 ，`Pending import`而 KMS 金鑰會變成無法使用。若要還原 KMS 金鑰，您必須重新匯入相同的金鑰材料。若要使用不同的金鑰材料，您必須建立新的 KMS 金鑰。

執行此命令之前，請將範例金鑰 ID 取代為 AWS 帳戶的有效金鑰 ID 或金鑰 ARN。

```
aws kms import-key-material \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --encrypted-key-material fileb://EncryptedKeyMaterial.bin \
    --import-token fileb://ImportToken.bin \
    --expiration-model KEY_MATERIAL_EXPIRES \
    --valid-to 2021-09-21T19:00:00Z
```

輸出：

```
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

如需匯入金鑰材料的詳細資訊，請參閱 Key Management Service 開發人員指南中的匯入金鑰材料。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ImportKeyMaterial。

list-aliases

以下程式碼範例顯示如何使用 list-aliases。

AWS CLI

範例 1：列出 AWS 帳戶和區域中的所有別名

下列範例使用 list-aliases命令列出 AWS 帳戶預設區域中的所有別名。輸出包含與 AWS 受管 KMS 金鑰和客戶受管 KMS 金鑰相關聯的別名。

```
aws kms list-aliases
```

輸出：

```
{
    "Aliases": [
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/testKey",
            "AliasName": "alias/testKey",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/FinanceDept",
            "AliasName": "alias/FinanceDept",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321"
        },
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/dynamodb",
            "AliasName": "alias/aws/dynamodb",
            "TargetKeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
        },
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/ebs",
            "AliasName": "alias/aws/ebs",
            "TargetKeyId": "0987ab65-43cd-21ef-09ab-87654321cdef"
        },
```

```
        ...
    ]
}
```

範例 2：列出特定 KMS 金鑰的所有別名

下列範例使用 `list-aliases`命令及其`key-id`參數來列出與特定 KMS 金鑰相關聯的所有別名。

每個別名只會與一個 KMS 金鑰相關聯，但 KMS 金鑰可以有多個別名。此命令非常有用，因為 AWS KMS 主控台只會為每個 KMS 金鑰列出一個別名。若要尋找 KMS 金鑰的所有別名，您必須使用 `list-aliases`命令。

此範例使用 `--key-id` 參數 KMS 金鑰的金鑰 ID，但您可以在此命令中使用金鑰 ID、金鑰 ARN、別名名稱或別名 ARN。

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

輸出：

```
{
    "Aliases": [
        {
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/oregon-test-key",
            "AliasName": "alias/oregon-test-key"
        },
        {
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/project121-test",
            "AliasName": "alias/project121-test"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的使用別名。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAliases。

list-grants

以下程式碼範例顯示如何使用 `list-grants`。

AWS CLI

檢視 AWS KMS 金鑰上的授予

下列`list-grants`範例顯示您帳戶中 Amazon DynamoDB 指定 AWS 受管 KMS 金鑰上的所有授予。此授與可讓 DynamoDB 代表您使用 KMS 金鑰來加密 DynamoDB 資料表，然後再寫入磁碟。您可以使用像這樣的命令來檢視 AWS 帳戶和區域中 AWS 受管 KMS 金鑰和客戶受管 KMS 金鑰的授予。

此命令使用 `key-id` 參數搭配金鑰 ID 來識別 KMS 金鑰。您可以使用金鑰 ID 或金鑰 ARN 來識別 KMS 金鑰。若要取得 AWS 受管 KMS 金鑰的金鑰 ID 或金鑰 ARN，請使用 `list-keys`或 `list-aliases`命令。

```
aws kms list-grants \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

輸出顯示授予授予 Amazon DynamoDB 許可，以使用 KMS 金鑰進行密碼編譯操作，並授予其許可，以檢視 KMS 金鑰 (`DescribeKey`) 的詳細資訊並淘汰授予 ()`RetireGrant`。`EncryptionContextSubset` 限制條件會將這些許可限制為包含指定加密內容對的請求。因此，授予中的許可僅在指定的帳戶和 DynamoDB 資料表上有效。

```
{
    "Grants": [
        {
            "Constraints": {
                "EncryptionContextSubset": {
                    "aws:dynamodb:subscriberId": "123456789012",
                    "aws:dynamodb:tableName": "Services"
                }
            },
            "IssuingAccount": "arn:aws:iam::123456789012:root",
            "Name": "8276b9a6-6cf0-46f1-b2f0-7993a7f8c89a",
            "Operations": [
                "Decrypt",
                "Encrypt",
                "GenerateDataKey",
                "ReEncryptFrom",
                "ReEncryptTo",
                "RetireGrant",
                "DescribeKey"
            ],
```

```
            "GrantId":
 "1667b97d27cf748cf05b487217dd4179526c949d14fb3903858e25193253fe59",
            "KeyId": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "RetiringPrincipal": "dynamodb.us-west-2.amazonaws.com",
            "GranteePrincipal": "dynamodb.us-west-2.amazonaws.com",
            "CreationDate": "2021-05-13T18:32:45.144000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 AWS KMS 中的授權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListGrants。

list-key-policies

以下程式碼範例顯示如何使用 list-key-policies。

AWS CLI

取得 KMS 金鑰的金鑰政策名稱

下列list-key-policies範例會取得範例帳戶和區域中客戶受管金鑰的金鑰政策名稱。您可以使用此命令來尋找 AWS 受管金鑰和客戶受管金鑰的金鑰政策名稱。

由於唯一有效的金鑰政策名稱是 default，因此此命令沒有用。

若要指定 KMS 金鑰，請使用 key-id 參數。此範例使用金鑰 ID 值，但您可以在此命令中使用金鑰 ID 或金鑰 ARN。

```
aws kms list-key-policies \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

輸出：

```
{
    "PolicyNames": [
    "default"
    ]
}
```

如需 AWS KMS 金鑰政策的詳細資訊，請參閱 Key Management Service 開發人員指南中的在 AWS KMS 中使用金鑰政策。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListKeyPolicies。

list-key-rotations

以下程式碼範例顯示如何使用 list-key-rotations。

AWS CLI

擷取所有已完成金鑰材料輪換的相關資訊

下列list-key-rotations範例列出指定 KMS 金鑰之所有已完成金鑰材料輪換的相關資訊。

```
aws kms list-key-rotations \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

輸出：

```
{
    "Rotations": [
        {
            "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "RotationDate": "2024-03-02T10:11:36.564000+00:00",
            "RotationType": "AUTOMATIC"
        },
        {
            "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "RotationDate": "2024-04-05T15:14:47.757000+00:00",
            "RotationType": "ON_DEMAND"
        }
    ],
    "Truncated": false
}
```

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的輪換金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListKeyRotations。

list-keys

以下程式碼範例顯示如何使用 list-keys。

AWS CLI

取得帳戶和區域中的 KMS 金鑰

下列list-keys範例取得 帳戶和區域中的 KMS 金鑰。此命令會同時傳回 AWS 受管金鑰和客戶受管金鑰。

```
aws kms list-keys
```

輸出：

```
{
    "Keys": [
        {
            "KeyArn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        {
            "KeyArn": "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
            "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321"
        },
        {
            "KeyArn": "arn:aws:kms:us-
east-2:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
            "KeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
        }
    ]
}
```

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的檢視AWS 金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListKeys。

list-resource-tags

以下程式碼範例顯示如何使用 list-resource-tags。

AWS CLI

取得 KMS 金鑰上的標籤

下列list-resource-tags範例會取得 KMS 金鑰的標籤。若要在 KMS 金鑰上新增或取代資源標籤，請使用 tag-resource命令。輸出顯示此 KMS 金鑰有兩個資源標籤，每個標籤都有金鑰和值。

若要指定 KMS 金鑰，請使用 key-id 參數。此範例使用金鑰 ID 值，但您可以在此命令中使用金鑰 ID 或金鑰 ARN。

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

輸出：

```
{
    "Tags": [
    {
        "TagKey": "Dept",
        "TagValue": "IT"
    },
    {
        "TagKey": "Purpose",
        "TagValue": "Test"
    }
    ],
    "Truncated": false
}
```

如需在 AWS KMS 中使用標籤的詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的標記金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourceTags。

list-retirable-grants

以下程式碼範例顯示如何使用 list-retirable-grants。

AWS CLI

檢視委託人可以淘汰的授予

下列list-retirable-grants範例顯示ExampleAdmin使用者可以在 AWS 帳戶和區域中的 KMS 金鑰淘汰的所有授予。您可以使用像這樣的命令來檢視任何帳戶主體可以在 AWS 帳戶和區域中的 KMS 金鑰上淘汰的授予。

必要retiring-principal參數的值必須是帳戶、使用者或角色的 Amazon Resource Name (ARN)。

您無法在此命令retiring-principal中指定 值的服務，即使服務可以是淘汰的委託人。若要尋找特定服務為淘汰委託人的授予，請使用 list-grants命令。

輸出顯示ExampleAdmin使用者有權淘汰帳戶和區域中兩個不同 KMS 金鑰的授予。除了淘汰委託人之外，帳戶還具有淘汰帳戶中任何授予的許可。

```
aws kms list-retirable-grants \
    --retiring-principal arn:aws:iam::111122223333:user/ExampleAdmin
```

輸出：

```
{
    "Grants": [
        {
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "GrantId":
 "156b69c63cb154aa21f59929ff19760717be8d9d82b99df53e18b94a15a5e88e",
            "Name": "",
            "CreationDate": 2021-01-14T20:17:36.419000+00:00,
            "GranteePrincipal": "arn:aws:iam::111122223333:user/ExampleUser",
            "RetiringPrincipal": "arn:aws:iam::111122223333:user/ExampleAdmin",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "Operations": [
                "Encrypt"
            ],
            "Constraints": {
                "EncryptionContextSubset": {
                    "Department": "IT"
                }
            }
        },
        {
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
            "GrantId":
 "8c94d1f12f5e69f440bae30eaec9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",
            "Name": "",
            "CreationDate": "2021-02-02T19:49:49.638000+00:00",
```

```
            "GranteePrincipal": "arn:aws:iam::111122223333:role/ExampleRole",
            "RetiringPrincipal": "arn:aws:iam::111122223333:user/ExampleAdmin",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "Operations": [
                "Decrypt"
            ],
            "Constraints": {
                "EncryptionContextSubset": {
                    "Department": "IT"
                }
            }
        }
    ],
    "Truncated": false
}
```

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 AWS KMS 中的授權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRetirableGrants。

put-key-policy

以下程式碼範例顯示如何使用 put-key-policy。

AWS CLI

變更 KMS 金鑰的金鑰政策

下列put-key-policy範例會變更客戶受管金鑰的金鑰政策。

若要開始，請建立金鑰政策並將其儲存在本機 JSON 檔案中。在此範例中，檔案為 key_policy.json。您也可以將金鑰政策指定為 policy 參數的字串值。

此金鑰政策中的第一個陳述式提供 AWS 帳戶使用 IAM 政策控制 KMS 金鑰存取的許可。第二個陳述式提供test-user使用者在 KMS 金鑰上執行 describe-key和 list-keys命令的許可。

key_policy.json 的內容：

```
{
    "Version" : "2012-10-17",
    "Id" : "key-default-1",
    "Statement" : [
```

```
        {
            "Sid" : "Enable IAM User Permissions",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:root"
            },
            "Action" : "kms:*",
            "Resource" : "*"
        },
        {
            "Sid" : "Allow Use of Key",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:user/test-user"
            },
            "Action" : [
                "kms:DescribeKey",
                "kms:ListKeys"
            ],
            "Resource" : "*"
        }
    ]
}
```

為了識別 KMS 金鑰，此範例使用金鑰 ID，但您也可以使用金鑰 ARN。若要指定金鑰政策，命令會使用 policy 參數。若要指出政策位於檔案中，會使用必要的file://字首。識別所有支援的作業系統上的檔案時，需要此字首。最後，命令會使用 值為 的 policy-name 參數default。如果未指定政策名稱，預設值為 default。唯一有效的值為 default。

```
aws kms put-key-policy \
    --policy-name default \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --policy file://key_policy.json
```

此命令不會產生任何輸出。若要驗證命令是否有效，請使用 get-key-policy命令。下列範例命令會取得相同 KMS 金鑰的金鑰政策。值為 的 output 參數會text傳回易於讀取的文字格式。

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

輸出：

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

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的變更金鑰政策。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutKeyPolicy。

re-encrypt

以下程式碼範例顯示如何使用 re-encrypt。

AWS CLI

範例 1：在不同的對稱 KMS 金鑰 (Linux 和 macOS) 下重新加密訊息。

下列re-encrypt命令範例示範使用 AWS CLI 重新加密資料的建議方法。

在 file.In 中提供 --ciphertext-blob 參數值的加密文字，使用 fileb://字首，告知 CLI 從二進位檔案讀取資料。如果檔案不在目前的目錄中，請輸入檔案的完整路徑。如需有關從檔案讀

取 AWS CLI 參數值的詳細資訊，請參閱《 AWS 命令列界面使用者指南》中的從檔案載入 AWS CLI 參數，以及《 AWS 命令列工具部落格》中的本機檔案參數最佳實務。指定解密加密文字的來源 KMS 金鑰。使用對稱加密 KMS 金鑰解密時，不需要 `--source-key-id` 參數。 AWS KMS 可以取得用於加密加密加密加密加密文字 Blob 中中繼資料資料的 KMS 金鑰。但是指定您正在使用的 KMS 金鑰永遠是最佳實務。此實務可確保您使用您想要的 KMS 金鑰，並防止您不小心使用您不信任的 KMS 金鑰解密加密文字。指定目的地 KMS 金鑰，這會重新加密資料。 `--destination-key-id` 參數一律為必要。此範例使用金鑰 ARN，但您可以使用任何有效的金鑰識別符。請求純文字輸出做為文字值。 `--query` 參數會告知 CLI 僅從輸出取得 `Plaintext` 欄位的值。`--output` 參數會以文字傳回輸出。Base64 解碼純文字，並將其儲存在檔案中。下列範例會將 `Plaintext` 參數的值輸送 (|) 至 Base64 公用程式，以解碼該參數。然後，它會將解碼的輸出重新導向 (>) 至 `ExamplePlaintext` 檔案。

執行此命令之前，請將範例金鑰 IDs 取代為 AWS 帳戶中有效的金鑰識別符。

```
aws kms re-encrypt \
    --ciphertext-blob fileb://ExampleEncryptedFile \
    --source-key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --destination-key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
    --query CiphertextBlob \
    --output text | base64 --decode > ExampleReEncryptedFile
```

此命令不會產生輸出。來自 `re-encrypt`命令的輸出是 base64 解碼並儲存在檔案中。

如需詳細資訊，請參閱 AWS Key Management Service API 參考中的 ReEncrypt。

範例 2：在不同的對稱 KMS 金鑰 (Windows 命令提示） 下重新加密加密的訊息。

下列`re-encrypt`命令範例與上一個命令範例相同，但它使用 `certutil`公用程式來對純文字資料進行 Base64-decode。此程序需要兩個命令，如下列範例所示。

執行此命令之前，請將範例金鑰 ID 取代為來自您 AWS 帳戶的有效金鑰 ID。

```
aws kms re-encrypt ^
    --ciphertext-blob fileb://ExampleEncryptedFile ^
    --source-key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^
    --destination-key-id 0987dcba-09fe-87dc-65ba-ab0987654321 ^
    --query CiphertextBlob ^
    --output text > ExampleReEncryptedFile.base64
```

然後使用 `certutil`公用程式

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

輸出：

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

如需詳細資訊，請參閱 AWS Key Management Service API 參考中的 ReEncrypt。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReEncrypt。

retire-grant

以下程式碼範例顯示如何使用 retire-grant。

AWS CLI

淘汰客戶主金鑰的授予

下列retire-grant範例會從 KMS 金鑰刪除授予。

下列範例命令會指定 grant-id和 key-id 參數。key-id 參數的值必須是 KMS 金鑰的金鑰 ARN。

```
aws kms retire-grant \
    --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

此命令不會產生輸出。若要確認授予已淘汰，請使用 list-grants命令。

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的淘汰和撤銷授予。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RetireGrant。

revoke-grant

以下程式碼範例顯示如何使用 revoke-grant。

AWS CLI

撤銷客戶主金鑰的授予

下列revoke-grant範例會從 KMS 金鑰刪除授予。下列範例命令會指定 grant-id和 key-id 參數。key-id 參數的值可以是 KMS 金鑰的金鑰 ID 或金鑰 ARN。

```
aws kms revoke-grant \
    --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

此命令不會產生輸出。若要確認授予已撤銷，請使用 list-grants命令。

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的淘汰和撤銷授予。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RevokeGrant。

rotate-key-on-demand

以下程式碼範例顯示如何使用 rotate-key-on-demand。

AWS CLI

執行 KMS 金鑰的隨需輪換

下列rotate-key-on-demand範例會立即為指定的 KMS 金鑰啟動金鑰材料的輪換。

```
aws kms rotate-key-on-demand \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

輸出：

```
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的如何執行隨需金鑰輪換。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RotateKeyOnDemand。

schedule-key-deletion

以下程式碼範例顯示如何使用 schedule-key-deletion。

AWS CLI

排程刪除客戶受管 KMS 金鑰。

下列schedule-key-deletion範例排程要在 15 天內刪除指定的客戶受管 KMS 金鑰。

--key-id 參數可識別 KMS 金鑰。此範例使用金鑰 ARN 值，但您可以使用金鑰 ID 或 KMS 金鑰的 ARN。 --pending-window-in-days 參數會指定 7-30 天等待期間的長度。根據預設，等待期間為 30 天。此範例會指定值 15，指示 AWS 命令完成後 15 天永久刪除 KMS 金鑰。

```
aws kms schedule-key-deletion \
    --key-id arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --pending-window-in-days 15
```

回應包含金鑰 ARN、金鑰狀態、等待期間 (PendingWindowInDays) 和刪除日期，以 Unix 時間表示。若要以當地時間檢視刪除日期，請使用 AWS KMS 主控台。金鑰PendingDeletion狀態為的 KMS 金鑰無法用於密碼編譯操作。

```
{
    "KeyId": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "DeletionDate": "2022-06-18T23:43:51.272000+00:00",
    "KeyState": "PendingDeletion",
    "PendingWindowInDays": 15
}
```

如需詳細資訊，請參閱 Key Management Service 開發人員指南中的刪除AWS 金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ScheduleKeyDeletion。

sign

以下程式碼範例顯示如何使用 sign。

AWS CLI

範例 1：產生訊息的數位簽章

下列sign範例會產生短訊息的密碼編譯簽章。命令的輸出包含 base-64 編碼Signature欄位，您可以使用 verify命令進行驗證。

您必須指定要簽署的訊息，以及非對稱 KMS 金鑰支援的簽署演算法。若要取得 KMS 金鑰的簽署演算法，請使用 describe-key命令。

在 AWS CLI v2 中， message 參數的值必須是 Base64-encoded。或者，您可以將訊息儲存在檔案中，並使用 fileb://字首，告知 AWS CLI 從檔案讀取二進位資料。

執行此命令之前，請將範例金鑰 ID 取代為來自您 AWS 帳戶的有效金鑰 ID。金鑰 ID 必須代表金鑰用量為 SIGN_VERIFY 的非對稱 KMS 金鑰。

```
msg=(echo 'Hello World' | base64)

aws kms sign \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://UnsignedMessage \
    --message-type RAW \
    --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256
```

輸出：

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "Signature": "ABCDEFhpyVYyTxbafE74ccSvEJLJr3zuoV1Hfymz4qv+/
fxmxNLA7SE1SiF8lHw80fKZZ3bJ...",
    "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

如需有關在 AWS KMS 中使用非對稱 KMS 金鑰的詳細資訊，請參閱 Key Management Service 開發人員指南中的 AWS KMS 中的非對稱金鑰。 AWS

範例 2：在 檔案中儲存數位簽章 (Linux 和 macOs)

下列sign範例會為儲存在本機檔案中的短訊息產生密碼編譯簽章。命令也會從回應取得 Signature 屬性，Base64-decodes，並將其儲存在 ExampleSignature 檔案中。您可以在驗證簽章的verify命令中使用簽章檔案。

sign 命令需要 Base64-encoded的訊息，以及非對稱 KMS 金鑰支援的簽署演算法。若要取得 KMS 金鑰支援的簽署演算法，請使用 describe-key命令。

執行此命令之前，請將範例金鑰 ID 取代為來自您 AWS 帳戶的有效金鑰 ID。金鑰 ID 必須代表金鑰用量為 SIGN_VERIFY 的非對稱 KMS 金鑰。

```
echo 'hello world' | base64 > EncodedMessage

aws kms sign \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://EncodedMessage \
    --message-type RAW \
    --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256 \
    --output text \
    --query Signature | base64 --decode > ExampleSignature
```

此命令不會產生輸出。此範例會擷取輸出的 `Signature` 屬性，並將其儲存在檔案中。

如需在 AWS KMS 中使用非對稱 KMS 金鑰的詳細資訊，請參閱 Key Management Service 開發人員指南中的 AWS KMS 中的非對稱金鑰。 AWS

- 如需 API 詳細資訊，請參閱登入AWS CLI 命令參考。

`tag-resource`

以下程式碼範例顯示如何使用 `tag-resource`。

AWS CLI

將標籤新增至 KMS 金鑰

下列`tag-resource`範例會將 `"Purpose":"Test"`和 `"Dept":"IT"` 標籤新增至客戶受管 KMS 金鑰。您可以使用這類標籤來標記 KMS 金鑰，並建立 KMS 金鑰類別以進行許可和稽核。

若要指定 KMS 金鑰，請使用 `key-id` 參數。此範例使用金鑰 ID 值，但您可以在此命令中使用金鑰 ID 或金鑰 ARN。

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

此命令不會產生輸出。若要檢視 AWS KMS KMS 金鑰上的標籤，請使用 `list-resource-tags`命令。

如需在 AWS KMS 中使用標籤的詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的標記金鑰。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從 KMS 金鑰刪除標籤

下列untag-resource範例會從客戶受管 KMS "Purpose"金鑰刪除具有 金鑰的標籤。

若要指定 KMS 金鑰，請使用 key-id 參數。此範例使用金鑰 ID 值，但您可以在此命令中使用金鑰 ID 或金鑰 ARN。執行此命令之前，請將範例金鑰 ID 取代為來自您 AWS 帳戶的有效金鑰 ID。

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

此命令不會產生輸出。若要檢視 AWS KMS KMS 金鑰上的標籤，請使用 list-resource-tags命令。

如需在 AWS KMS 中使用標籤的詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的標記金鑰。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-alias

以下程式碼範例顯示如何使用 update-alias。

AWS CLI

將別名與不同的 KMS 金鑰建立關聯

下列update-alias範例會將別名alias/test-key與不同的 KMS 金鑰建立關聯。

--alias-name 參數指定別名。別名名稱值必須以 開頭alias/。 --target-key-id 參數會指定要與別名建立關聯的 KMS 金鑰。您不需要為別名指定目前的 KMS 金鑰。

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

此命令不會產生輸出。若要尋找別名，請使用 `list-aliases`命令。

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的更新別名。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateAlias。

update-custom-key-store

以下程式碼範例顯示如何使用 `update-custom-key-store`。

AWS CLI

範例 1：編輯自訂金鑰存放區的易記名稱

下列`update-custom-key-store`範例會變更自訂金鑰存放區的名稱。此範例適用於 an AWS CloudHSM 金鑰存放區或外部金鑰存放區。

使用 `custom-key-store-id`來識別金鑰存放區。使用 `new-custom-key-store-name` 參數來指定新的易記名稱。

若要更新 an AWS CloudHSM 金鑰存放區的易記名稱，您必須先中斷連接金鑰存放區，例如使用`disconnect-custom-key-store`命令。您可以在外部金鑰存放區連線或中斷連線時更新其易記名稱。若要尋找自訂金鑰存放區的連線狀態，請使用 `describe-custom-key-store`命令。

```
aws kms update-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0 \
    --new-custom-key-store-name ExampleKeyStore
```

此命令不會傳回任何資料。若要驗證命令是否正常運作，請使用 `describe-custom-key-stores`命令。

如需更新 an AWS CloudHSM 金鑰存放區的詳細資訊，請參閱 Key AWS Management Service 開發人員指南中的編輯 AWS CloudHSM 金鑰存放區設定。

如需更新外部金鑰存放區的詳細資訊，請參閱 Key Management Service 開發人員指南中的編輯外部金鑰存放區屬性。 AWS

範例 2：編輯 an AWS CloudHSM 金鑰存放區的 kmsuser 密碼

下列update-custom-key-store範例會將kmsuser密碼的值更新為與指定金鑰存放區相關聯之 CloudHSM 叢集kmsuser中 的目前密碼。此命令不會變更叢集kmsuser的密碼。它只會告訴 AWS KMS 目前的密碼。如果 KMS 沒有目前的kmsuser密碼，則無法連線到 AWS CloudHSM 金鑰存放區。

注意：在更新 an AWS CloudHSM 金鑰存放區之前，您必須中斷連線。使用 disconnect-custom-key-store 命令。命令完成後，您可以重新連接 AWS CloudHSM 金鑰存放區。使用 connect-custom-key-store 命令。

```
aws kms update-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0 \
    --key-store-password ExamplePassword
```

此命令不會傳回任何輸出。若要驗證變更是否有效，請使用 describe-custom-key-stores命令。

如需更新 an AWS CloudHSM 金鑰存放區的詳細資訊，請參閱 Key AWS Management Service 開發人員指南中的編輯 AWS CloudHSM 金鑰存放區設定。

範例 3：編輯 an AWS CloudHSM 金鑰存放區的 AWS CloudHSM 叢集

下列範例會將與 an AWS CloudHSM 金鑰存放區相關聯的 AWS CloudHSM 叢集變更為相關叢集，例如相同叢集的不同備份。

注意：在更新 an AWS CloudHSM 金鑰存放區之前，您必須中斷連線。使用 disconnect-custom-key-store 命令。命令完成後，您可以重新連接 AWS CloudHSM 金鑰存放區。使用 connect-custom-key-store 命令。

```
aws kms update-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0 \
    --cloud-hsm-cluster-id cluster-1a23b4cdefg
```

此命令不會傳回任何輸出。若要驗證變更是否有效，請使用 describe-custom-key-stores命令。

如需更新 an AWS CloudHSM 金鑰存放區的詳細資訊，請參閱 Key AWS Management Service 開發人員指南中的編輯 AWS CloudHSM 金鑰存放區設定。

範例 4：編輯外部金鑰存放區的代理身分驗證憑證

下列範例會更新外部金鑰存放區的代理身分驗證憑證。您必須同時指定 raw-secret-access-key和 access-key-id，即使您只變更其中一個值。您可以使用此功能來修正無效的登入資料，或在外部金鑰存放區代理輪換登入資料時變更登入資料。

在外部金鑰存放區上建立 AWS KMS 的代理身分驗證憑證。然後，使用此命令向 AWS KMS 提供登入資料。 AWS KMS 使用此登入資料來簽署其對外部金鑰存放區代理的請求。

您可以在外部金鑰存放區連線或中斷連線時更新代理身分驗證憑證。若要尋找自訂金鑰存放區的連線狀態，請使用 describe-custom-key-store命令。

```
aws kms update-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0 \
    --xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,
 RawSecretAccessKey=DXjSUawnel2fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

此命令不會傳回任何輸出。若要驗證變更是否有效，請使用 describe-custom-key-stores命令。

如需更新外部金鑰存放區的詳細資訊，請參閱 Key Management Service 開發人員指南中的編輯外部金鑰存放區屬性。 AWS

範例 5：編輯外部金鑰存放區的代理連線

下列範例會將外部金鑰存放區代理連線選項從公有端點連線變更為 VPC 端點服務連線。除了變更xks-proxy-connectivity值之外，您還必須變更xks-proxy-uri-endpoint值，以反映與 VPC 端點服務相關聯的私有 DNS 名稱。您還必須新增 xks-proxy-vpc-endpoint-service-name值。

注意：在更新外部存放區的代理連線之前，您必須中斷連線。使用 disconnect-custom-key-store 命令。命令完成後，您可以使用 connect-custom-key-store命令重新連接外部金鑰存放區。

```
aws kms update-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0 \
    --xks-proxy-connectivity VPC_ENDPOINT_SERVICE \
    --xks-proxy-uri-endpoint "https://myproxy-private.xks.example.com" \
    --xks-proxy-vpc-endpoint-service-name "com.amazonaws.vpce.us-east-1.vpce-svc-
example"
```

此命令不會傳回任何輸出。若要驗證變更是否有效，請使用 describe-custom-key-stores命令。

如需更新外部金鑰存放區的詳細資訊，請參閱 Key Management Service 開發人員指南中的編輯外部金鑰存放區屬性。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateCustomKeyStore。

update-key-description

以下程式碼範例顯示如何使用 update-key-description。

AWS CLI

範例 1：新增或變更描述到客戶受管 KMS 金鑰

下列update-key-description範例會將描述新增至客戶受管 KMS 金鑰。您可以使用相同的命令來變更現有的描述。

--key-id 參數識別 命令中的 KMS 金鑰。此範例使用金鑰 ARN 值，但您可以使用金鑰 ID 或 KMS 金鑰的金鑰 ARN。 --description 參數會指定新的描述。此參數的值會取代 KMS 金鑰的目前描述，如果有的話。

```
aws kms update-key-description \
    --key-id arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --description "IT Department test key"
```

此命令不會產生輸出。若要檢視 KMS 金鑰的描述，請使用 describe-key命令。

如需詳細資訊，請參閱 AWS Key Management Service API 參考中的 UpdateKeyDescription。

範例 2：刪除客戶受管 KMS 金鑰的描述

下列update-key-description範例會刪除客戶受管 KMS 金鑰的描述。

--key-id 參數識別 命令中的 KMS 金鑰。此範例使用金鑰 ID 值，但您可以使用 KMS 金鑰的金鑰 ID 或金鑰 ARN。具有空字串值 ('') 的 --description 參數會刪除現有的描述。

```
aws kms update-key-description \
    --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
```

```
    --description ''
```

此命令不會產生輸出。若要檢視 KMS 金鑰的描述，請使用 describe-key 命令。

如需詳細資訊，請參閱 AWS Key Management Service API 參考中的 UpdateKeyDescription。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateKeyDescription。

`verify-mac`

以下程式碼範例顯示如何使用 `verify-mac`。

AWS CLI

範例 1：驗證 HMAC

下列`verify-mac`命令會驗證特定訊息的 HMAC、HMAC KMS 金鑰和 MAC 演算法。MacValid 值中的 'true' 值表示 HMAC 有效。

在 AWS CLI v2 中， `message` 參數的值必須是 Base64-encoded。或者，您可以將訊息儲存在檔案中，並使用 `fileb://`字首，告知 AWS CLI 從檔案讀取二進位資料。

您指定的 MAC 不能是 base64 編碼。如需對`generate-mac`命令傳回的 MAC 進行解碼的說明，請參閱`generate-mac`命令範例。

執行此命令之前，請將範例金鑰 ID 取代為來自您 AWS 帳戶的有效金鑰 ID。金鑰 ID 必須代表金鑰使用量為 的 HMAC KMS 金鑰`GENERATE_VERIFY_MAC`。

```
msg=(echo 'Hello World' | base64)

aws kms verify-mac \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://Message \
    --mac-algorithm HMAC_SHA_384 \
    --mac fileb://ExampleMac
```

輸出：

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

```
    "MacValid": true,
    "MacAlgorithm": "HMAC_SHA_384"
}
```

如需在 KMS AWS 中使用 HMAC KMS 金鑰的詳細資訊，請參閱《 金鑰管理服務開發人員指南》中的 AWS KMS 中的 HMAC 金鑰。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 VerifyMac。

verify

以下程式碼範例顯示如何使用 verify。

AWS CLI

驗證數位簽章

下列verify命令會驗證簡短的 Base64-encoded訊息的密碼編譯簽章。金鑰 ID、訊息、訊息類型和簽署演算法必須與用來簽署訊息的演算法相同。

在 AWS CLI v2 中， message 參數的值必須是 Base64-encoded。或者，您可以將訊息儲存在檔案中，並使用 fileb://字首，告知 AWS CLI 從檔案讀取二進位資料。

您指定的簽章不能是 base64 編碼。如需對sign命令傳回的簽章進行解碼的說明，請參閱sign命令範例。

命令的輸出包含布林值SignatureValid欄位，指出簽章已驗證。如果簽章驗證失敗，verify命令也會失敗。

執行此命令之前，請將範例金鑰 ID 取代為來自您 AWS 帳戶的有效金鑰 ID。

```
aws kms verify \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://EncodedMessage \
    --message-type RAW \
    --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256 \
    --signature fileb://ExampleSignature
```

輸出：

```
{
```

```
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "SignatureValid": true,
    "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

如需在 KMS 中使用非對稱 AWS KMS 金鑰的詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的使用非對稱金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的驗證。

使用 的 Lake Formation 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Lake Formation 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`add-lf-tags-to-resource`

以下程式碼範例顯示如何使用 `add-lf-tags-to-resource`。

AWS CLI

將一或多個 LF 標籤連接至現有資源

下列`add-lf-tags-to-resource`範例會將指定的 LF 標籤連接至資料表資源。

```
aws lakeformation add-lf-tags-to-resource \
    --cli-input-json file://input.json
```

`input.json` 的內容：

```
{
    "CatalogId": "123456789111",
    "Resource": {
        "Table": {
            "CatalogId": "123456789111",
            "DatabaseName": "tpc",
            "Name": "dl_tpc_promotion"
        }
    },
    "LFTags": [{
        "CatalogId": "123456789111",
        "TagKey": "usergroup",
        "TagValues": [
            "analyst"
        ]
    }]
}
```

輸出：

```
{
    "Failures": []
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的將 LF 標籤指派給 Data Catalog 資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddLfTagsToResource。

batch-grant-permissions

以下程式碼範例顯示如何使用 batch-grant-permissions。

AWS CLI

將資源的許可大量授予委託人

下列batch-grant-permissions範例大量將指定資源的存取權授予委託人。

```
aws lakeformation batch-grant-permissions \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "123456789111",
    "Entries": [{
            "Id": "1",
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
developer"
            },
            "Resource": {
                "Table": {
                    "CatalogId": "123456789111",
                    "DatabaseName": "tpc",
                    "Name": "dl_tpc_promotion"
                }
            },
            "Permissions": [
                "ALL"
            ],
            "PermissionsWithGrantOption": [
                "ALL"
            ]
        },
        {
            "Id": "2",
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
developer"
            },
            "Resource": {
                "Table": {
                    "CatalogId": "123456789111",
                    "DatabaseName": "tpc",
                    "Name": "dl_tpc_customer"
                }
            },
            "Permissions": [
                "ALL"
            ],
            "PermissionsWithGrantOption": [
                "ALL"
            ]
        },
```

```
        {
            "Id": "3",
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
business-analyst"
            },
            "Resource": {
                "Table": {
                    "CatalogId": "123456789111",
                    "DatabaseName": "tpc",
                    "Name": "dl_tpc_promotion"
                }
            },
            "Permissions": [
                "ALL"
            ],
            "PermissionsWithGrantOption": [
                "ALL"
            ]
        },
        {
            "Id": "4",
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
developer"
            },
            "Resource": {
                "DataCellsFilter": {
                    "TableCatalogId": "123456789111",
                    "DatabaseName": "tpc",
                    "TableName": "dl_tpc_item",
                    "Name": "developer_item"
                }
            },
            "Permissions": [
                "SELECT"
            ],
            "PermissionsWithGrantOption": []
        }
    ]
}
```

輸出：

```
{
    "Failures": []
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的授予和撤銷 Data Catalog 資源的許可。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGrantPermissions。

batch-revoke-permissions

以下程式碼範例顯示如何使用 batch-revoke-permissions。

AWS CLI

從委託人大量撤銷資源的許可

下列batch-revoke-permissions範例大量撤銷委託人對指定資源的存取權。

```
aws lakeformation batch-revoke-permissions \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "123456789111",
    "Entries": [{
            "Id": "1",
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
developer"
            },
            "Resource": {
                "Table": {
                    "CatalogId": "123456789111",
                    "DatabaseName": "tpc",
                    "Name": "dl_tpc_promotion"
                }
            },
            "Permissions": [
                "ALL"
```

```
            ],
            "PermissionsWithGrantOption": [
                "ALL"
            ]
        },
        {
            "Id": "2",
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
business-analyst"
            },
            "Resource": {
                "Table": {
                    "CatalogId": "123456789111",
                    "DatabaseName": "tpc",
                    "Name": "dl_tpc_promotion"
                }
            },
            "Permissions": [
                "ALL"
            ],
            "PermissionsWithGrantOption": [
                "ALL"
            ]
        }
    ]
}
```

輸出：

```
{
    "Failures": []
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的授予和撤銷 Data Catalog 資源的許可。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchRevokePermissions。

cancel-transaction

以下程式碼範例顯示如何使用 cancel-transaction。

AWS CLI

取消交易

下列cancel-transaction範例會取消交易。

```
aws lakeformation cancel-transaction \
    --transaction-id='b014d972ca8347b89825e33c5774aec4'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的讀取和寫入交易中的資料湖。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelTransaction。

commit-transaction

以下程式碼範例顯示如何使用 commit-transaction。

AWS CLI

遞交交易

下列commit-transaction範例會遞交交易。

```
aws lakeformation commit-transaction \
    --transaction-id='b014d972ca8347b89825e33c5774aec4'
```

輸出：

```
{
    "TransactionStatus": "committed"
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的讀取和寫入交易中的資料湖。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的CommitTransaction。

create-data-cells-filter

以下程式碼範例顯示如何使用 create-data-cells-filter。

AWS CLI

範例 1：建立資料儲存格篩選條件

下列create-data-cells-filter範例會建立資料儲存格篩選條件，以允許根據資料列條件授予特定資料欄的存取權。

```
aws lakeformation create-data-cells-filter \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "TableData": {
        "ColumnNames": ["p_channel_details", "p_start_date_sk", "p_promo_name"],
        "DatabaseName": "tpc",
        "Name": "developer_promotion",
        "RowFilter": {
            "FilterExpression": "p_promo_name='ese'"
        },
        "TableCatalogId": "123456789111",
        "TableName": "dl_tpc_promotion"
    }
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Lake Formation 開發人員指南中的 Lake Formation 中的資料篩選和儲存格層級安全性。 AWS

範例 2：建立資料欄篩選條件

下列create-data-cells-filter範例會建立資料篩選條件，以允許其授予特定資料欄的存取權。

```
aws lakeformation create-data-cells-filter \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "TableData": {
```

```
        "ColumnNames": ["p_channel_details", "p_start_date_sk", "p_promo_name"],
        "DatabaseName": "tpc",
        "Name": "developer_promotion_allrows",
        "RowFilter": {
            "AllRowsWildcard": {}
        },
        "TableCatalogId": "123456789111",
        "TableName": "dl_tpc_promotion"
    }
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Lake Formation 開發人員指南中的 Lake Formation 中的資料篩選和儲存格層級安全性。 AWS

範例 3：使用排除資料欄建立資料篩選條件

下列create-data-cells-filter範例會建立資料篩選條件，以允許 1 授予上述資料欄以外的所有存取權。

```
aws lakeformation create-data-cells-filter \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "TableData": {
        "ColumnWildcard": {
            "ExcludedColumnNames": ["p_channel_details", "p_start_date_sk"]
        },
        "DatabaseName": "tpc",
        "Name": "developer_promotion_excludecolumn",
        "RowFilter": {
            "AllRowsWildcard": {}
        },
        "TableCatalogId": "123456789111",
        "TableName": "dl_tpc_promotion"
    }
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Lake Formation 開發人員指南中的 Lake Formation 中的資料篩選和儲存格層級安全性。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDataCellsFilter。

create-lf-tag

以下程式碼範例顯示如何使用 create-lf-tag。

AWS CLI

建立 LF 標籤

下列create-lf-tag範例會建立具有指定名稱和值的 LF 標籤。

```
aws lakeformation create-lf-tag \
    --catalog-id '123456789111' \
    --tag-key 'usergroup' \
    --tag-values '["developer","analyst","campaign"]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的管理中繼資料存取控制的 LF 標籤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLfTag。

delete-data-cells-filter

以下程式碼範例顯示如何使用 delete-data-cells-filter。

AWS CLI

刪除資料儲存格篩選條件

下列delete-data-cells-filter範例會刪除指定的資料儲存格篩選條件。

```
aws lakeformation delete-data-cells-filter \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
```

```
    "TableCatalogId": "123456789111",
    "DatabaseName": "tpc",
    "TableName": "dl_tpc_promotion",
    "Name": "developer_promotion"
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Lake Formation 開發人員指南中的 Lake Formation 中的資料篩選和儲存格層級安全性。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDataCellsFilter。

delete-lf-tag

以下程式碼範例顯示如何使用 delete-lf-tag。

AWS CLI

刪除 LF 標籤定義

下列delete-lf-tag範例會刪除 LF-Tag 定義。

```
aws lakeformation delete-lf-tag \
    --catalog-id '123456789111' \
    --tag-key 'usergroup'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的管理中繼資料存取控制的 LF 標籤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLfTag。

delete-objects-on-cancel

以下程式碼範例顯示如何使用 delete-objects-on-cancel。

AWS CLI

在交易取消時刪除物件

下列delete-objects-on-cancel範例會在交易取消時刪除列出的 s3 物件。

```
aws lakeformation delete-objects-on-cancel \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "012345678901",
    "DatabaseName": "tpc",
    "TableName": "dl_tpc_household_demographics_gov",
    "TransactionId": "1234d972ca8347b89825e33c5774aec4",
    "Objects": [{
        "Uri": "s3://lf-data-lake-012345678901/target/
dl_tpc_household_demographics_gov/run-unnamed-1-part-block-0-r-00000-snappy-
ff26b17504414fe88b302cd795eabd00.parquet",
        "ETag": "1234ab1fc50a316b149b4e1f21a73800"
    }]
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的讀取和寫入交易中的資料湖。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteObjectsOnCancel。

deregister-resource

以下程式碼範例顯示如何使用 deregister-resource。

AWS CLI

取消註冊資料湖儲存

下列deregister-resource範例會取消註冊由 Lake Formation 管理的資源。

```
aws lakeformation deregister-resource \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123"
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的將 Amazon S3 位置新增至您的資料湖。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterResource。

describe-transaction

以下程式碼範例顯示如何使用 describe-transaction。

AWS CLI

擷取交易詳細資訊

下列describe-transaction範例會傳回單一交易的詳細資訊。

```
aws lakeformation describe-transaction \
    --transaction-id='8cb4b1a7cc8d486fbaca9a64e7d9f5ce'
```

輸出：

```
{
    "TransactionDescription": {
        "TransactionId": "12345972ca8347b89825e33c5774aec4",
        "TransactionStatus": "committed",
        "TransactionStartTime": "2022-08-10T14:29:04.046000+00:00",
        "TransactionEndTime": "2022-08-10T14:29:09.681000+00:00"
    }
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的讀取和寫入交易中的資料湖。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTransaction。

extend-transaction

以下程式碼範例顯示如何使用 extend-transaction。

AWS CLI

擴展交易

下列extend-transaction範例會擴展交易。

```
aws lakeformation extend-transaction \
    --transaction-id='8cb4b1a7cc8d486fbaca9a64e7d9f5ce'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的讀取和寫入交易中的資料湖。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ExtendTransaction。

get-data-lake-settings

以下程式碼範例顯示如何使用 get-data-lake-settings。

AWS CLI

擷取 AWS Lake Formation 管理的資料湖設定

下列get-data-lake-settings範例會擷取資料湖管理員和其他資料湖設定的清單。

```
aws lakeformation get-data-lake-settings \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "123456789111"
}
```

輸出：

```
{
    "DataLakeSettings": {
        "DataLakeAdmins": [{
            "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-admin"
        }],
        "CreateDatabaseDefaultPermissions": [],
        "CreateTableDefaultPermissions": [
                {
                        "Principal": {
```

```
                            "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
                        },
                        "Permissions": [
                            "ALL"
                        ]
                }
        ],
        "TrustedResourceOwners": [],
        "AllowExternalDataFiltering": true,
        "ExternalDataFilteringAllowList": [{
            "DataLakePrincipalIdentifier": "123456789111"
        }],
        "AuthorizedSessionTagValueList": [
            "Amazon EMR"
        ]
    }
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的變更資料湖的預設安全設定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDataLakeSettings。

get-effective-permissions-for-path

以下程式碼範例顯示如何使用 get-effective-permissions-for-path。

AWS CLI

擷取位於特定路徑之資源的許可

下列get-effective-permissions-for-path範例會傳回位於 Amazon S3 路徑之指定資料表或資料庫資源的 Lake Formation 許可。

```
aws lakeformation get-effective-permissions-for-path \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "123456789111",
    "ResourceArn": "arn:aws:s3:::lf-data-lake-123456789111"
}
```

輸出：

```
{
    "Permissions": [{
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
campaign-manager"
            },
            "Resource": {
                "Database": {
                    "Name": "tpc"
                }
            },
            "Permissions": [
                "DESCRIBE"
            ],
            "PermissionsWithGrantOption": []
        },
        {
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/EMR-
RuntimeRole"
            },
            "Resource": {
                "Database": {
                    "Name": "tpc"
                }
            },
            "Permissions": [
                "ALL"
            ],
            "PermissionsWithGrantOption": []
        },
        {
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:saml-
provider/oktaSAMLProvider:user/emr-developer"
            },
            "Resource": {
                "Database": {
                    "Name": "tpc"
                }
            },
            "Permissions": [
```

```
                "ALL",
                "DESCRIBE"
            ],
            "PermissionsWithGrantOption": []
        },
        {
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
admin"
            },
            "Resource": {
                "Database": {
                    "Name": "tpc"
                }
            },
            "Permissions": [
                "ALL",
                "ALTER",
                "CREATE_TABLE",
                "DESCRIBE",
                "DROP"
            ],
            "PermissionsWithGrantOption": [
                "ALL",
                "ALTER",
                "CREATE_TABLE",
                "DESCRIBE",
                "DROP"
            ]
        },
        {
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/LF-
GlueServiceRole"
            },
            "Resource": {
                "Database": {
                    "Name": "tpc"
                }
            },
            "Permissions": [
                "CREATE_TABLE"
            ],
            "PermissionsWithGrantOption": []
```

```
        }
    ],
    "NextToken":
 "E5SlJDSTZleUp6SWpvaU9UQTNORE0zTXpFeE5Ua3pJbjE5TENKbGVIQnBjbUYwYVc5dUlqcDdJbk5sWTI5dVpITWlP
}
```

如需詳細資訊，請參閱 Lake Formation 開發人員指南中的管理 Lake Formation 許可。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetEffectivePermissionsForPath。

get-lf-tag

以下程式碼範例顯示如何使用 get-lf-tag。

AWS CLI

擷取 LF 標籤定義

下列get-lf-tag範例會擷取 LF 標籤定義。

```
aws lakeformation get-lf-tag \
    --catalog-id '123456789111' \
    --tag-key 'usergroup'
```

輸出：

```
{
    "CatalogId": "123456789111",
    "TagKey": "usergroup",
    "TagValues": [
        "analyst",
        "campaign",
        "developer"
    ]
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的管理中繼資料存取控制的 LF 標籤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLfTag。

get-query-state

以下程式碼範例顯示如何使用 get-query-state。

AWS CLI

擷取已提交查詢的狀態

下列get-query-state範例會傳回先前提交的查詢狀態。

```
aws lakeformation get-query-state \
    --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

輸出：

```
{
    "State": "FINISHED"
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的交易資料操作。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetQueryState。

get-query-statistics

以下程式碼範例顯示如何使用 get-query-statistics。

AWS CLI

擷取查詢統計資料

下列get-query-statistics範例會擷取查詢規劃和執行的統計資料。

```
aws lakeformation get-query-statistics \
    --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

輸出：

```
{
    "ExecutionStatistics": {
        "AverageExecutionTimeMillis": 0,
        "DataScannedBytes": 0,
```

```
        "WorkUnitsExecutedCount": 0
    },
    "PlanningStatistics": {
        "EstimatedDataToScanBytes": 43235,
        "PlanningTimeMillis": 2377,
        "QueueTimeMillis": 440,
        "WorkUnitsGeneratedCount": 1
    },
    "QuerySubmissionTime": "2022-08-11T02:14:38.641870+00:00"
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的交易資料操作。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetQueryStatistics。

get-resource-lf-tags

以下程式碼範例顯示如何使用 get-resource-lf-tags。

AWS CLI

列出 LF 標籤

下列list-lf-tags範例會傳回請求者有權檢視的 LF 標籤清單。

```
aws lakeformation list-lf-tags \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "123456789111",
    "ResourceShareType": "ALL",
    "MaxResults": 2
}
```

輸出：

```
{
"LFTags": [{
        "CatalogId": "123456789111",
        "TagKey": "category",
```

```
        "TagValues": [
            "private",
            "public"
        ]
    },
    {
        "CatalogId": "123456789111",
        "TagKey": "group",
        "TagValues": [
            "analyst",
            "campaign",
            "developer"
        ]
    }],
    "NextToken": "kIiwiZXhwaXJhdGlvbiI6eyJzZWNvbmRzIjoxNjYwMDY4dCI6ZmFsc2V9"
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的管理中繼資料存取控制的 LF 標籤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResourceLfTags。

get-table-objects

以下程式碼範例顯示如何使用 get-table-objects。

AWS CLI

列出受管資料表的物件

下列get-table-objects範例會傳回組成指定受管資料表的一組 Amazon S3 物件。

```
aws lakeformation get-table-objects \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "012345678901",
    "DatabaseName": "tpc",
    "TableName": "dl_tpc_household_demographics_gov",
    "QueryAsOfTime": "2022-08-10T15:00:00"
```

```
}
```

輸出：

```
{
    "Objects": [{
        "PartitionValues": [],
        "Objects": [{
            "Uri": "s3://lf-data-lake-012345678901/target/
dl_tpc_household_demographics_gov/run-unnamed-1-part-block-0-r-00000-snappy-
ff26b17504414fe88b302cd795eabd00.parquet",
            "ETag": "12345b1fc50a316b149b4e1f21a73800",
            "Size": 43235
        }]
    }]
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的讀取和寫入交易中的資料湖。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTableObjects。

get-work-unit-results

以下程式碼範例顯示如何使用 get-work-unit-results。

AWS CLI

擷取指定查詢的工作單位

下列get-work-unit-results範例會傳回查詢所產生的工作單位。

```
aws lakeformation get-work-units \
    --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b' \
    --work-unit-id '0' \
    --work-unit-token 'B2fMSdmQXe9umX8Ux8XCo4=' outfile
```

輸出：

```
outfile with Blob content.
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的交易資料操作。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetWorkUnitResults。

get-work-units

以下程式碼範例顯示如何使用 get-work-units。

AWS CLI

擷取工作單位

下列get-work-units範例會擷取 StartQueryPlanning 操作所產生的工作單位。

```
aws lakeformation get-work-units \
    --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

輸出：

```
{
    "WorkUnitRanges": [{
        "WorkUnitIdMax": 0,
        "WorkUnitIdMin": 0,
        "WorkUnitToken":
 "1234eMAk4kLO4umqEL4Z5WuxL04AXwABABVhd3MtY3J5cHRvLXB1YmxpYy1rZXkAREEwYm9QbkhINmFYTWphbmMxZW
+f88jzGrYq22gE6jkQlpOB
+Oet2eqNUmFudAAAAfjB8BgkqhkiG9w0BBwagbzBtAgEAMGgGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMCOEWRda
wAAAAEAAAAAAAAAAAAAAAEAAACX3/w5h75QAPomfKH+cyEKYU1yccUmBl
+VSojiGOtdsUk7vcjYXUUboYm3dvqRqX2s4gROMOn
+Ij8R0/8jYmnHkpvyAFNVRPyETyIKg7k5Z9+5I1c2d3446Jw/moWGGxjH8AEG9h27ytmOhozxDOEi/
F2ZoXz6wlGDfGUo/2WxCkYOhTyNaw6TM
+7drTM7yrW4iNVLUM0LX0xnFjIAhLhooWJek6vjQZUAZzBlAjBH8okRtYP8R7AY2Wls/
hqFBhG0V4l42AC0LxsuZbMQrE2SzWZUZ0E9Uew7/n0cyX4CMQDR79INyv4ysMByW9kKGGKyba+cCNklExMR
+btBQBmMuB2fMSdmQXe9umX8Ux8XCo4="
    }],
    "QueryId": "1234273f-4a62-4cda-8d98-69615ee8be9b"
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的交易資料操作。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetWorkUnits。

grant-permissions

以下程式碼範例顯示如何使用 grant-permissions。

AWS CLI

範例 1：使用 LF-Tags 將許可授予資源上的委託人

下列grant-permissions範例會將所有許可授予資料庫資源上符合 LF-Tag 政策的主體。

```
aws lakeformation grant-permissions \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "123456789111",
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-admin"
    },
    "Resource": {
        "LFTagPolicy": {
            "CatalogId": "123456789111",
            "ResourceType": "DATABASE",
            "Expression": [{
                "TagKey": "usergroup",
                "TagValues": [
                    "analyst",
                    "developer"
                ]
            }]
        }
    },
    "Permissions": [
        "ALL"
    ],
    "PermissionsWithGrantOption": [
        "ALL"
    ]
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的授予和撤銷 Data Catalog 資源的許可。

範例 2：將資料欄層級許可授予委託人

下列grant-permissions範例會授予許可，以選取委託人的特定資料欄。

```
aws lakeformation grant-permissions \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "123456789111",
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
    },
    "Resource": {
        "TableWithColumns": {
            "CatalogId": "123456789111",
            "ColumnNames": ["p_end_date_sk"],
            "DatabaseName": "tpc",
            "Name": "dl_tpc_promotion"
        }
    },
    "Permissions": [
        "SELECT"
    ],
    "PermissionsWithGrantOption": []
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的授予和撤銷 Data Catalog 資源的許可。

範例 3：將資料表許可授予委託人

下列grant-permissions範例會將指定資料庫之所有資料表的選取許可授予委託人。

```
aws lakeformation grant-permissions \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "123456789111",
```

```
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
    },
    "Resource": {
        "Table": {
            "CatalogId": "123456789111",
            "DatabaseName": "tpc",
            "TableWildcard": {}
        }
    },
    "Permissions": [
        "SELECT"
    ],
    "PermissionsWithGrantOption": []
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的授予和撤銷 Data Catalog 資源的許可。

範例 4：將 LF 標籤的許可授予委託人

下列grant-permissions範例會將 LF 標籤的關聯許可授予委託人。

```
aws lakeformation grant-permissions \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "123456789111",
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
    },
    "Resource": {
        "LFTag": {
            "CatalogId": "123456789111",
            "TagKey": "category",
            "TagValues": [
                "private", "public"
            ]
        }
```

```
    },
    "Permissions": [
        "ASSOCIATE"
    ],
    "PermissionsWithGrantOption": []
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的授予和撤銷 Data Catalog 資源的許可。

範例 5：將資料位置的許可授予委託人

下列grant-permissions範例會將資料位置的許可授予委託人。

```
aws lakeformation grant-permissions \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "123456789111",
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
    },
    "Resource": {
        "DataLocation": {
            "CatalogId": "123456789111",
            "ResourceArn": "arn:aws:s3:::lf-data-lake-123456789111"
        }
    },
    "Permissions": [
        "DATA_LOCATION_ACCESS"
    ],
    "PermissionsWithGrantOption": []
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的授予和撤銷 Data Catalog 資源的許可。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GrantPermissions。

list-data-cells-filter

以下程式碼範例顯示如何使用 list-data-cells-filter。

AWS CLI

列出資料儲存格篩選條件

下列list-data-cells-filter範例列出指定資料表的資料儲存格篩選條件。

```
aws lakeformation list-data-cells-filter \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "MaxResults": 2,
    "Table": {
        "CatalogId": "123456789111",
        "DatabaseName": "tpc",
        "Name": "dl_tpc_promotion"
    }
}
```

輸出：

```
{
    "DataCellsFilters": [{
            "TableCatalogId": "123456789111",
            "DatabaseName": "tpc",
            "TableName": "dl_tpc_promotion",
            "Name": "developer_promotion",
            "RowFilter": {
                "FilterExpression": "p_promo_name='ese'"
            },
            "ColumnNames": [
                "p_channel_details",
                "p_start_date_sk",
                "p_purpose",
```

```
                "p_promo_id",
                "p_promo_name",
                "p_end_date_sk",
                "p_discount_active"
            ]
        },
        {
            "TableCatalogId": "123456789111",
            "DatabaseName": "tpc",
            "TableName": "dl_tpc_promotion",
            "Name": "developer_promotion_allrows",
            "RowFilter": {
                "FilterExpression": "TRUE",
                "AllRowsWildcard": {}
            },
            "ColumnNames": [
                "p_channel_details",
                "p_start_date_sk",
                "p_promo_name"
            ]
        }
    ],
    "NextToken": "2MDA2MTgwNiwibmFub3MiOjE0MDAwMDAwMH19"
}
```

如需詳細資訊，請參閱 Lake Formation 開發人員指南中的 Lake Formation 中的資料篩選和儲存格層級安全性。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDataCellsFilter。

list-permissions

以下程式碼範例顯示如何使用 `list-permissions`。

AWS CLI

範例 1：擷取資源的主體許可清單

下列`list-permissions`範例會傳回資料庫資源的主體許可清單。

```
aws lakeformation list-permissions \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "123456789111",
    "ResourceType": "DATABASE",
    "MaxResults": 2
}
```

輸出：

```
{
    "PrincipalResourcePermissions": [{
        "Principal": {
            "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
campaign-manager"
        },
        "Resource": {
            "Database": {
                "CatalogId": "123456789111",
                "Name": "tpc"
            }
        },
        "Permissions": [
            "DESCRIBE"
        ],
        "PermissionsWithGrantOption": []
    }],
    "NextToken":
 "E5SlJDSTZleUp6SWpvaU9UQTNORE0zTXpFeE5Ua3pJbjE5TENKbGVIQnBjbUYwYVc5dUlqcDdJbk5sWTI5dVpITWlP
}
```

如需詳細資訊，請參閱 Lake Formation 開發人員指南中的管理 Lake Formation 許可。 AWS

範例 2：使用資料篩選條件擷取資料表上的主體許可清單

下列list-permissions範例會列出資料表上的許可，並將相關資料篩選條件授予委託人。

```
aws lakeformation list-permissions \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "123456789111",
    "Resource": {
        "Table": {
            "CatalogId": "123456789111",
            "DatabaseName": "tpc",
            "Name": "dl_tpc_customer"
        }
    },
    "IncludeRelated": "TRUE",
    "MaxResults": 10
}
```

輸出：

```
{
    "PrincipalResourcePermissions": [{
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/
Admin"
            },
            "Resource": {
                "Table": {
                    "CatalogId": "123456789111",
                    "DatabaseName": "customer",
                    "Name": "customer_invoice"
                }
            },
            "Permissions": [
                "ALL",
                "ALTER",
                "DELETE",
                "DESCRIBE",
                "DROP",
                "INSERT"
            ],
            "PermissionsWithGrantOption": [
                "ALL",
                "ALTER",
                "DELETE",
                "DESCRIBE",
                "DROP",
                "INSERT"
```

```
            ]
        },
        {
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/
Admin"
            },
            "Resource": {
                "TableWithColumns": {
                    "CatalogId": "123456789111",
                    "DatabaseName": "customer",
                    "Name": "customer_invoice",
                    "ColumnWildcard": {}
                }
            },
            "Permissions": [
                "SELECT"
            ],
            "PermissionsWithGrantOption": [
                "SELECT"
            ]
        },
        {
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/
Admin"
            },
            "Resource": {
                "DataCellsFilter": {
                    "TableCatalogId": "123456789111",
                    "DatabaseName": "customer",
                    "TableName": "customer_invoice",
                    "Name": "dl_us_customer"
                }
            },
            "Permissions": [
                "DESCRIBE",
                "SELECT",
                "DROP"
            ],
            "PermissionsWithGrantOption": []
        }
    ],
    "NextToken": "VyeUFjY291bnRQZXJtaXNzaW9ucyI6ZmFsc2V9"
```

```
}
```

如需詳細資訊，請參閱 Lake Formation 開發人員指南中的管理 Lake Formation 許可。 AWS

範例 3：擷取 LF 標籤上主體許可的清單

下列list-permissions範例列出授予委託人之 LF 標籤的許可。

```
aws lakeformation list-permissions \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "123456789111",
    "Resource": {
        "LFTag": {
            "CatalogId": "123456789111",
            "TagKey": "category",
            "TagValues": [
                "private"
            ]
        }
    },
    "MaxResults": 10
}
```

輸出：

```
{
    "PrincipalResourcePermissions": [{
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
admin"
            },
            "Resource": {
                "LFTag": {
                    "CatalogId": "123456789111",
                    "TagKey": "category",
                    "TagValues": [
                        "*"
                    ]
                }
```

```
            },
            "Permissions": [
                "DESCRIBE"
            ],
            "PermissionsWithGrantOption": [
                "DESCRIBE"
            ]
        },
        {
            "Principal": {
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
admin"
            },
            "Resource": {
                "LFTag": {
                    "CatalogId": "123456789111",
                    "TagKey": "category",
                    "TagValues": [
                        "*"
                    ]
                }
            },
            "Permissions": [
                "ASSOCIATE"
            ],
            "PermissionsWithGrantOption": [
                "ASSOCIATE"
            ]
        }
    ],
    "NextToken": "EJwY21GMGFXOXVJanA3SW5Ocm1pc3Npb25zIjpmYWxzZX0="
}
```

如需詳細資訊，請參閱 Lake Formation 開發人員指南中的管理 Lake Formation 許可。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPermissions。

`list-resources`

以下程式碼範例顯示如何使用 `list-resources`。

AWS CLI

列出 Lake Formation 管理的資源

下列list-resources範例列出符合 Lake Formation 所管理條件的資源。

```
aws lakeformation list-resources \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "FilterConditionList": [{
        "Field": "ROLE_ARN",
        "ComparisonOperator": "CONTAINS",
        "StringValueList": [
            "123456789111"
        ]
    }],
    "MaxResults": 10
}
```

輸出：

```
{
    "ResourceInfoList": [{
            "ResourceArn": "arn:aws:s3:::lf-data-lake-123456789111",
            "RoleArn": "arn:aws:iam::123456789111:role/LF-GlueServiceRole",
            "LastModified": "2022-07-21T02:12:46.669000+00:00"
        },
        {
            "ResourceArn": "arn:aws:s3:::lf-emr-test-123456789111",
            "RoleArn": "arn:aws:iam::123456789111:role/EMRLFS3Role",
            "LastModified": "2022-07-29T16:22:03.211000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱 Lake Formation 開發人員指南中的管理 Lake Formation 許可。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResources。

list-transactions

以下程式碼範例顯示如何使用 list-transactions。

AWS CLI

列出所有交易詳細資訊

下列list-transactions範例會傳回有關交易及其狀態的中繼資料。

```
aws lakeformation list-transactions \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "123456789111",
    "StatusFilter": "ALL",
    "MaxResults": 3
}
```

輸出：

```
{
    "Transactions": [{
            "TransactionId": "1234569f08804cb790d950d4d0fe485e",
            "TransactionStatus": "committed",
            "TransactionStartTime": "2022-08-10T14:32:29.220000+00:00",
            "TransactionEndTime": "2022-08-10T14:32:33.751000+00:00"
        },
        {
            "TransactionId": "12345972ca8347b89825e33c5774aec4",
            "TransactionStatus": "committed",
            "TransactionStartTime": "2022-08-10T14:29:04.046000+00:00",
            "TransactionEndTime": "2022-08-10T14:29:09.681000+00:00"
        },
        {
            "TransactionId": "12345daf6cb047dbba8ad9b0414613b2",
            "TransactionStatus": "committed",
            "TransactionStartTime": "2022-08-10T13:56:51.261000+00:00",
            "TransactionEndTime": "2022-08-10T13:56:51.547000+00:00"
        }
    ],
    "NextToken": "77X1ebypsI7os+X2lhHsZLGNCDK3nNGpwRdFpicSOHgcX1/
QMoniUAKcpR3kj3ts3PVdMA=="
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的讀取和寫入交易中的資料湖。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTransactions。

put-data-lake-settings

以下程式碼範例顯示如何使用 put-data-lake-settings。

AWS CLI

設定 AWS Lake Formation 受管資料湖設定

下列put-data-lake-settings範例會設定資料湖管理員和其他資料湖設定的清單。

```
aws lakeformation put-data-lake-settings \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "DataLakeSettings": {
        "DataLakeAdmins": [{
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
admin"
            }
        ],
        "CreateDatabaseDefaultPermissions": [],
        "CreateTableDefaultPermissions": [],
        "TrustedResourceOwners": [],
        "AllowExternalDataFiltering": true,
        "ExternalDataFilteringAllowList": [{
            "DataLakePrincipalIdentifier ": "123456789111"
        }],
        "AuthorizedSessionTagValueList": ["Amazon EMR"]
    }
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的變更資料湖的預設安全設定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutDataLakeSettings。

register-resource

以下程式碼範例顯示如何使用 register-resource。

AWS CLI

範例 1：使用服務連結角色註冊資料湖儲存

下列register-resource範例會使用服務連結角色，將資源註冊為由 Lake Formation 管理。

```
aws lakeformation register-resource \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123",
    "UseServiceLinkedRole": true
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的將 Amazon S3 位置新增至您的資料湖。

範例 2：使用自訂角色註冊資料湖儲存

下列register-resource範例會使用自訂角色，將資源註冊為由 Lake Formation 管理。

```
aws lakeformation register-resource \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123",
    "UseServiceLinkedRole": false,
    "RoleArn": "arn:aws:iam::123456789111:role/LF-GlueServiceRole"
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的將 Amazon S3 位置新增至您的資料湖。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterResource。

`remove-lf-tags-from-resource`

以下程式碼範例顯示如何使用 `remove-lf-tags-from-resource`。

AWS CLI

從資源中移除 LF-Tag

下列`remove-lf-tags-from-resource`範例會移除與資料表資源的 LF-Tag 關聯。

```
aws lakeformation remove-lf-tags-from-resource \
    --cli-input-json file://input.json
```

`input.json` 的內容：

```
{
    "CatalogId": "123456789111",
    "Resource": {
        "Table": {
            "CatalogId": "123456789111",
            "DatabaseName": "tpc",
            "Name": "dl_tpc_promotion"
        }
    },
    "LFTags": [{
        "CatalogId": "123456789111",
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }]
}
```

輸出：

```
{
    "Failures": []
```

```
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的將 LF 標籤指派給 Data Catalog 資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveLfTagsFromResource。

`revoke-permissions`

以下程式碼範例顯示如何使用 `revoke-permissions`。

AWS CLI

從委託人撤銷資源的許可

下列`revoke-permissions`範例會撤銷特定資料庫特定資料表的委託人存取權。

```
aws lakeformation revoke-permissions \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "123456789111",
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
    },
    "Resource": {
        "Table": {
            "CatalogId": "123456789111",
            "DatabaseName": "tpc",
            "Name": "dl_tpc_promotion"
        }
    },
    "Permissions": [
        "ALL"
    ],
    "PermissionsWithGrantOption": []
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的授予和撤銷 Data Catalog 資源的許可。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RevokePermissions。

search-databases-by-lf-tags

以下程式碼範例顯示如何使用 search-databases-by-lf-tags。

AWS CLI

依 LFTags 搜尋資料庫資源

下列search-databases-by-lf-tags範例搜尋符合 LFTag 表達式的資料庫資源。

```
aws lakeformation search-databases-by-lf-tags \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "MaxResults": 1,
    "CatalogId": "123456789111",
    "Expression": [{
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }]
}
```

輸出：

```
{
    "DatabaseList": [{
        "Database": {
            "CatalogId": "123456789111",
            "Name": "tpc"
        },
        "LFTags": [{
            "CatalogId": "123456789111",
            "TagKey": "usergroup",
            "TagValues": [
```

```
                "developer"
            ]
        }]
    }]
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的檢視 LF 標籤指派給的資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchDatabasesByLfTags。

search-tables-by-lf-tags

以下程式碼範例顯示如何使用 search-tables-by-lf-tags。

AWS CLI

依 LFTags 搜尋資料表資源

下列search-tables-by-lf-tags範例搜尋符合 LFTag 表達式的資料表資源。

```
aws lakeformation search-tables-by-lf-tags \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "MaxResults": 2,
    "CatalogId": "123456789111",
    "Expression": [{
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }]
}
```

輸出：

```
{
    "NextToken": "c2VhcmNoQWxsVGFnc0luVGFibGVzIjpmYWxzZX0=",
    "TableList": [{
        "Table": {
            "CatalogId": "123456789111",
```

```
            "DatabaseName": "tpc",
            "Name": "dl_tpc_item"
        },
        "LFTagOnDatabase": [{
            "CatalogId": "123456789111",
            "TagKey": "usergroup",
            "TagValues": [
                "developer"
            ]
        }],
        "LFTagsOnTable": [{
            "CatalogId": "123456789111",
            "TagKey": "usergroup",
            "TagValues": [
                "developer"
            ]
        }],
        "LFTagsOnColumns": [{
                "Name": "i_item_desc",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_container",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_wholesale_cost",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
```

```
                ]
            }]
        },
        {
            "Name": "i_manufact_id",
            "LFTags": [{
                "CatalogId": "123456789111",
                "TagKey": "usergroup",
                "TagValues": [
                    "developer"
                ]
            }]
        },
        {
            "Name": "i_brand_id",
            "LFTags": [{
                "CatalogId": "123456789111",
                "TagKey": "usergroup",
                "TagValues": [
                    "developer"
                ]
            }]
        },
        {
            "Name": "i_formulation",
            "LFTags": [{
                "CatalogId": "123456789111",
                "TagKey": "usergroup",
                "TagValues": [
                    "developer"
                ]
            }]
        },
        {
            "Name": "i_current_price",
            "LFTags": [{
                "CatalogId": "123456789111",
                "TagKey": "usergroup",
                "TagValues": [
                    "developer"
                ]
            }]
        },
        {
```

```
                "Name": "i_size",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_rec_start_date",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_manufact",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_item_sk",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_manager_id",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
```

```
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_item_id",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_class_id",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_class",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_category",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
```

```
            },
            {
                "Name": "i_category_id",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_brand",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_units",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_rec_end_date",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_color",
                "LFTags": [{
```

```
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            },
            {
                "Name": "i_product_name",
                "LFTags": [{
                    "CatalogId": "123456789111",
                    "TagKey": "usergroup",
                    "TagValues": [
                        "developer"
                    ]
                }]
            }
        ]
    }]
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的檢視 LF 標籤指派給的資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchTablesByLfTags。

start-query-planning

以下程式碼範例顯示如何使用 start-query-planning。

AWS CLI

處理查詢陳述式

下列start-query-planning範例會提交處理查詢陳述式的請求。

```
aws lakeformation start-query-planning \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "QueryPlanningContext": {
        "CatalogId": "012345678901",
```

```
        "DatabaseName": "tpc"
    },
    "QueryString": "select * from dl_tpc_household_demographics_gov where
 hd_income_band_sk=9"
}
```

輸出：

```
{
    "QueryId": "772a273f-4a62-4cda-8d98-69615ee8be9b"
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的讀取和寫入交易中的資料湖。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartQueryPlanning。

`start-transaction`

以下程式碼範例顯示如何使用 `start-transaction`。

AWS CLI

開始新的交易

下列`start-transaction`範例會啟動新的交易並傳回其交易 ID。

```
aws lakeformation start-transaction \
    --transaction-type = 'READ_AND_WRITE'
```

輸出：

```
{
    "TransactionId": "b014d972ca8347b89825e33c5774aec4"
}
```

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的讀取和寫入交易中的資料湖。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartTransaction。

`update-lf-tag`

以下程式碼範例顯示如何使用 `update-lf-tag`。

AWS CLI

更新 LF 標籤定義

下列update-lf-tag範例會更新 LF-Tag 定義。

```
aws lakeformation update-lf-tag \
    --catalog-id '123456789111' \
    --tag-key 'usergroup' \
    --tag-values-to-add '["admin"]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的管理中繼資料存取控制的 LF 標籤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateLfTag。

update-table-objects

以下程式碼範例顯示如何使用 update-table-objects。

AWS CLI

修改受管資料表的物件

下列update-table-objects範例會將提供的 S3 物件新增至指定的受管資料表。

```
aws lakeformation update-table-objects \
    --cli-input-json file://input.json
```

input.json 的內容：

```
{
    "CatalogId": "012345678901",
    "DatabaseName": "tpc",
    "TableName": "dl_tpc_household_demographics_gov",
    "TransactionId": "12347a9f75424b9b915f6ff201d2a190",
    "WriteOperations": [{
        "AddObject": {
            "Uri": "s3://lf-data-lake-012345678901/target/
dl_tpc_household_demographics_gov/run-unnamed-1-part-block-0-r-00000-snappy-
ff26b17504414fe88b302cd795eabd00.parquet",
```

```
            "ETag": "1234ab1fc50a316b149b4e1f21a73800",
            "Size": 42200
        }
    }]
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的讀取和寫入交易中的資料湖。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateTableObjects。

使用 的 Lambda 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Lambda 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`add-layer-version-permission`

以下程式碼範例顯示如何使用 `add-layer-version-permission`。

AWS CLI

將許可新增至 layer 版本

下列`add-layer-version-permission`範例授予指定帳戶使用第 1 版 layer 的許可`my-layer`。

```
aws lambda add-layer-version-permission \
    --layer-name my-layer \
    --statement-id xaccount \
```

```
    --action lambda:GetLayerVersion  \
    --principal 123456789012 \
    --version-number 1
```

輸出：

```
{
    "RevisionId": "35d87451-f796-4a3f-a618-95a3671b0a0c",
    "Statement":
    {
        "Sid":"xaccount",
        "Effect":"Allow",
        "Principal":{
            "AWS":"arn:aws:iam::210987654321:root"
        },
        "Action":"lambda:GetLayerVersion",
        "Resource":"arn:aws:lambda:us-east-2:123456789012:layer:my-layer:1"
    }
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 Lambda Layers。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddLayerVersionPermission。

add-permission

以下程式碼範例顯示如何使用 add-permission。

AWS CLI

將許可新增至現有的 Lambda 函數

下列add-permission範例會授予 Amazon SNS 服務許可，以叫用名為 的函數my-function。

```
aws lambda add-permission \
    --function-name my-function \
    --action lambda:InvokeFunction \
    --statement-id sns \
    --principal sns.amazonaws.com
```

輸出：

```
{
    "Statement":
    {
        "Sid":"sns",
        "Effect":"Allow",
        "Principal":{
            "Service":"sns.amazonaws.com"
        },
        "Action":"lambda:InvokeFunction",
        "Resource":"arn:aws:lambda:us-east-2:123456789012:function:my-function"
    }
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的使用 Lambda 的資源型政策。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddPermission。

create-alias

以下程式碼範例顯示如何使用 create-alias。

AWS CLI

建立 Lambda 函數的別名

以下 create-alias 範例會建立一個名為 LIVE 的別名，此別名指向 my-function Lambda 函數的版本 1。

```
aws lambda create-alias \
    --function-name my-function \
    --description "alias for live version of function" \
    --function-version 1 \
    --name LIVE
```

輸出：

```
{
    "FunctionVersion": "1",
    "Name": "LIVE",
    "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:LIVE",
    "RevisionId": "873282ed-4cd3-4dc8-a069-d0c647e470c6",
    "Description": "alias for live version of function"
```

```
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的設定 Lambda 函數別名。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateAlias。

create-event-source-mapping

以下程式碼範例顯示如何使用 create-event-source-mapping。

AWS CLI

在事件來源和 AWS Lambda 函數之間建立映射

下列create-event-source-mapping範例會在 SQS 佇列和 my-function Lambda 函數之間建立映射。

```
aws lambda create-event-source-mapping \
    --function-name my-function \
    --batch-size 5 \
    --event-source-arn arn:aws:sqs:us-west-2:123456789012:mySQSqueue
```

輸出：

```
{
    "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "StateTransitionReason": "USER_INITIATED",
    "LastModified": 1569284520.333,
    "BatchSize": 5,
    "State": "Creating",
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 Lambda 事件來源映射。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateEventSourceMapping。

create-function

以下程式碼範例顯示如何使用 create-function。

AWS CLI

若要建立 Lambda 函數

下列 create-function 範例會建立名為 my-function 的 Lambda 函數。

```
aws lambda create-function \
    --function-name my-function \
    --runtime nodejs18.x \
    --zip-file fileb://my-function.zip \
    --handler my-function.handler \
    --role arn:aws:iam::123456789012:role/service-role/MyTestFunction-role-tges6bf4
```

my-function.zip 的內容：

```
This file is a deployment package that contains your function code and any
 dependencies.
```

輸出：

```
{
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "CodeSha256": "PFn4S+er27qk+UuZSTKEQfNKG/XNn7QJs90mJgq6oH8=",
    "FunctionName": "my-function",
    "CodeSize": 308,
    "RevisionId": "873282ed-4cd3-4dc8-a069-d0c647e470c6",
    "MemorySize": 128,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "Version": "$LATEST",
    "Role": "arn:aws:iam::123456789012:role/service-role/MyTestFunction-role-
zgur6bf4",
    "Timeout": 3,
    "LastModified": "2023-10-14T22:26:11.234+0000",
    "Handler": "my-function.handler",
    "Runtime": "nodejs18.x",
    "Description": ""
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 AWS Lambda 函數組態。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateFunction。

delete-alias

以下程式碼範例顯示如何使用 delete-alias。

AWS CLI

若要刪除 Lambda 函數的別名

下列 delete-alias 範例會從 my-function Lambda 函數中刪除名為 LIVE 的別名。

```
aws lambda delete-alias \
    --function-name my-function \
    --name LIVE
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的設定 Lambda 函數別名。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteAlias。

delete-event-source-mapping

以下程式碼範例顯示如何使用 delete-event-source-mapping。

AWS CLI

刪除事件來源與 AWS Lambda 函數之間的映射

下列delete-event-source-mapping範例會刪除 SQS 佇列和 my-function Lambda 函數之間的映射。

```
aws lambda delete-event-source-mapping \
    --uuid  a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

輸出：

```
{
    "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "StateTransitionReason": "USER_INITIATED",
    "LastModified": 1569285870.271,
    "BatchSize": 5,
    "State": "Deleting",
```

```
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 Lambda 事件來源映射。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteEventSourceMapping。

`delete-function-concurrency`

以下程式碼範例顯示如何使用 `delete-function-concurrency`。

AWS CLI

若要從函數中移除預留並行執行限制

下列 `delete-function-concurrency` 範例會從 `my-function` 函數中刪除預留並行執行限制。

```
aws lambda delete-function-concurrency \
    --function-name  my-function
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中 Lambda 函數的預留並行。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteFunctionConcurrency。

`delete-function-event-invoke-config`

以下程式碼範例顯示如何使用 `delete-function-event-invoke-config`。

AWS CLI

刪除非同步調用組態

下列`delete-function-event-invoke-config`範例會刪除指定函數之`GREEN`別名的非同步叫用組態。

```
aws lambda delete-function-event-invoke-config --function-name my-function:GREEN
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFunctionEventInvokeConfig。

delete-function

以下程式碼範例顯示如何使用 delete-function。

AWS CLI

範例 1：若要依函數名稱刪除 Lambda 函數

下列 delete-function 範例會透過指定函數的名稱來刪除名為 my-function 的 Lambda 函數。

```
aws lambda delete-function \
    --function-name my-function
```

此命令不會產生輸出。

範例 2：若要依函數 ARN 刪除 Lambda 函數

下列 delete-function 範例會透過指定函數的 ARN 來刪除名為 my-function 的 Lambda 函數。

```
aws lambda delete-function \
    --function-name arn:aws:lambda:us-west-2:123456789012:function:my-function
```

此命令不會產生輸出。

範例 3：若要依部分函數 ARN 刪除 Lambda 函數

下列 delete-function 範例會透過指定函數的部分 ARN 來刪除名為 my-function 的 Lambda 函數。

```
aws lambda delete-function \
    --function-name 123456789012:function:my-function
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 AWS Lambda 函數組態。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteFunction。

delete-layer-version

以下程式碼範例顯示如何使用 delete-layer-version。

AWS CLI

刪除 Lambda 層的版本

下列delete-layer-version範例會刪除名為 的 layer 第 2 版my-layer。

```
aws lambda delete-layer-version \
    --layer-name my-layer \
    --version-number 2
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 Lambda Layers。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLayerVersion。

delete-provisioned-concurrency-config

以下程式碼範例顯示如何使用 delete-provisioned-concurrency-config。

AWS CLI

若要刪除已佈建的並行組態

下列 delete-provisioned-concurrency-config 範例會刪除指定函數之 GREEN 別名的已佈建並行組態。

```
aws lambda delete-provisioned-concurrency-config \
    --function-name my-function \
    --qualifier GREEN
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteProvisionedConcurrencyConfig。

get-account-settings

以下程式碼範例顯示如何使用 get-account-settings。

AWS CLI

擷取 區域中您帳戶的詳細資訊 AWS

下列 get-account-settings 範例顯示帳戶的 Lambda 限制和用量資訊。

```
aws lambda get-account-settings
```

輸出：

```
{
    "AccountLimit": {
       "CodeSizeUnzipped": 262144000,
       "UnreservedConcurrentExecutions": 1000,
       "ConcurrentExecutions": 1000,
       "CodeSizeZipped": 52428800,
       "TotalCodeSize": 80530636800
    },
    "AccountUsage": {
       "FunctionCount": 4,
       "TotalCodeSize": 9426
    }
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 AWS Lambda 限制。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetAccountSettings。

get-alias

以下程式碼範例顯示如何使用 get-alias。

AWS CLI

若要擷取函數別名的詳細資訊

下列 get-alias 範例會顯示 my-function Lambda 函數中名為 LIVE 的別名詳細資訊。

```
aws lambda get-alias \
    --function-name my-function \
    --name LIVE
```

輸出：

```
{
    "FunctionVersion": "3",
    "Name": "LIVE",
```

```
    "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:LIVE",
    "RevisionId": "594f41fb-b85f-4c20-95c7-6ca5f2a92c93",
    "Description": "alias for live version of function"
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的設定 Lambda 函數別名。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetAlias。

get-event-source-mapping

以下程式碼範例顯示如何使用 get-event-source-mapping。

AWS CLI

擷取事件來源映射的詳細資訊

下列get-event-source-mapping範例顯示 SQS 佇列和 my-function Lambda 函數之間映射的詳細資訊。

```
aws lambda get-event-source-mapping \
    --uuid "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

輸出：

```
{
    "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "StateTransitionReason": "USER_INITIATED",
    "LastModified": 1569284520.333,
    "BatchSize": 5,
    "State": "Enabled",
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 Lambda 事件來源映射。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetEventSourceMapping。

get-function-concurrency

以下程式碼範例顯示如何使用 get-function-concurrency。

AWS CLI

若要檢視函數的預留並行設定

下列 get-function-concurrency 範例會擷取指定函數的預留並行設定。

```
aws lambda get-function-concurrency \
    --function-name my-function
```

輸出：

```
{
    "ReservedConcurrentExecutions": 250
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetFunctionConcurrency。

get-function-configuration

以下程式碼範例顯示如何使用 get-function-configuration。

AWS CLI

若要擷取 Lambda 函數的版本特定設定

下列 get-function-configuration 範例顯示 my-function 函數第 2 版的設定。

```
aws lambda get-function-configuration \
    --function-name  my-function:2
```

輸出：

```
{
    "FunctionName": "my-function",
    "LastModified": "2019-09-26T20:28:40.438+0000",
    "RevisionId": "e52502d4-9320-4688-9cd6-152a6ab7490d",
    "MemorySize": 256,
    "Version": "2",
    "Role": "arn:aws:iam::123456789012:role/service-role/my-function-role-uy3l9qyq",
    "Timeout": 3,
```

```
    "Runtime": "nodejs10.x",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "CodeSha256": "5tT2qgzYUHaqwR716pZ2dpkn/0J1FrzJmlKidWoaCgk=",
    "Description": "",
    "VpcConfig": {
        "SubnetIds": [],
        "VpcId": "",
        "SecurityGroupIds": []
    },
    "CodeSize": 304,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:2",
    "Handler": "index.handler"
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 AWS Lambda 函數組態。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetFunctionConfiguration。

get-function-event-invoke-config

以下程式碼範例顯示如何使用 get-function-event-invoke-config。

AWS CLI

檢視非同步調用組態

下列get-function-event-invoke-config範例會擷取指定函數之BLUE別名的非同步叫用組態。

```
aws lambda get-function-event-invoke-config \
    --function-name my-function:BLUE
```

輸出：

```
{
    "LastModified": 1577824396.653,
    "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-
function:BLUE",
    "MaximumRetryAttempts": 0,
    "MaximumEventAgeInSeconds": 3600,
```

```
    "DestinationConfig": {
        "OnSuccess": {},
        "OnFailure": {
            "Destination": "arn:aws:sqs:us-east-2:123456789012:failed-invocations"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFunctionEventInvokeConfig。

get-function

以下程式碼範例顯示如何使用 get-function。

AWS CLI

若要擷取函數相關資訊

下列 get-function 範例顯示 my-function 函數的相關資訊。

```
aws lambda get-function \
    --function-name  my-function
```

輸出：

```
{
    "Concurrency": {
        "ReservedConcurrentExecutions": 100
    },
    "Code": {
        "RepositoryType": "S3",
        "Location": "https://awslambda-us-west-2-tasks.s3.us-west-2.amazonaws.com/
snapshots/123456789012/my-function..."
    },
    "Configuration": {
        "TracingConfig": {
            "Mode": "PassThrough"
        },
        "Version": "$LATEST",
        "CodeSha256": "5tT2qgzYUHoqwR616pZ2dpkn/0J1FrzJmlKidWaaCgk=",
        "FunctionName": "my-function",
        "VpcConfig": {
```

```
            "SubnetIds": [],
            "VpcId": "",
            "SecurityGroupIds": []
        },
        "MemorySize": 128,
        "RevisionId": "28f0fb31-5c5c-43d3-8955-03e76c5c1075",
        "CodeSize": 304,
        "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
        "Handler": "index.handler",
        "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-role-
uy3l9qyq",
        "Timeout": 3,
        "LastModified": "2019-09-24T18:20:35.054+0000",
        "Runtime": "nodejs10.x",
        "Description": ""
    }
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 AWS Lambda 函數組態。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetFunction。

get-layer-version-by-arn

以下程式碼範例顯示如何使用 get-layer-version-by-arn。

AWS CLI

擷取 Lambda layer 版本的相關資訊

下列get-layer-version-by-arn範例顯示具有指定 Amazon Resource Name (ARN) 的 layer 版本相關資訊。

```
aws lambda get-layer-version-by-arn \
    --arn "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-Python311-
SciPy1x:2"
```

輸出：

```
{
    "LayerVersionArn": "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-
Python311-SciPy1x:2",
```

```
    "Description": "AWS Lambda SciPy layer for Python 3.11 (scipy-1.1.0,
 numpy-1.15.4) https://github.com/scipy/scipy/releases/tag/v1.1.0 https://
github.com/numpy/numpy/releases/tag/v1.15.4",
    "CreatedDate": "2023-10-12T10:09:38.398+0000",
    "LayerArn": "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-Python311-
SciPy1x",
    "Content": {
        "CodeSize": 41784542,
        "CodeSha256": "GGmv8ocUw4cly0T8HL0Vx/f5V4RmSCGNjDIslY4VskM=",
        "Location": "https://awslambda-us-west-2-layers.s3.us-west-2.amazonaws.com/
snapshots/123456789012/..."
    },
    "Version": 2,
    "CompatibleRuntimes": [
        "python3.11"
    ],
    "LicenseInfo": "SciPy: https://github.com/scipy/scipy/blob/main/LICENSE.txt,
 NumPy: https://github.com/numpy/numpy/blob/main/LICENSE.txt"
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 Lambda Layers。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLayerVersionByArn。

get-layer-version-policy

以下程式碼範例顯示如何使用 get-layer-version-policy。

AWS CLI

擷取 Lambda layer 版本的許可政策

下列get-layer-version-policy範例顯示名為 之 layer 第 1 版的政策資訊my-layer。

```
aws lambda get-layer-version-policy \
    --layer-name my-layer \
    --version-number 1
```

輸出：

```
{
    "Policy": {
```

```
        "Version":"2012-10-17",
        "Id":"default",
        "Statement":
        [
            {
                "Sid":"xaccount",
                "Effect":"Allow",
                "Principal": {"AWS":"arn:aws:iam::123456789012:root"},
                "Action":"lambda:GetLayerVersion",
                "Resource":"arn:aws:lambda:us-west-2:123456789012:layer:my-layer:1"
            }
        ]
    },
    "RevisionId": "c68f21d2-cbf0-4026-90f6-1375ee465cd0"
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 Lambda Layers。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLayerVersionPolicy。

get-layer-version

以下程式碼範例顯示如何使用 get-layer-version。

AWS CLI

擷取 Lambda layer 版本的相關資訊

下列get-layer-version範例顯示名為 之 layer 第 1 版的資訊my-layer。

```
aws lambda get-layer-version \
    --layer-name my-layer \
    --version-number 1
```

輸出：

```
{
    "Content": {
        "Location": "https://awslambda-us-east-2-layers.s3.us-east-2.amazonaws.com/
snapshots/123456789012/my-layer-4aaa2fbb-ff77-4b0a-ad92-5b78a716a96a?
versionId=27iWyA73cCAYqyH...",
        "CodeSha256": "tv9jJO+rPbXUUXuRKi7CwHzKtLDkDRJLB3cC3Z/ouXo=",
        "CodeSize": 169
```

```
    },
    "LayerArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-layer",
    "LayerVersionArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-layer:1",
    "Description": "My Python layer",
    "CreatedDate": "2018-11-14T23:03:52.894+0000",
    "Version": 1,
    "LicenseInfo": "MIT",
    "CompatibleRuntimes": [
        "python3.10",
        "python3.11"
    ]
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 Lambda Layers。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLayerVersion。

get-policy

以下程式碼範例顯示如何使用 get-policy。

AWS CLI

若要擷取函數、版本或別名的資源型 IAM 政策

下列 get-policy 範例顯示 my-function Lambda 函數的政策資訊。

```
aws lambda get-policy \
    --function-name my-function
```

輸出：

```
{
    "Policy": {
        "Version":"2012-10-17",
        "Id":"default",
        "Statement":
        [
            {
                "Sid":"iot-events",
                "Effect":"Allow",
                "Principal": {"Service":"iotevents.amazonaws.com"},
                "Action":"lambda:InvokeFunction",
```

```
                "Resource":"arn:aws:lambda:us-west-2:123456789012:function:my-
function"
            }
        ]
    },
    "RevisionId": "93017fc9-59cb-41dc-901b-4845ce4bf668"
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的使用 Lambda 的資源型政策。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetPolicy。

get-provisioned-concurrency-config

以下程式碼範例顯示如何使用 get-provisioned-concurrency-config。

AWS CLI

若要檢視已佈建的並行組態

下列 get-provisioned-concurrency-config 範例會顯示指定函數之 BLUE 別名的已佈建並行組態詳情。

```
aws lambda get-provisioned-concurrency-config \
    --function-name my-function \
    --qualifier BLUE
```

輸出：

```
{
    "RequestedProvisionedConcurrentExecutions": 100,
    "AvailableProvisionedConcurrentExecutions": 100,
    "AllocatedProvisionedConcurrentExecutions": 100,
    "Status": "READY",
    "LastModified": "2019-12-31T20:28:49+0000"
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI API 參考》中的 GetProvisionedConcurrencyConfig。

invoke

以下程式碼範例顯示如何使用 invoke。

AWS CLI

範例 1：若要同步調用 Lambda 函數

下列 invoke 範例會同步調用 my-function 函數。如果您使用 CLI AWS 第 2 版，則需要 cli-binary-format選項。如需詳細資訊，請參閱《AWS 命令列介面使用者指南》中的 AWS CLI 支援的全域命令列選項。

```
aws lambda invoke \
    --function-name my-function \
    --cli-binary-format raw-in-base64-out \
    --payload '{ "name": "Bob" }' \
    response.json
```

輸出：

```
{
    "ExecutedVersion": "$LATEST",
    "StatusCode": 200
}
```

如需詳細資訊，請參閱《Lambda 開發人員指南》中的同步叫用 Lambda 函數。 AWS

範例 2：若要非同步調用 Lambda 函數

下列 invoke 範例會非同步調用 my-function 函數。如果您使用 CLI AWS 第 2 版，則需要 cli-binary-format選項。如需詳細資訊，請參閱《AWS 命令列介面使用者指南》中的 AWS CLI 支援的全域命令列選項。

```
aws lambda invoke \
    --function-name my-function \
    --invocation-type Event \
    --cli-binary-format raw-in-base64-out \
    --payload '{ "name": "Bob" }' \
    response.json
```

輸出：

```
{
    "StatusCode": 202
```

```
}
```

如需詳細資訊，請參閱《Lambda 開發人員指南》中的非同步叫用 Lambda 函數。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 Invoke。

`list-aliases`

以下程式碼範例顯示如何使用 `list-aliases`。

AWS CLI

擷取 Lambda 函數的別名清單

下列`list-aliases`範例顯示 `my-function` Lambda 函數的別名清單。

```
aws lambda list-aliases \
    --function-name my-function
```

輸出：

```
{
    "Aliases": [
        {
            "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:BETA",
            "RevisionId": "a410117f-ab16-494e-8035-7e204bb7933b",
            "FunctionVersion": "2",
            "Name": "BETA",
            "Description": "alias for beta version of function"
        },
        {
            "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:LIVE",
            "RevisionId": "21d40116-f8b1-40ba-9360-3ea284da1bb5",
            "FunctionVersion": "1",
            "Name": "LIVE",
            "Description": "alias for live version of function"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的設定 Lambda 函數別名。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAliases。

`list-event-source-mappings`

以下程式碼範例顯示如何使用 `list-event-source-mappings`。

AWS CLI

列出函數的事件來源映射

下列`list-event-source-mappings`範例顯示 `my-function` Lambda 函數的事件來源映射清單。

```
aws lambda list-event-source-mappings \
    --function-name my-function
```

輸出：

```
{
    "EventSourceMappings": [
        {
            "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "StateTransitionReason": "USER_INITIATED",
            "LastModified": 1569284520.333,
            "BatchSize": 5,
            "State": "Enabled",
            "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function",
            "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 Lambda 事件來源映射。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEventSourceMappings。

`list-function-event-invoke-configs`

以下程式碼範例顯示如何使用 `list-function-event-invoke-configs`。

AWS CLI

檢視非同步調用組態的清單

下列list-function-event-invoke-configs範例列出指定函數的非同步調用組態。

```
aws lambda list-function-event-invoke-configs \
    --function-name my-function
```

輸出：

```
{
    "FunctionEventInvokeConfigs": [
        {
            "LastModified": 1577824406.719,
            "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-
function:GREEN",
            "MaximumRetryAttempts": 2,
            "MaximumEventAgeInSeconds": 1800
        },
        {
            "LastModified": 1577824396.653,
            "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-
function:BLUE",
            "MaximumRetryAttempts": 0,
            "MaximumEventAgeInSeconds": 3600
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFunctionEventInvokeConfigs。

list-functions

以下程式碼範例顯示如何使用 list-functions。

AWS CLI

若要擷取 Lambda 函數的清單

下列 list-functions 範例會顯示目前使用者的所有函數清單。

```
aws lambda list-functions
```

輸出：

```
{
    "Functions": [
        {
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "Version": "$LATEST",
            "CodeSha256": "dBG9m8SGdmlEjw/JYXlhhvCrAv5TxvXsbL/RMr0fT/I=",
            "FunctionName": "helloworld",
            "MemorySize": 128,
            "RevisionId": "1718e831-badf-4253-9518-d0644210af7b",
            "CodeSize": 294,
            "FunctionArn": "arn:aws:lambda:us-
west-2:123456789012:function:helloworld",
            "Handler": "helloworld.handler",
            "Role": "arn:aws:iam::123456789012:role/service-role/MyTestFunction-
role-zgur6bf4",
            "Timeout": 3,
            "LastModified": "2023-09-23T18:32:33.857+0000",
            "Runtime": "nodejs18.x",
            "Description": ""
        },
        {
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "Version": "$LATEST",
            "CodeSha256": "sU0cJ2/hOZevwV/lTxCuQqK3gDZP3i8gUoqUUVRmY6E=",
            "FunctionName": "my-function",
            "VpcConfig": {
                "SubnetIds": [],
                "VpcId": "",
                "SecurityGroupIds": []
            },
            "MemorySize": 256,
            "RevisionId": "93017fc9-59cb-41dc-901b-4845ce4bf668",
            "CodeSize": 266,
            "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function",
```

```
            "Handler": "index.handler",
            "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qyq",
            "Timeout": 3,
            "LastModified": "2023-10-01T16:47:28.490+0000",
            "Runtime": "nodejs18.x",
            "Description": ""
        },
        {
            "Layers": [
                {
                    "CodeSize": 41784542,
                    "Arn": "arn:aws:lambda:us-west-2:420165488524:layer:AWSLambda-
Python37-SciPy1x:2"
                },
                {
                    "CodeSize": 4121,
                    "Arn": "arn:aws:lambda:us-
west-2:123456789012:layer:pythonLayer:1"
                }
            ],
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "Version": "$LATEST",
            "CodeSha256": "ZQukCqxtkqFgyF2cU41Avj99TKQ/hNihPtDtRcc08mI=",
            "FunctionName": "my-python-function",
            "VpcConfig": {
                "SubnetIds": [],
                "VpcId": "",
                "SecurityGroupIds": []
            },
            "MemorySize": 128,
            "RevisionId": "80b4eabc-acf7-4ea8-919a-e874c213707d",
            "CodeSize": 299,
            "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
python-function",
            "Handler": "lambda_function.lambda_handler",
            "Role": "arn:aws:iam::123456789012:role/service-role/my-python-function-
role-z5g7dr6n",
            "Timeout": 3,
            "LastModified": "2023-10-01T19:40:41.643+0000",
            "Runtime": "python3.11",
            "Description": ""
```

```
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 AWS Lambda 函數組態。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListFunctions。

`list-layer-versions`

以下程式碼範例顯示如何使用 `list-layer-versions`。

AWS CLI

列出 AWS Lambda 層的版本

下列`list-layers-versions`範例顯示名為 之 layer 版本的相關資訊`my-layer`。

```
aws lambda list-layer-versions \
    --layer-name my-layer
```

輸出：

```
{
    "Layers": [
        {
            "LayerVersionArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-
layer:2",
            "Version": 2,
            "Description": "My layer",
            "CreatedDate": "2023-11-15T00:37:46.592+0000",
            "CompatibleRuntimes": [
                "python3.10",
                "python3.11"
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 Lambda Layers。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListLayerVersions。

list-layers

以下程式碼範例顯示如何使用 list-layers。

AWS CLI

列出與函數執行時間相容的層

下列list-layers範例顯示與 Python 3.11 執行時間相容的 layer 的相關資訊。

```
aws lambda list-layers \
    --compatible-runtime python3.11
```

輸出：

```
{
    "Layers": [
        {
            "LayerName": "my-layer",
            "LayerArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-layer",
            "LatestMatchingVersion": {
                "LayerVersionArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-
layer:2",
                "Version": 2,
                "Description": "My layer",
                "CreatedDate": "2023-11-15T00:37:46.592+0000",
                "CompatibleRuntimes": [
                    "python3.10",
                    "python3.11"
                ]
            }
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 Lambda Layers。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListLayers。

list-provisioned-concurrency-configs

以下程式碼範例顯示如何使用 list-provisioned-concurrency-configs。

AWS CLI

若要取得已佈建並行組態的清單。

下列 list-provisioned-concurrency-configs 範例會列出指定函數的已佈建並行組態。

```
aws lambda list-provisioned-concurrency-configs \
    --function-name my-function
```

輸出：

```
{
    "ProvisionedConcurrencyConfigs": [
        {
            "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-
function:GREEN",
            "RequestedProvisionedConcurrentExecutions": 100,
            "AvailableProvisionedConcurrentExecutions": 100,
            "AllocatedProvisionedConcurrentExecutions": 100,
            "Status": "READY",
            "LastModified": "2019-12-31T20:29:00+0000"
        },
        {
            "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-
function:BLUE",
            "RequestedProvisionedConcurrentExecutions": 100,
            "AvailableProvisionedConcurrentExecutions": 100,
            "AllocatedProvisionedConcurrentExecutions": 100,
            "Status": "READY",
            "LastModified": "2019-12-31T20:28:49+0000"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListProvisionedConcurrencyConfigs。

list-tags

以下程式碼範例顯示如何使用 list-tags。

AWS CLI

若要擷取 Lambda 函數的標籤清單

下列 list-tags 範例顯示連接至 my-function Lambda 函數的標籤。

```
aws lambda list-tags \
    --resource arn:aws:lambda:us-west-2:123456789012:function:my-function
```

輸出：

```
{
    "Tags": {
        "Category": "Web Tools",
        "Department": "Sales"
    }
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的標記 Lambda 函數。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTags。

list-versions-by-function

以下程式碼範例顯示如何使用 list-versions-by-function。

AWS CLI

若要擷取函數的版本清單

下列 list-versions-by-function 範例會顯示 my-function Lambda 函數的版本清單。

```
aws lambda list-versions-by-function \
    --function-name my-function
```

輸出：

```
{
    "Versions": [
        {
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "Version": "$LATEST",
            "CodeSha256": "sU0cJ2/hOZevwV/lTxCuQqK3gDZP3i8gUoqUUVRmY6E=",
            "FunctionName": "my-function",
```

```
            "VpcConfig": {
                "SubnetIds": [],
                "VpcId": "",
                "SecurityGroupIds": []
            },
            "MemorySize": 256,
            "RevisionId": "93017fc9-59cb-41dc-901b-4845ce4bf668",
            "CodeSize": 266,
            "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:$LATEST",
            "Handler": "index.handler",
            "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qyq",
            "Timeout": 3,
            "LastModified": "2019-10-01T16:47:28.490+0000",
            "Runtime": "nodejs10.x",
            "Description": ""
        },
        {
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "Version": "1",
            "CodeSha256": "5tT2qgzYUHoqwR616pZ2dpkn/0J1FrzJmlKidWaaCgk=",
            "FunctionName": "my-function",
            "VpcConfig": {
                "SubnetIds": [],
                "VpcId": "",
                "SecurityGroupIds": []
            },
            "MemorySize": 256,
            "RevisionId": "949c8914-012e-4795-998c-e467121951b1",
            "CodeSize": 304,
            "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:1",
            "Handler": "index.handler",
            "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qyq",
            "Timeout": 3,
            "LastModified": "2019-09-26T20:28:40.438+0000",
            "Runtime": "nodejs10.x",
            "Description": "new version"
        },
        {
```

```
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "Version": "2",
            "CodeSha256": "sU0cJ2/hOZevwV/lTxCuQqK3gDZP3i8gUoqUUVRmY6E=",
            "FunctionName": "my-function",
            "VpcConfig": {
                "SubnetIds": [],
                "VpcId": "",
                "SecurityGroupIds": []
            },
            "MemorySize": 256,
            "RevisionId": "cd669f21-0f3d-4e1c-9566-948837f2e2ea",
            "CodeSize": 266,
            "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:2",
            "Handler": "index.handler",
            "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qyq",
            "Timeout": 3,
            "LastModified": "2019-10-01T16:47:28.490+0000",
            "Runtime": "nodejs10.x",
            "Description": "newer version"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的設定 Lambda 函數別名。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListVersionsByFunction。

publish-layer-version

以下程式碼範例顯示如何使用 publish-layer-version。

AWS CLI

建立 Lambda layer 版本

下列publish-layer-version範例會建立新的 Python 程式庫層版本。命令會擷取指定 S3 儲存貯layer.zip體中名為 檔案的 layer 內容。

```
aws lambda publish-layer-version \
```

```
    --layer-name my-layer \
    --description "My Python layer" \
    --license-info "MIT" \
    --content S3Bucket=lambda-layers-us-west-2-123456789012,S3Key=layer.zip \
    --compatible-runtimes python3.10 python3.11
```

輸出：

```
{
    "Content": {
        "Location": "https://awslambda-us-west-2-layers.s3.us-west-2.amazonaws.com/
snapshots/123456789012/my-layer-4aaa2fbb-ff77-4b0a-ad92-5b78a716a96a?
versionId=27iWyA73cCAYqyH...",
        "CodeSha256": "tv9jJO+rPbXUUXuRKi7CwHzKtLDkDRJLB3cC3Z/ouXo=",
        "CodeSize": 169
    },
    "LayerArn": "arn:aws:lambda:us-west-2:123456789012:layer:my-layer",
    "LayerVersionArn": "arn:aws:lambda:us-west-2:123456789012:layer:my-layer:1",
    "Description": "My Python layer",
    "CreatedDate": "2023-11-14T23:03:52.894+0000",
    "Version": 1,
    "LicenseInfo": "MIT",
    "CompatibleRuntimes": [
        "python3.10",
        "python3.11"
    ]
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 Lambda Layers。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PublishLayerVersion。

publish-version

以下程式碼範例顯示如何使用 publish-version。

AWS CLI

若要建立新版本的函數

以下 publish-version 範例會發布新版本的 my-function Lambda 函數。

```
aws lambda publish-version \
```

```
    --function-name my-function
```

輸出：

```
{
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "CodeSha256": "dBG9m8SGdmlEjw/JYXlhhvCrAv5TxvXsbL/RMr0fT/I=",
    "FunctionName": "my-function",
    "CodeSize": 294,
    "RevisionId": "f31d3d39-cc63-4520-97d4-43cd44c94c20",
    "MemorySize": 128,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:3",
    "Version": "2",
    "Role": "arn:aws:iam::123456789012:role/service-role/MyTestFunction-role-
zgur6bf4",
    "Timeout": 3,
    "LastModified": "2019-09-23T18:32:33.857+0000",
    "Handler": "my-function.handler",
    "Runtime": "nodejs10.x",
    "Description": ""
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的設定 Lambda 函數別名。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PublishVersion。

put-function-concurrency

以下程式碼範例顯示如何使用 put-function-concurrency。

AWS CLI

若要設定函數的預留並行限制

下列 put-function-concurrency 範例會設定 my-function 函數的 100 個預留並行執行。

```
aws lambda put-function-concurrency \
    --function-name  my-function  \
    --reserved-concurrent-executions 100
```

輸出：

```
{
    "ReservedConcurrentExecutions": 100
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中 Lambda 函數的預留並行。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutFunctionConcurrency。

put-function-event-invoke-config

以下程式碼範例顯示如何使用 put-function-event-invoke-config。

AWS CLI

設定非同步調用的錯誤處理

下列put-function-event-invoke-config範例會設定一小時的最大事件存留期，並停用指定函數的重試。

```
aws lambda put-function-event-invoke-config \
    --function-name my-function \
    --maximum-event-age-in-seconds 3600 \
    --maximum-retry-attempts 0
```

輸出：

```
{
    "LastModified": 1573686021.479,
    "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-function:
$LATEST",
    "MaximumRetryAttempts": 0,
    "MaximumEventAgeInSeconds": 3600,
    "DestinationConfig": {
        "OnSuccess": {},
        "OnFailure": {}
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutFunctionEventInvokeConfig。

put-provisioned-concurrency-config

以下程式碼範例顯示如何使用 put-provisioned-concurrency-config。

AWS CLI

若要配置佈建並行

下列 put-provisioned-concurrency-config 範例會為指定函數的 BLUE 別名配置 100 個佈建並行。

```
aws lambda put-provisioned-concurrency-config \
    --function-name my-function \
    --qualifier BLUE \
    --provisioned-concurrent-executions 100
```

輸出：

```
{
    "Requested ProvisionedConcurrentExecutions": 100,
    "Allocated ProvisionedConcurrentExecutions": 0,
    "Status": "IN_PROGRESS",
    "LastModified": "2019-11-21T19:32:12+0000"
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutProvisionedConcurrencyConfig。

remove-layer-version-permission

以下程式碼範例顯示如何使用 remove-layer-version-permission。

AWS CLI

刪除 layer-version 許可

下列remove-layer-version-permission範例會刪除 帳戶設定 layer 版本的許可。

```
aws lambda remove-layer-version-permission \
    --layer-name my-layer \
    --statement-id xaccount \
    --version-number 1
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 Lambda Layers。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveLayerVersionPermission。

`remove-permission`

以下程式碼範例顯示如何使用 `remove-permission`。

AWS CLI

若要從現有的 Lambda 函數中移除許可

下列 `remove-permission` 範例會移除許可以調用名為 `my-function` 的函數。

```
aws lambda remove-permission \
    --function-name my-function \
    --statement-id sns
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的使用 Lambda 的資源型政策。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 RemovePermission。

`tag-resource`

以下程式碼範例顯示如何使用 `tag-resource`。

AWS CLI

若要將標籤新增至現有 Lambda 函數

下列 `tag-resource` 範例會將索引鍵名稱為 `DEPARTMENT` 且值為 `Department A` 的標籤新增至指定的 Lambda 函數。

```
aws lambda tag-resource \
    --resource arn:aws:lambda:us-west-2:123456789012:function:my-function \
    --tags "DEPARTMENT=Department A"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的標記 Lambda 函數。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

若要從現有的 Lambda 函數中移除標籤

下列 untag-resource 範例會從 my-function Lambda 函數中移除索引鍵名稱為 DEPARTMENT 的標籤。

```
aws lambda untag-resource \
    --resource arn:aws:lambda:us-west-2:123456789012:function:my-function \
    --tag-keys DEPARTMENT
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的標記 Lambda 函數。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-alias

以下程式碼範例顯示如何使用 update-alias。

AWS CLI

若要更新函數別名

下列 update-alias 範例會更新名為 LIVE 的別名，以指向 my-function Lambda 函數的第 3 版。

```
aws lambda update-alias \
    --function-name my-function \
    --function-version 3 \
    --name LIVE
```

輸出：

```
{
    "FunctionVersion": "3",
    "Name": "LIVE",
    "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:LIVE",
    "RevisionId": "594f41fb-b85f-4c20-95c7-6ca5f2a92c93",
    "Description": "alias for live version of function"
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的設定 Lambda 函數別名。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateAlias。

update-event-source-mapping

以下程式碼範例顯示如何使用 update-event-source-mapping。

AWS CLI

更新事件來源與 AWS Lambda 函數之間的映射

下列update-event-source-mapping範例會將指定映射中的批次大小更新為 8。

```
aws lambda update-event-source-mapping \
    --uuid  "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \
    --batch-size 8
```

輸出：

```
{
    "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "StateTransitionReason": "USER_INITIATED",
    "LastModified": 1569284520.333,
    "BatchSize": 8,
    "State": "Updating",
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 Lambda 事件來源映射。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateEventSourceMapping。

update-function-code

以下程式碼範例顯示如何使用 update-function-code。

AWS CLI

若要更新 Lambda 函數的程式碼

下列 update-function-code 範例會使用指定 zip 檔案的內容替換 my-function 函數未發布 ($LATEST) 版本的程式碼。

```
aws lambda update-function-code \
    --function-name  my-function \
    --zip-file fileb://my-function.zip
```

輸出：

```
{
    "FunctionName": "my-function",
    "LastModified": "2019-09-26T20:28:40.438+0000",
    "RevisionId": "e52502d4-9320-4688-9cd6-152a6ab7490d",
    "MemorySize": 256,
    "Version": "$LATEST",
    "Role": "arn:aws:iam::123456789012:role/service-role/my-function-role-uy3l9qyq",
    "Timeout": 3,
    "Runtime": "nodejs10.x",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "CodeSha256": "5tT2qgzYUHaqwR716pZ2dpkn/0J1FrzJmlKidWoaCgk=",
    "Description": "",
    "VpcConfig": {
        "SubnetIds": [],
        "VpcId": "",
        "SecurityGroupIds": []
    },
    "CodeSize": 304,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "Handler": "index.handler"
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 AWS Lambda 函數組態。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateFunctionCode。

update-function-configuration

以下程式碼範例顯示如何使用 update-function-configuration。

AWS CLI

若要修改函數的組態

下列 update-function-configuration 範例會將 my-function 函數未發布 ($LATEST) 版本的記憶體大小修改為 256 MB。

```
aws lambda update-function-configuration \
    --function-name  my-function \
    --memory-size 256
```

輸出：

```
{
    "FunctionName": "my-function",
    "LastModified": "2019-09-26T20:28:40.438+0000",
    "RevisionId": "e52502d4-9320-4688-9cd6-152a6ab7490d",
    "MemorySize": 256,
    "Version": "$LATEST",
    "Role": "arn:aws:iam::123456789012:role/service-role/my-function-role-uy3l9qyq",
    "Timeout": 3,
    "Runtime": "nodejs10.x",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "CodeSha256": "5tT2qgzYUHaqwR716pZ2dpkn/0J1FrzJmlKidWoaCgk=",
    "Description": "",
    "VpcConfig": {
        "SubnetIds": [],
        "VpcId": "",
        "SecurityGroupIds": []
    },
    "CodeSize": 304,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "Handler": "index.handler"
}
```

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 AWS Lambda 函數組態。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateFunctionConfiguration。

update-function-event-invoke-config

以下程式碼範例顯示如何使用 update-function-event-invoke-config。

AWS CLI

更新非同步調用組態

下列update-function-event-invoke-config範例會將故障時目的地新增至指定函數的現有非同步叫用組態。

```
aws lambda update-function-event-invoke-config \
    --function-name my-function \
    --destination-config '{"OnFailure":{"Destination": "arn:aws:sqs:us-
east-2:123456789012:destination"}}'
```

輸出：

```
{
    "LastModified": 1573687896.493,
    "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-function:
$LATEST",
    "MaximumRetryAttempts": 0,
    "MaximumEventAgeInSeconds": 3600,
    "DestinationConfig": {
        "OnSuccess": {},
        "OnFailure": {
            "Destination": "arn:aws:sqs:us-east-2:123456789012:destination"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateFunctionEventInvokeConfig。

使用 的授權管理員範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 License Manager 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-license-configuration`

以下程式碼範例顯示如何使用 `create-license-configuration`。

AWS CLI

範例 1：建立授權組態

下列`create-license-configuration`範例會建立硬性限制為 10 個核心的授權組態。

```
aws license-manager create-license-configuration --name my-license-configuration \
    --license-counting-type Core \
    --license-count 10 \
    --license-count-hard-limit
```

輸出：

```
{
  "LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba41EXAMPLE1111"
}
```

範例 2：建立授權組態

下列`create-license-configuration`範例會建立軟性限制為 100 個 vCPUs的授權組態。它使用規則來啟用 vCPU 最佳化。

```
aws license-manager create-license-configuration --name my-license-configuration
    --license-counting-type vCPU \
    --license-count 100 \
    --license-rules "#honorVcpuOptimization=true"
```

輸出：

```
{
  "LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba41EXAMPLE2222"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLicenseConfiguration。

delete-license-configuration

以下程式碼範例顯示如何使用 delete-license-configuration。

AWS CLI

刪除授權組態

下列delete-license-configuration範例會刪除指定的授權組態。

```
aws license-manager delete-license-configuration \
    --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLicenseConfiguration。

get-license-configuration

以下程式碼範例顯示如何使用 get-license-configuration。

AWS CLI

取得授權組態資訊

下列get-license-configuration範例顯示指定授權組態的詳細資訊。

```
aws license-manager get-license-configuration \
    --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

輸出：

```
{
    "LicenseConfigurationId": "lic-38b658717b87478aaa7c00883EXAMPLE",
    "LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE",
    "Name": "my-license-configuration",
    "LicenseCountingType": "vCPU",
    "LicenseRules": [],
    "LicenseCountHardLimit": false,
    "ConsumedLicenses": 0,
    "Status": "AVAILABLE",
    "OwnerAccountId": "123456789012",
    "ConsumedLicenseSummaryList": [
        {
            "ResourceType": "EC2_INSTANCE",
            "ConsumedLicenses": 0
        },
        {
            "ResourceType": "EC2_HOST",
            "ConsumedLicenses": 0
        },
        {
            "ResourceType": "SYSTEMS_MANAGER_MANAGED_INSTANCE",
            "ConsumedLicenses": 0
        }
    ],
    "ManagedResourceSummaryList": [
        {
            "ResourceType": "EC2_INSTANCE",
            "AssociationCount": 0
        },
        {
            "ResourceType": "EC2_HOST",
            "AssociationCount": 0
        },
        {
            "ResourceType": "EC2_AMI",
            "AssociationCount": 2
        },
        {
            "ResourceType": "SYSTEMS_MANAGER_MANAGED_INSTANCE",
            "AssociationCount": 0
        }
```

```
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLicenseConfiguration。

get-service-settings

以下程式碼範例顯示如何使用 get-service-settings。

AWS CLI

取得 License Manager 設定

下列get-service-settings範例顯示目前區域中 License Manager 的服務設定。

```
aws license-manager get-service-settings
```

下列顯示停用跨帳戶資源探索時的輸出範例。

```
{
    "OrganizationConfiguration": {
        "EnableIntegration": false
    },
    "EnableCrossAccountsDiscovery": false
}
```

下列顯示啟用跨帳戶資源探索時的輸出範例。

```
{
    "S3BucketArn": "arn:aws:s3:::aws-license-manager-service-c22d6279-35c4-47c4-bb",
    "OrganizationConfiguration": {
        "EnableIntegration": true
    },
    "EnableCrossAccountsDiscovery": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetServiceSettings。

list-associations-for-license-configuration

以下程式碼範例顯示如何使用 list-associations-for-license-configuration。

AWS CLI

取得授權組態的關聯

下列list-associations-for-license-configuration範例顯示指定授權組態之關聯的詳細資訊。

```
aws license-manager list-associations-for-license-configuration \
    --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

輸出：

```
{
    "LicenseConfigurationAssociations": [
        {
            "ResourceArn": "arn:aws:ec2:us-west-2::image/ami-1234567890abcdef0",
            "ResourceType": "EC2_AMI",
            "ResourceOwnerId": "123456789012",
            "AssociationTime": 1568825118.617
        },
        {
            "ResourceArn": "arn:aws:ec2:us-west-2::image/ami-0abcdef1234567890",
            "ResourceType": "EC2_AMI",
            "ResourceOwnerId": "123456789012",
            "AssociationTime": 1568825118.946
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAssociationsForLicenseConfiguration。

list-license-configurations

以下程式碼範例顯示如何使用 list-license-configurations。

AWS CLI

範例 1：列出所有授權組態

下列list-license-configurations範例列出所有授權組態。

```
aws license-manager list-license-configurations
```

輸出：

```
{
    "LicenseConfigurations": [
        {
            "LicenseConfigurationId": "lic-6eb6586f508a786a2ba4f56c1EXAMPLE",
            "LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE",
            "Name": "my-license-configuration",
            "LicenseCountingType": "Core",
            "LicenseRules": [],
            "LicenseCount": 10,
            "LicenseCountHardLimit": true,
            "ConsumedLicenses": 0,
            "Status": "AVAILABLE",
            "OwnerAccountId": "123456789012",
            "ConsumedLicenseSummaryList": [
                {
                    "ResourceType": "EC2_INSTANCE",
                    "ConsumedLicenses": 0
                },
                {
                    "ResourceType": "EC2_HOST",
                    "ConsumedLicenses": 0
                },
                {
                    "ResourceType": "SYSTEMS_MANAGER_MANAGED_INSTANCE",
                    "ConsumedLicenses": 0
                }
            ],
            "ManagedResourceSummaryList": [
                {
                    "ResourceType": "EC2_INSTANCE",
                    "AssociationCount": 0
                },
                {
                    "ResourceType": "EC2_HOST",
                    "AssociationCount": 0
                },
                {
                    "ResourceType": "EC2_AMI",
```

```
                    "AssociationCount": 0
                },
                {
                    "ResourceType": "SYSTEMS_MANAGER_MANAGED_INSTANCE",
                    "AssociationCount": 0
                }
            ]
        },
        {
            ...
        }
    ]
}
```

範例 2：列出特定授權組態

下列`list-license-configurations`範例僅列出指定的授權組態。

```
aws license-manager list-license-configurations \
    --license-configuration-arns arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListLicenseConfigurations。

list-license-specifications-for-resource

以下程式碼範例顯示如何使用 `list-license-specifications-for-resource`。

AWS CLI

列出資源的授權組態

下列`list-license-specifications-for-resource`範例列出與指定 Amazon Machine Image (AMI) 相關聯的授權組態。

```
aws license-manager list-license-specifications-for-resource \
    --resource-arn arn:aws:ec2:us-west-2::image/ami-1234567890abcdef0
```

輸出：

```
{
```

```
    "LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListLicenseSpecificationsForResource。

`list-resource-inventory`

以下程式碼範例顯示如何使用 `list-resource-inventory`。

AWS CLI

列出資源庫存中的資源

下列`list-resource-inventory`範例列出使用 Systems Manager 庫存管理的資源。

```
aws license-manager list-resource-inventory
```

輸出：

```
{
    "ResourceInventoryList": [
        {
            "Platform": "Red Hat Enterprise Linux Server",
            "ResourceType": "EC2Instance",
            "PlatformVersion": "7.4",
            "ResourceArn": "arn:aws:ec2:us-west-2:1234567890129:instance/
i-05d3cdfb05bd36376",
            "ResourceId": "i-05d3cdfb05bd36376",
            "ResourceOwningAccountId": "1234567890129"
        },
        {
            "Platform": "Amazon Linux",
            "ResourceType": "EC2Instance",
            "PlatformVersion": "2",
            "ResourceArn": "arn:aws:ec2:us-west-2:1234567890129:instance/
i-0b1d036cfd4594808",
            "ResourceId": "i-0b1d036cfd4594808",
            "ResourceOwningAccountId": "1234567890129"
        },
```

```
        {
            "Platform": "Microsoft Windows Server 2019 Datacenter",
            "ResourceType": "EC2Instance",
            "PlatformVersion": "10.0.17763",
            "ResourceArn": "arn:aws:ec2:us-west-2:1234567890129:instance/
i-0cdb3b54a2a8246ad",
            "ResourceId": "i-0cdb3b54a2a8246ad",
            "ResourceOwningAccountId": "1234567890129"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourceInventory。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出授權組態的標籤

下列`list-tags-for-resource`範例列出指定授權組態的標籤。

```
aws license-manager list-tags-for-resource \
    --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-
configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

輸出：

```
{
    "Tags": [
        {
            "Key": "project",
            "Value": "lima"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

list-usage-for-license-configuration

以下程式碼範例顯示如何使用 list-usage-for-license-configuration。

AWS CLI

列出用於授權組態的授權

下列list-usage-for-license-configuration範例列出使用指定授權組態之授權的資源相關資訊。例如，如果授權類型為 vCPU，則任何執行個體每個 vCPU 都會耗用一個授權。

```
aws license-manager list-usage-for-license-configuration \
    --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

輸出：

```
{
    "LicenseConfigurationUsageList": [
        {
            "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-04a636d18e83cfacb",
            "ResourceType": "EC2_INSTANCE",
            "ResourceStatus": "running",
            "ResourceOwnerId": "123456789012",
            "AssociationTime": 1570892850.519,
            "ConsumedLicenses": 2
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListUsageForLicenseConfiguration。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

新增標籤授權組態

下列tag-resource範例會將指定的標籤（索引鍵名稱和值）新增至指定的授權組態。

```
aws license-manager tag-resource \
    --tags Key=project,Value=lima \
    --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-
configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從授權組態中移除標籤

下列untag-resource範例會從指定的授權組態中移除指定的標籤（金鑰名稱和資源）。

```
aws license-manager untag-resource \
    --tag-keys project \
    --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-
configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-license-configuration

以下程式碼範例顯示如何使用 update-license-configuration。

AWS CLI

更新授權組態

下列update-license-configuration範例會更新指定的授權組態，以移除硬性限制。

```
aws license-manager update-license-configuration \
    --no-license-count-hard-limit \
    --license-configuration-arn arn:aws:license-manager:us-
west-2:880185128111:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

此命令不會產生輸出。

下列update-license-configuration範例會更新指定的授權組態，將其狀態變更為DISABLED。

```
aws license-manager update-license-configuration \
    --license-configuration-status DISABLED
    --license-configuration-arn arn:aws:license-manager:us-
west-2:880185128111:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateLicenseConfiguration。

update-license-specifications-for-resource

以下程式碼範例顯示如何使用 update-license-specifications-for-resource。

AWS CLI

更新資源的授權組態

下列update-license-specifications-for-resource範例會移除一個授權組態並新增另一個授權組態，以取代與指定 Amazon Machine Image (AMI) 相關聯的授權組態。

```
aws license-manager update-license-specifications-for-resource \
    --resource-arn arn:aws:ec2:us-west-2::image/ami-1234567890abcdef0 \
    --remove-license-specifications LicenseConfigurationArn=arn:aws:license-
manager:us-west-2:123456789012:license-
configuration:lic-38b658717b87478aaa7c00883EXAMPLE \
    --add-license-specifications LicenseConfigurationArn=arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-42b6deb06e5399a980d555927EXAMPLE
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateLicenseSpecificationsForResource。

update-service-settings

以下程式碼範例顯示如何使用 update-service-settings。

AWS CLI

更新 License Manager 設定

下列update-service-settings範例會啟用目前 AWS 區域中 License Manager 的跨帳戶資源探索。Amazon S3 儲存貯體是 Systems Manager 庫存所需的資源資料同步。

```
aws license-manager update-service-settings \
    --organization-configuration EnableIntegration=true \
    --enable-cross-accounts-discovery \
    --s3-bucket-arn arn:aws:s3:::aws-license-manager-service-abcd1234EXAMPLE
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateServiceSettings。

Lightsail 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 Lightsail。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`allocate-static-ip`

以下程式碼範例顯示如何使用 allocate-static-ip。

AWS CLI

建立靜態 IP

下列allocate-static-ip範例會建立指定的靜態 IP，此 IP 可以連接到執行個體。

```
aws lightsail allocate-static-ip \
    --static-ip-name StaticIp-1
```

輸出：

```
{
    "operations": [
        {
            "id": "b5d06d13-2f19-4683-889f-dEXAMPLEed79",
            "resourceName": "StaticIp-1",
            "resourceType": "StaticIp",
            "createdAt": 1571071325.076,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "AllocateStaticIp",
            "status": "Succeeded",
            "statusChangedAt": 1571071325.274
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AllocateStaticIp。

attach-disk

以下程式碼範例顯示如何使用 attach-disk。

AWS CLI

將區塊儲存磁碟連接至執行個體

下列attach-disk範例會將磁碟連接至WordPress_Multisite-1磁碟路徑為 的Disk-1執行個體 /dev/xvdf

```
aws lightsail attach-disk \
    --disk-name Disk-1 \
    --disk-path /dev/xvdf \
    --instance-name WordPress_Multisite-1
```

輸出：

```
{
    "operations": [
        {
            "id": "10a08267-19ce-43be-b913-6EXAMPLE7e80",
            "resourceName": "Disk-1",
            "resourceType": "Disk",
            "createdAt": 1571071465.472,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "WordPress_Multisite-1",
            "operationType": "AttachDisk",
            "status": "Started",
            "statusChangedAt": 1571071465.472
        },
        {
            "id": "2912c477-5295-4539-88c9-bEXAMPLEd1f0",
            "resourceName": "WordPress_Multisite-1",
            "resourceType": "Instance",
            "createdAt": 1571071465.474,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "Disk-1",
            "operationType": "AttachDisk",
            "status": "Started",
            "statusChangedAt": 1571071465.474
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachDisk。

attach-instances-to-load-balancer

以下程式碼範例顯示如何使用 attach-instances-to-load-balancer。

AWS CLI

將執行個體連接至負載平衡器

下列attach-instances-to-load-balancer範例會將執行個體 MEAN-1、 MEAN-2和 MEAN-3 連接至負載平衡器 LoadBalancer-1。

```
aws lightsail attach-instances-to-load-balancer \
    --instance-names {"MEAN-1","MEAN-2","MEAN-3"} \
    --load-balancer-name LoadBalancer-1
```

輸出：

```
{
    "operations": [
        {
            "id": "8055d19d-abb2-40b9-b527-1EXAMPLE3c7b",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571071699.892,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "MEAN-2",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Started",
            "statusChangedAt": 1571071699.892
        },
        {
            "id": "c35048eb-8538-456a-a118-0EXAMPLEfb73",
            "resourceName": "MEAN-2",
            "resourceType": "Instance",
            "createdAt": 1571071699.887,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "LoadBalancer-1",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Started",
```

```
            "statusChangedAt": 1571071699.887
        },
        {
            "id": "910d09e0-adc5-4372-bc2e-0EXAMPLEd891",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571071699.882,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "MEAN-3",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Started",
            "statusChangedAt": 1571071699.882
        },
        {
            "id": "178b18ac-43e8-478c-9bed-1EXAMPLE4755",
            "resourceName": "MEAN-3",
            "resourceType": "Instance",
            "createdAt": 1571071699.901,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "LoadBalancer-1",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Started",
            "statusChangedAt": 1571071699.901
        },
        {
            "id": "fb62536d-2a98-4190-a6fc-4EXAMPLE7470",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571071699.885,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "MEAN-1",
            "operationType": "AttachInstancesToLoadBalancer",
```

```
            "status": "Started",
            "statusChangedAt": 1571071699.885
        },
        {
            "id": "787dac0d-f98d-46c3-8571-3EXAMPLE5a85",
            "resourceName": "MEAN-1",
            "resourceType": "Instance",
            "createdAt": 1571071699.901,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "LoadBalancer-1",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Started",
            "statusChangedAt": 1571071699.901
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachInstancesToLoadBalancer。

attach-load-balancer-tls-certificate

以下程式碼範例顯示如何使用 attach-load-balancer-tls-certificate。

AWS CLI

將 TLS 憑證連接至負載平衡器

下列attach-load-balancer-tls-certificate範例會將負載平衡器 TLS 憑證連接至Certificate2負載平衡器 LoadBalancer-1。

```
aws lightsail attach-load-balancer-tls-certificate \
    --certificate-name Certificate2 \
    --load-balancer-name LoadBalancer-1
```

輸出：

```
{
```

```
    "operations": [
        {
            "id": "cf1ad6e3-3cbb-4b8a-a7f2-3EXAMPLEa118",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571072255.416,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "Certificate2",
            "operationType": "AttachLoadBalancerTlsCertificate",
            "status": "Succeeded",
            "statusChangedAt": 1571072255.416
        },
        {
            "id": "dae1bcfb-d531-4c06-b4ea-bEXAMPLEc04e",
            "resourceName": "Certificate2",
            "resourceType": "LoadBalancerTlsCertificate",
            "createdAt": 1571072255.416,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "LoadBalancer-1",
            "operationType": "AttachLoadBalancerTlsCertificate",
            "status": "Succeeded",
            "statusChangedAt": 1571072255.416
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachLoadBalancerTlsCertificate。

attach-static-ip

以下程式碼範例顯示如何使用 attach-static-ip。

AWS CLI

將靜態 IP 連接至執行個體

下列attach-static-ip範例會將靜態 IP 連接至StaticIp-1執行個體 MEAN-1。

```
aws lightsail attach-static-ip \
    --static-ip-name StaticIp-1 \
    --instance-name MEAN-1
```

輸出：

```
{
    "operations": [
        {
            "id": "45e6fa13-4808-4b8d-9292-bEXAMPLE20b2",
            "resourceName": "StaticIp-1",
            "resourceType": "StaticIp",
            "createdAt": 1571072569.375,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "MEAN-1",
            "operationType": "AttachStaticIp",
            "status": "Succeeded",
            "statusChangedAt": 1571072569.375
        },
        {
            "id": "9ee09a17-863c-4e51-8a6d-3EXAMPLE5475",
            "resourceName": "MEAN-1",
            "resourceType": "Instance",
            "createdAt": 1571072569.376,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "StaticIp-1",
            "operationType": "AttachStaticIp",
            "status": "Succeeded",
            "statusChangedAt": 1571072569.376
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachStaticIp。

close-instance-public-ports

以下程式碼範例顯示如何使用 close-instance-public-ports。

AWS CLI

關閉執行個體的防火牆連接埠

下列close-instance-public-ports範例會關閉執行個體 22上的 TCP 連接埠MEAN-2。

```
aws lightsail close-instance-public-ports \
    --instance-name MEAN-2 \
    --port-info fromPort=22,protocol=TCP,toPort=22
```

輸出：

```
{
    "operation": {
        "id": "4f328636-1c96-4649-ae6d-1EXAMPLEf446",
        "resourceName": "MEAN-2",
        "resourceType": "Instance",
        "createdAt": 1571072845.737,
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "isTerminal": true,
        "operationDetails": "22/tcp",
        "operationType": "CloseInstancePublicPorts",
        "status": "Succeeded",
        "statusChangedAt": 1571072845.737
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CloseInstancePublicPorts。

copy-snapshot

以下程式碼範例顯示如何使用 copy-snapshot。

AWS CLI

範例 1：在相同區域內複製快照 AWS

下列copy-snapshot範例會將執行個體快照複製MEAN-1-1571075291為相同 AWS 區域MEAN-1-Copy內的執行個體快照us-west-2。

```
aws lightsail copy-snapshot \
    --source-snapshot-name MEAN-1-1571075291 \
    --target-snapshot-name MEAN-1-Copy \
    --source-region us-west-2
```

輸出：

```
{
    "operations": [
        {
            "id": "ced16fc1-f401-4556-8d82-1EXAMPLEb982",
            "resourceName": "MEAN-1-Copy",
            "resourceType": "InstanceSnapshot",
            "createdAt": 1571075581.498,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "us-west-2:MEAN-1-1571075291",
            "operationType": "CopySnapshot",
            "status": "Started",
            "statusChangedAt": 1571075581.498
        }
    ]
}
```

如需詳細資訊，請參閱 Lightsail 開發指南中的將快照從一個 AWS 區域複製到 中的另一個Amazon Lightsail區域。

範例 2：將快照從一個區域複製到另一個 AWS 區域

下列copy-snapshot範例會將執行個體快照MEAN-1-1571075291作為執行個體快照MEAN-1-1571075291-Copy從 AWS 區域複製到 us-west-2 us-east-1。

```
aws lightsail copy-snapshot \
```

```
    --source-snapshot-name MEAN-1-1571075291 \
    --target-snapshot-name MEAN-1-1571075291-Copy \
    --source-region us-west-2 \
    --region us-east-1
```

輸出：

```
{
    "operations": [
        {
            "id": "91116b79-119c-4451-b44a-dEXAMPLEd97b",
            "resourceName": "MEAN-1-1571075291-Copy",
            "resourceType": "InstanceSnapshot",
            "createdAt": 1571075695.069,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-east-1"
            },
            "isTerminal": false,
            "operationDetails": "us-west-2:MEAN-1-1571075291",
            "operationType": "CopySnapshot",
            "status": "Started",
            "statusChangedAt": 1571075695.069
        }
    ]
}
```

如需詳細資訊，請參閱Lightsail《開發指南》中的將快照從一個 AWS 區域複製到另一個區域 Amazon Lightsail。

範例 3：在相同區域內複製自動快照 AWS

下列copy-snapshot範例會在 AWS 區域 WordPress-1-10142019中將執行個體2019-10-14的自動快照複製WordPress-1為手動快照us-west-2。

```
aws lightsail copy-snapshot \
    --source-resource-name WordPress-1 \
    --restore-date 2019-10-14 \
    --target-snapshot-name WordPress-1-10142019 \
    --source-region us-west-2
```

輸出：

```
{
    "operations": [
        {
            "id": "be3e6754-cd1d-48e6-ad9f-2EXAMPLE1805",
            "resourceName": "WordPress-1-10142019",
            "resourceType": "InstanceSnapshot",
            "createdAt": 1571082412.311,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "us-west-2:WordPress-1",
            "operationType": "CopySnapshot",
            "status": "Started",
            "statusChangedAt": 1571082412.311
        }
    ]
}
```

如需詳細資訊，請參閱Lightsail《 開發指南》中的在 中保留執行個體或磁碟的自動快照Amazon Lightsail。

範例 4：將自動快照從一個區域複製到另一個 AWS 區域

下列copy-snapshot範例會將執行個體2019-10-14的自動快照WordPress-1作為手動快照WordPress-1-10142019從 AWS 區域複製到 us-west-2 us-east-1。

```
aws lightsail copy-snapshot \
    --source-resource-name WordPress-1 \
    --restore-date 2019-10-14 \
    --target-snapshot-name WordPress-1-10142019 \
    --source-region us-west-2 \
    --region us-east-1
```

輸出：

```
{
    "operations": [
        {
            "id": "dffa128b-0b07-476e-b390-bEXAMPLE3775",
            "resourceName": "WordPress-1-10142019",
```

```
            "resourceType": "InstanceSnapshot",
            "createdAt": 1571082493.422,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-east-1"
            },
            "isTerminal": false,
            "operationDetails": "us-west-2:WordPress-1",
            "operationType": "CopySnapshot",
            "status": "Started",
            "statusChangedAt": 1571082493.422
        }
    ]
}
```

如需詳細資訊，請參閱Lightsail《 開發指南》中的在 中保留執行個體或磁碟的自動快照Amazon Lightsail。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CopySnapshot。

create-disk-from-snapshot

以下程式碼範例顯示如何使用 create-disk-from-snapshot。

AWS CLI

從磁碟快照建立磁碟

下列create-disk-from-snapshot範例Disk-2會從指定的區塊儲存磁碟快照建立名為 的區塊儲存磁碟。磁碟是在指定的 AWS 區域和可用區域中建立，具有 32 GB 的儲存空間。

```
aws lightsail create-disk-from-snapshot \
    --disk-name Disk-2 \
    --disk-snapshot-name Disk-1-1566839161 \
    --availability-zone us-west-2a \
    --size-in-gb 32
```

輸出：

```
{
    "operations": [
        {
```

```
            "id": "d42b605d-5ef1-4b4a-8791-7a3e8b66b5e7",
            "resourceName": "Disk-2",
            "resourceType": "Disk",
            "createdAt": 1569624941.471,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateDiskFromSnapshot",
            "status": "Started",
            "statusChangedAt": 1569624941.791
        }
    ]
}
```

如需詳細資訊，請參閱《 Lightsail開發人員指南》中的從 中的快照建立區塊儲存磁碟Amazon Lightsail。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDiskFromSnapshot。

create-disk-snapshot

以下程式碼範例顯示如何使用 create-disk-snapshot。

AWS CLI

範例 1：建立磁碟的快照

下列create-disk-snapshot範例會建立名為指定區塊儲存磁碟DiskSnapshot-1的快照。

```
aws lightsail create-disk-snapshot \
    --disk-name Disk-1 \
    --disk-snapshot-name DiskSnapshot-1
```

輸出：

```
{
    "operations": [
        {
            "id": "fa74c6d2-03a3-4f42-a7c7-792f124d534b",
            "resourceName": "DiskSnapshot-1",
            "resourceType": "DiskSnapshot",
```

```
            "createdAt": 1569625129.739,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "Disk-1",
            "operationType": "CreateDiskSnapshot",
            "status": "Started",
            "statusChangedAt": 1569625129.739
        },
        {
            "id": "920a25df-185c-4528-87cd-7b85f5488c06",
            "resourceName": "Disk-1",
            "resourceType": "Disk",
            "createdAt": 1569625129.739,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "DiskSnapshot-1",
            "operationType": "CreateDiskSnapshot",
            "status": "Started",
            "statusChangedAt": 1569625129.739
        }
    ]
}
```

範例 2：建立執行個體系統磁碟的快照

下列create-disk-snapshot範例會建立指定執行個體系統磁碟的快照。

```
aws lightsail create-disk-snapshot \
    --instance-name WordPress-1 \
    --disk-snapshot-name SystemDiskSnapshot-1
```

輸出：

```
{
    "operations": [
        {
            "id": "f508cf1c-6597-42a6-a4c3-4aebd75af0d9",
```

```
            "resourceName": "SystemDiskSnapshot-1",
            "resourceType": "DiskSnapshot",
            "createdAt": 1569625294.685,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "WordPress-1",
            "operationType": "CreateDiskSnapshot",
            "status": "Started",
            "statusChangedAt": 1569625294.685
        },
        {
            "id": "0bb9f712-da3b-4d99-b508-3bf871d989e5",
            "resourceName": "WordPress-1",
            "resourceType": "Instance",
            "createdAt": 1569625294.685,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "SystemDiskSnapshot-1",
            "operationType": "CreateDiskSnapshot",
            "status": "Started",
            "statusChangedAt": 1569625294.685
        }
    ]
}
```

如需詳細資訊，請參閱 中的快照Amazon Lightsail和《 Lightsail開發人員指南》中的在 中建立執行個體根磁碟區的快照Amazon Lightsail。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDiskSnapshot。

`create-disk`

以下程式碼範例顯示如何使用 `create-disk`。

AWS CLI

建立區塊儲存磁碟

下列create-disk範例Disk-1會在指定的 AWS 區域和可用區域中建立具有 32 GB 儲存空間的區塊儲存磁碟。

```
aws lightsail create-disk \
    --disk-name Disk-1 \
    --availability-zone us-west-2a \
    --size-in-gb 32
```

輸出：

```
{
    "operations": [
        {
            "id": "1c85e2ec-86ba-4697-b936-77f4d3dc013a",
            "resourceName": "Disk-1",
            "resourceType": "Disk",
            "createdAt": 1569449220.36,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateDisk",
            "status": "Started",
            "statusChangedAt": 1569449220.588
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDisk。

create-domain-entry

以下程式碼範例顯示如何使用 create-domain-entry。

AWS CLI

建立網域項目 (DNS 記錄)

下列create-domain-entry範例會為指向執行個體 IP 地址的指定網域頂點建立 DNS 記錄 (A)。

注意： Lightsail的網域相關 API 操作僅適用於 us-east-1區域。如果您的 CLI 設定檔設定為使用不同的區域，您必須包含 --region us-east-1 參數，否則命令會失敗。

```
aws lightsail create-domain-entry \
    --region us-east-1 \
    --domain-name example.com \
    --domain-entry name=example.com,type=A,target=192.0.2.0
```

輸出：

```
{
    "operation": {
        "id": "5be4494d-56f4-41fc-8730-693dcd0ef9e2",
        "resourceName": "example.com",
        "resourceType": "Domain",
        "createdAt": 1569865296.519,
        "location": {
            "availabilityZone": "all",
            "regionName": "global"
        },
        "isTerminal": true,
        "operationType": "CreateDomainEntry",
        "status": "Succeeded",
        "statusChangedAt": 1569865296.519
    }
}
```

如需詳細資訊，請參閱《 開發人員指南》中的 DNS Amazon Lightsail 和建立 DNS 區域以管理網域的 DNS 記錄Amazon Lightsail。 Lightsail

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDomainEntry。

create-domain

以下程式碼範例顯示如何使用 create-domain。

AWS CLI

建立網域 (DNS 區域)

下列create-domain範例會為指定的網域建立 DNS 區域。

注意：Lightsail的網域相關 API 操作僅適用於 us-east-1區域。如果您的 CLI 設定檔設定為使用不同的區域，您必須包含 --region us-east-1 參數，否則命令會失敗。

```
aws lightsail create-domain \
    --region us-east-1 \
    --domain-name example.com
```

輸出：

```
{
    "operation": {
        "id": "64e522c8-9ae1-4c05-9b65-3f237324dc34",
        "resourceName": "example.com",
        "resourceType": "Domain",
        "createdAt": 1569864291.92,
        "location": {
            "availabilityZone": "all",
            "regionName": "global"
        },
        "isTerminal": true,
        "operationType": "CreateDomain",
        "status": "Succeeded",
        "statusChangedAt": 1569864292.109
    }
}
```

如需詳細資訊，請參閱《Lightsail開發人員指南》中的 DNS Amazon Lightsail 和建立 DNS 區域以管理網域的 DNS 記錄Amazon Lightsail。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateDomain。

create-instance-snapshot

以下程式碼範例顯示如何使用 create-instance-snapshot。

AWS CLI

建立執行個體的快照

下列create-instance-snapshot範例會從指定的執行個體建立快照。

```
aws lightsail create-instance-snapshot \
```

```
    --instance-name WordPress-1 \
    --instance-snapshot-name WordPress-Snapshot-1
```

輸出：

```
{
    "operations": [
        {
            "id": "4c3db559-9dd0-41e7-89c0-2cb88c19786f",
            "resourceName": "WordPress-Snapshot-1",
            "resourceType": "InstanceSnapshot",
            "createdAt": 1569866438.48,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "WordPress-1",
            "operationType": "CreateInstanceSnapshot",
            "status": "Started",
            "statusChangedAt": 1569866438.48
        },
        {
            "id": "c04fdc45-2981-488c-88b5-d6d2fd759a6a",
            "resourceName": "WordPress-1",
            "resourceType": "Instance",
            "createdAt": 1569866438.48,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "WordPress-Snapshot-1",
            "operationType": "CreateInstanceSnapshot",
            "status": "Started",
            "statusChangedAt": 1569866438.48
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateInstanceSnapshot。

create-instances-from-snapshot

以下程式碼範例顯示如何使用 create-instances-from-snapshot。

AWS CLI

從快照建立執行個體

下列create-instances-from-snapshot範例會使用 12 USD 的套件，在指定的 AWS 區域和可用區域中從指定的執行個體快照建立執行個體。

注意：您指定的套件在規格中必須等於或大於用於建立快照的原始來源執行個體套件。

```
aws lightsail create-instances-from-snapshot \
    --instance-snapshot-name WordPress-1-1569866208 \
    --instance-names WordPress-2 \
    --availability-zone us-west-2a \
    --bundle-id small_3_0
```

輸出：

```
{
    "operations": [
        {
            "id": "003f8271-b711-464d-b9b8-7f3806cb496e",
            "resourceName": "WordPress-2",
            "resourceType": "Instance",
            "createdAt": 1569865914.908,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateInstancesFromSnapshot",
            "status": "Started",
            "statusChangedAt": 1569865914.908
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateInstancesFromSnapshot。

create-instances

以下程式碼範例顯示如何使用 create-instances。

AWS CLI

範例 1：建立單一執行個體

下列create-instances範例會使用 WordPress 藍圖和 5.00 USD 套件，在指定的 AWS 區域和可用區域中建立執行個體。

```
aws lightsail create-instances \
    --instance-names Instance-1 \
    --availability-zone us-west-2a \
    --blueprint-id wordpress \
    --bundle-id nano_3_0
```

輸出：

```
{
    "operations": [
        {
            "id": "9a77158f-7be3-4d6d-8054-cf5ae2b720cc",
            "resourceName": "Instance-1",
            "resourceType": "Instance",
            "createdAt": 1569447986.061,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateInstance",
            "status": "Started",
            "statusChangedAt": 1569447986.061
        }
    ]
}
```

範例 2：一次建立多個執行個體

下列create-instances範例會使用 WordPress 藍圖和 5.00 USD 套件，在指定的 AWS 區域和可用區域中建立三個執行個體。

```
aws lightsail create-instances \
    --instance-names {"Instance1","Instance2","Instance3"} \
    --availability-zone us-west-2a \
    --blueprint-id wordpress \
    --bundle-id nano_3_0
```

輸出：

```
{
    "operations": [
        {
            "id": "5492f015-9d2e-48c6-8eea-b516840e6903",
            "resourceName": "Instance1",
            "resourceType": "Instance",
            "createdAt": 1569448780.054,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateInstance",
            "status": "Started",
            "statusChangedAt": 1569448780.054
        },
        {
            "id": "c58b5f46-2676-44c8-b95c-3ad375898515",
            "resourceName": "Instance2",
            "resourceType": "Instance",
            "createdAt": 1569448780.054,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateInstance",
            "status": "Started",
            "statusChangedAt": 1569448780.054
        },
        {
            "id": "a5ad8006-9bee-4499-9eb7-75e42e6f5882",
            "resourceName": "Instance3",
            "resourceType": "Instance",
            "createdAt": 1569448780.054,
```

```
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateInstance",
            "status": "Started",
            "statusChangedAt": 1569448780.054
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateInstances。

create-key-pair

以下程式碼範例顯示如何使用 create-key-pair。

AWS CLI

建立一組金鑰對

下列create-key-pair範例會建立金鑰對，您可用來驗證和連線至執行個體。

```
aws lightsail create-key-pair \
    --key-pair-name MyPersonalKeyPair
```

輸出提供私有金鑰 base64 值，可用來向使用已建立金鑰對的執行個體進行身分驗證。注意：複製私有金鑰 base64 值並貼到安全的位置，因為您稍後無法擷取。

```
{
    "keyPair": {
        "name": "MyPersonalKeyPair",
        "arn": "arn:aws:lightsail:us-west-2:111122223333:KeyPair/55025c71-198f-403b-
b42f-a69433e724fb",
        "supportCode": "621291663362/MyPersonalKeyPair",
        "createdAt": 1569866556.567,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
```

```
        "resourceType": "KeyPair"
    },
    "publicKeyBase64": "ssh-rsa ssh-rsa
 AAAAB3NzaC1yc2EAAAADAQABAAABAQCV0xUEwx96amPERH7K1bVT1tTFl9OmNk6o7m5YVHk9xlOdMbDRbFvhtXvw4jz
+BHUgedGUXno6uF7agqxZNO1kPLJBIVTW26SSYBJ0tE
+y804UyVsjrbUqCaMXDhmfXpWulMPwuXhwcKh7e8hwoTfkiX0E6Ql
+KqF/MiA3w6DCjEqvvdIO7SiEZJFsuGNfYDDN3w60Rel5MUhmn3OJdn4y/
A7NWb3IxL4pPfVE4rgFRKU8n1jp9kwRnlVMVBOWuGXk6n+H6M2f1 ",
    "privateKeyBase64": "-----BEGIN RSA PRIVATE KEY-----
EXAMPLETCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
\nVVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6\nb24xFDASBgNVBAsTC0lBTSBD
\nBgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
\nMTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
\nVQQHEwdTZWF0dGxlMQ8wDQEXAMPLEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
\nb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
\nYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMEXAMPLE4GmWIWJ
\n21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
\nrDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
\nIbb3OhjZnzcvQAaREXAMPLEMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4\nnUhVVxYUntneD9+h8Mg9q6q
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
\nFFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OEXAMPLELvjx79LjSTb
\nNYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=\n-----END RSA PRIVATE KEY-----",
    "operation": {
        "id": "67f984db-9994-45fe-ad38-59bafcaf82ef",
        "resourceName": "MyPersonalKeyPair",
        "resourceType": "KeyPair",
        "createdAt": 1569866556.567,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "isTerminal": true,
        "operationType": "CreateKeyPair",
        "status": "Succeeded",
        "statusChangedAt": 1569866556.704
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateKeyPair。

create-load-balancer-tls-certificate

以下程式碼範例顯示如何使用 create-load-balancer-tls-certificate。

AWS CLI

為負載平衡器建立 TLS 憑證

下列create-load-balancer-tls-certificate範例會建立連接到指定負載平衡器的 TLS 憑證。建立的憑證會套用至指定的網域。注意：負載平衡器只能建立兩個憑證。

```
aws lightsail create-load-balancer-tls-certificate \
    --certificate-alternative-names abc.example.com \
    --certificate-domain-name example.com \
    --certificate-name MySecondCertificate \
    --load-balancer-name MyFirstLoadBalancer
```

輸出：

```
{
    "operations": [
        {
            "id": "be663aed-cb46-41e2-9b23-e2f747245bd4",
            "resourceName": "MySecondCertificate",
            "resourceType": "LoadBalancerTlsCertificate",
            "createdAt": 1569867364.971,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "MyFirstLoadBalancer",
            "operationType": "CreateLoadBalancerTlsCertificate",
            "status": "Succeeded",
            "statusChangedAt": 1569867365.219
        },
        {
            "id": "f3dfa930-969e-41cc-ac7d-337178716f6d",
            "resourceName": "MyFirstLoadBalancer",
            "resourceType": "LoadBalancer",
            "createdAt": 1569867364.971,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "MySecondCertificate",
```

```
            "operationType": "CreateLoadBalancerTlsCertificate",
            "status": "Succeeded",
            "statusChangedAt": 1569867365.219
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLoadBalancerTlsCertificate。

create-load-balancer

以下程式碼範例顯示如何使用 create-load-balancer。

AWS CLI

建立負載平衡器

下列create-load-balancer範例會建立具有 TLS 憑證的負載平衡器。TLS 憑證適用於指定的網域，並將流量路由到連接埠 80 上的執行個體。

```
aws lightsail create-load-balancer \
    --certificate-alternative-names www.example.com test.example.com \
    --certificate-domain-name example.com \
    --certificate-name Certificate-1 \
    --instance-port 80 \
    --load-balancer-name LoadBalancer-1
```

輸出：

```
{
    "operations": [
        {
            "id": "cc7b920a-83d8-4762-a74e-9174fe1540be",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1569867169.406,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateLoadBalancer",
```

```
            "status": "Started",
            "statusChangedAt": 1569867169.406
        },
        {
            "id": "658ed43b-f729-42f3-a8e4-3f8024d3c98d",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancerTlsCertificate",
            "createdAt": 1569867170.193,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "LoadBalancer-1",
            "operationType": "CreateLoadBalancerTlsCertificate",
            "status": "Succeeded",
            "statusChangedAt": 1569867170.54
        },
        {
            "id": "4757a342-5181-4870-b1e0-227eebc35ab5",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1569867170.193,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "Certificate-1",
            "operationType": "CreateLoadBalancerTlsCertificate",
            "status": "Succeeded",
            "statusChangedAt": 1569867170.54
        }
    ]
}
```

如需詳細資訊，請參閱《 Lightsail開發人員指南》中的Lightsail負載平衡器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLoadBalancer。

create-relational-database-from-snapshot

以下程式碼範例顯示如何使用 create-relational-database-from-snapshot。

AWS CLI

從快照建立受管資料庫

下列create-relational-database-from-snapshot範例使用 15 USD 的標準資料庫套件，從指定 AWS 區域和可用區域中的指定快照建立受管資料庫。注意：您指定的套件在規格中必須等於或大於用於建立快照的原始來源資料庫套件。

```
aws lightsail create-relational-database-from-snapshot \
    --relational-database-snapshot-name Database-Oregon-1-1566839359 \
    --relational-database-name Database-1 \
    --availability-zone us-west-2a \
    --relational-database-bundle-id micro_1_0 \
    --no-publicly-accessible
```

輸出：

```
{
    "operations": [
        {
            "id": "ad6d9193-9d5c-4ea1-97ae-8fe6de600b4c",
            "resourceName": "Database-1",
            "resourceType": "RelationalDatabase",
            "createdAt": 1569867916.938,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "CreateRelationalDatabaseFromSnapshot",
            "status": "Started",
            "statusChangedAt": 1569867918.643
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRelationalDatabaseFromSnapshot。

create-relational-database-snapshot

以下程式碼範例顯示如何使用 create-relational-database-snapshot。

AWS CLI

建立受管資料庫的快照

下列create-relational-database-snapshot範例會建立指定受管資料庫的快照。

```
aws lightsail create-relational-database-snapshot \
    --relational-database-name Database1 \
    --relational-database-snapshot-name RelationalDatabaseSnapshot1
```

輸出：

```
{
    "operations": [
        {
            "id": "853667fb-ea91-4c02-8d20-8fc5fd43b9eb",
            "resourceName": "RelationalDatabaseSnapshot1",
            "resourceType": "RelationalDatabaseSnapshot",
            "createdAt": 1569868074.645,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "Database1",
            "operationType": "CreateRelationalDatabaseSnapshot",
            "status": "Started",
            "statusChangedAt": 1569868074.645
        },
        {
            "id": "fbafa521-3cac-4be8-9773-1c143780b239",
            "resourceName": "Database1",
            "resourceType": "RelationalDatabase",
            "createdAt": 1569868074.645,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "RelationalDatabaseSnapshot1",
            "operationType": "CreateRelationalDatabaseSnapshot",
            "status": "Started",
            "statusChangedAt": 1569868074.645
```

```
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRelationalDatabaseSnapshot。

create-relational-database

以下程式碼範例顯示如何使用 create-relational-database。

AWS CLI

建立受管資料庫

下列create-relational-database範例使用 MySQL 5.6 資料庫引擎 (mysql_5_6) 和 15 USD 標準資料庫套件 (micro_1_0)，在指定的 AWS 區域和可用區域中建立受管資料庫。受管資料庫會預先填入主要使用者名稱，且無法公開存取。

```
aws lightsail create-relational-database \
    --relational-database-name Database-1 \
    --availability-zone us-west-2a \
    --relational-database-blueprint-id mysql_5_6 \
    --relational-database-bundle-id micro_1_0 \
    --master-database-name dbmaster \
    --master-username user \
    --no-publicly-accessible
```

輸出：

```
{
    "operations": [
        {
            "id": "b52bedee-73ed-4798-8d2a-9c12df89adcd",
            "resourceName": "Database-1",
            "resourceType": "RelationalDatabase",
            "createdAt": 1569450017.244,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
```

```
            "operationType": "CreateRelationalDatabase",
            "status": "Started",
            "statusChangedAt": 1569450018.637
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRelationalDatabase。

delete-auto-snapshot

以下程式碼範例顯示如何使用 delete-auto-snapshot。

AWS CLI

刪除自動快照

下列delete-auto-snapshot範例會刪除執行個體 2019-10-10的自動快照WordPress-1。

```
aws lightsail delete-auto-snapshot \
    --resource-name WordPress-1 \
    --date 2019-10-10
```

輸出：

```
{
    "operations": [
        {
            "id": "31c36e09-3d52-46d5-b6d8-7EXAMPLE534a",
            "resourceName": "WordPress-1",
            "resourceType": "Instance",
            "createdAt": 1571088141.501,
            "location": {
                "availabilityZone": "us-west-2",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "DeleteAutoSnapshot-2019-10-10",
            "operationType": "DeleteAutoSnapshot",
            "status": "Succeeded"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱Lightsail《 開發指南》中的在 中刪除執行個體或磁碟的自動快照Amazon Lightsail。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAutoSnapshot。

delete-disk-snapshot

以下程式碼範例顯示如何使用 delete-disk-snapshot。

AWS CLI

刪除區塊儲存磁碟的快照

下列delete-disk-snapshot範例會刪除區塊儲存磁碟的指定快照

```
aws lightsail delete-disk-snapshot \
    --disk-snapshot-name DiskSnapshot-1
```

輸出：

```
{
    "operations": [
        {
            "id": "d1e5766d-b81e-4595-ad5d-02afbccfcd5d",
            "resourceName": "DiskSnapshot-1",
            "resourceType": "DiskSnapshot",
            "createdAt": 1569873552.79,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "DeleteDiskSnapshot",
            "status": "Succeeded",
            "statusChangedAt": 1569873552.79
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDiskSnapshot。

delete-disk

以下程式碼範例顯示如何使用 delete-disk。

AWS CLI

刪除區塊儲存磁碟

下列delete-disk範例會刪除指定的區塊儲存磁碟。

```
aws lightsail delete-disk \
    --disk-name Disk-1
```

輸出：

```
{
    "operations": [
        {
            "id": "6378c70f-4d75-4f7a-ab66-730fca0bb2fc",
            "resourceName": "Disk-1",
            "resourceType": "Disk",
            "createdAt": 1569872887.864,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "DeleteDisk",
            "status": "Succeeded",
            "statusChangedAt": 1569872887.864
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDisk。

delete-domain-entry

以下程式碼範例顯示如何使用 delete-domain-entry。

AWS CLI

刪除網域項目 (DNS 記錄)

下列delete-domain-entry範例會從現有網域刪除指定的網域項目。

注意：Lightsail的網域相關 API 操作僅適用於 us-east-1區域。如果您的 CLI 設定檔設定為使用不同的區域，您必須包含 --region us-east-1 參數，否則命令會失敗。

```
aws lightsail delete-domain-entry \
    --region us-east-1 \
    --domain-name example.com \
    --domain-entry name=123.example.com,target=192.0.2.0,type=A
```

輸出：

```
{
    "operation": {
        "id": "06eacd01-d785-420e-8daa-823150c7dca1",
        "resourceName": "example.com ",
        "resourceType": "Domain",
        "createdAt": 1569874157.005,
        "location": {
            "availabilityZone": "all",
            "regionName": "global"
        },
        "isTerminal": true,
        "operationType": "DeleteDomainEntry",
        "status": "Succeeded",
        "statusChangedAt": 1569874157.005
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDomainEntry。

delete-domain

以下程式碼範例顯示如何使用 delete-domain。

AWS CLI

刪除網域 (DNS 區域)

下列delete-domain範例會刪除指定的網域和網域中的所有項目 (DNS 記錄)。

注意：Lightsail的網域相關 API 操作僅適用於 us-east-1區域。如果您的 CLI 設定檔設定為使用不同的區域，您必須包含 --region us-east-1 參數，否則命令會失敗。

```
aws lightsail delete-domain \
    --region us-east-1 \
    --domain-name example.com
```

輸出：

```
{
    "operation": {
        "id": "fcef5265-5af1-4a46-a3d7-90b5e18b9b32",
        "resourceName": "example.com",
        "resourceType": "Domain",
        "createdAt": 1569873788.13,
        "location": {
            "availabilityZone": "all",
            "regionName": "global"
        },
        "isTerminal": true,
        "operationType": "DeleteDomain",
        "status": "Succeeded",
        "statusChangedAt": 1569873788.13
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDomain。

delete-instance-snapshot

以下程式碼範例顯示如何使用 delete-instance-snapshot。

AWS CLI

標題

下列delete-instance-snapshot範例會刪除執行個體的指定快照。

```
aws lightsail delete-instance-snapshot \
    --instance-snapshot-name WordPress-1-Snapshot-1
```

輸出：

```
{
    "operations": [
        {
            "id": "14dad182-976a-46c6-bfd4-9480482bf0ea",
            "resourceName": "WordPress-1-Snapshot-1",
            "resourceType": "InstanceSnapshot",
            "createdAt": 1569874524.562,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "DeleteInstanceSnapshot",
            "status": "Succeeded",
            "statusChangedAt": 1569874524.562
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteInstanceSnapshot。

delete-instance

以下程式碼範例顯示如何使用 delete-instance。

AWS CLI

刪除執行個體

下列delete-instance範例會刪除指定的執行個體。

```
aws lightsail delete-instance \
    --instance-name WordPress-1
```

輸出：

```
{
    "operations": [
        {
            "id": "d77345a3-8f80-4d2e-b47d-aaa622718df2",
            "resourceName": "Disk-1",
            "resourceType": "Disk",
```

```
            "createdAt": 1569874357.469,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "WordPress-1",
            "operationType": "DetachDisk",
            "status": "Started",
            "statusChangedAt": 1569874357.469
        },
        {
            "id": "708fa606-2bfd-4e48-a2c1-0b856585b5b1",
            "resourceName": "WordPress-1",
            "resourceType": "Instance",
            "createdAt": 1569874357.465,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "Disk-1",
            "operationType": "DetachDisk",
            "status": "Started",
            "statusChangedAt": 1569874357.465
        },
        {
            "id": "3187e823-8acb-405d-b098-fad5ceb17bec",
            "resourceName": "WordPress-1",
            "resourceType": "Instance",
            "createdAt": 1569874357.829,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "DeleteInstance",
            "status": "Succeeded",
            "statusChangedAt": 1569874357.829
        }
    ]
}
```


- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteInstance。

delete-key-pair

以下程式碼範例顯示如何使用 delete-key-pair。

AWS CLI

刪除金鑰對

下列delete-key-pair範例會刪除指定的金鑰對。

```
aws lightsail delete-key-pair \
    --key-pair-name MyPersonalKeyPair
```

輸出：

```
{
    "operation": {
        "id": "81621463-df38-4810-b866-6e801a15abbf",
        "resourceName": "MyPersonalKeyPair",
        "resourceType": "KeyPair",
        "createdAt": 1569874626.466,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "isTerminal": true,
        "operationType": "DeleteKeyPair",
        "status": "Succeeded",
        "statusChangedAt": 1569874626.685
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteKeyPair。

delete-known-host-keys

以下程式碼範例顯示如何使用 delete-known-host-keys。

AWS CLI

從執行個體刪除已知主機金鑰

下列delete-known-host-keys範例會從指定的執行個體刪除已知的主機金鑰。

```
aws lightsail delete-known-host-keys \
    --instance-name Instance-1
```

輸出：

```
{
    "operations": [
        {
            "id": "c61afe9c-45a4-41e6-a97e-d212364da3f5",
            "resourceName": "Instance-1",
            "resourceType": "Instance",
            "createdAt": 1569874760.201,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "DeleteKnownHostKeys",
            "status": "Succeeded",
            "statusChangedAt": 1569874760.201
        }
    ]
}
```

如需詳細資訊，請參閱Lightsail《開發指南》中的對瀏覽器Amazon Lightsail型 SSH 或 RDP 用戶端的連線問題進行故障診斷。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteKnownHostKeys。

delete-load-balancer-tls-certificate

以下程式碼範例顯示如何使用 delete-load-balancer-tls-certificate。

AWS CLI

刪除負載平衡器的 TLS 憑證

下列delete-load-balancer-tls-certificate範例會從指定的負載平衡器刪除指定 TLS 憑證。

```
aws lightsail delete-load-balancer-tls-certificate \
    --load-balancer-name MyFirstLoadBalancer \
```

```
    --certificate-name MyFirstCertificate
```

輸出：

```
{
    "operations": [
        {
            "id": "50bec274-e45e-4caa-8a69-b763ef636583",
            "resourceName": "MyFirstCertificate",
            "resourceType": "LoadBalancerTlsCertificate",
            "createdAt": 1569874989.48,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "DeleteLoadBalancerTlsCertificate",
            "status": "Started",
            "statusChangedAt": 1569874989.48
        },
        {
            "id": "78c58cdc-a59a-4b27-8213-500638634a8f",
            "resourceName": "MyFirstLoadBalancer",
            "resourceType": "LoadBalancer",
            "createdAt": 1569874989.48,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "DeleteLoadBalancerTlsCertificate",
            "status": "Started",
            "statusChangedAt": 1569874989.48
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLoadBalancerTlsCertificate。

delete-load-balancer

以下程式碼範例顯示如何使用 delete-load-balancer。

AWS CLI

刪除負載平衡器

下列delete-load-balancer範例會刪除指定的負載平衡器和任何相關聯的 TLS 憑證。

```
aws lightsail delete-load-balancer \
    --load-balancer-name MyFirstLoadBalancer
```

輸出：

```
{
    "operations": [
        {
            "id": "a8c968c7-72a3-4680-a714-af8f03eea535",
            "resourceName": "MyFirstLoadBalancer",
            "resourceType": "LoadBalancer",
            "createdAt": 1569875092.125,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "DeleteLoadBalancer",
            "status": "Succeeded",
            "statusChangedAt": 1569875092.125
        },
        {
            "id": "f91a29fc-8ce3-4e69-a227-ea70ca890bf5",
            "resourceName": "MySecondCertificate",
            "resourceType": "LoadBalancerTlsCertificate",
            "createdAt": 1569875091.938,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "DeleteLoadBalancerTlsCertificate",
            "status": "Started",
            "statusChangedAt": 1569875091.938
        },
        {
            "id": "cf64c060-154b-4eb4-ba57-84e2e41563d6",
```

```
            "resourceName": "MyFirstLoadBalancer",
            "resourceType": "LoadBalancer",
            "createdAt": 1569875091.94,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "DeleteLoadBalancerTlsCertificate",
            "status": "Started",
            "statusChangedAt": 1569875091.94
        }
    ]
}
```

如需詳細資訊，請參閱 指南中的標題。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLoadBalancer。

delete-relational-database-snapshot

以下程式碼範例顯示如何使用 delete-relational-database-snapshot。

AWS CLI

刪除受管資料庫的快照

下列delete-relational-database-snapshot範例會刪除受管資料庫的指定快照。

```
aws lightsail delete-relational-database-snapshot \
    --relational-database-snapshot-name Database-Oregon-1-1566839359
```

輸出：

```
{
    "operations": [
        {
            "id": "b99acae8-735b-4823-922f-30af580e3729",
            "resourceName": "Database-Oregon-1-1566839359",
            "resourceType": "RelationalDatabaseSnapshot",
            "createdAt": 1569875293.58,
            "location": {
                "availabilityZone": "all",
```

```
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "DeleteRelationalDatabaseSnapshot",
            "status": "Succeeded",
            "statusChangedAt": 1569875293.58
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRelationalDatabaseSnapshot。

delete-relational-database

以下程式碼範例顯示如何使用 delete-relational-database。

AWS CLI

刪除受管資料庫

下列delete-relational-database範例會刪除指定的受管資料庫。

```
aws lightsail delete-relational-database \
    --relational-database-name Database-1
```

輸出：

```
{
    "operations": [
        {
            "id": "3b0c41c1-053d-46f0-92a3-14f76141dc86",
            "resourceName": "Database-1",
            "resourceType": "RelationalDatabase",
            "createdAt": 1569875210.999,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "DeleteRelationalDatabase",
            "status": "Started",
            "statusChangedAt": 1569875210.999
```

```
        },
        {
            "id": "01ddeae8-a87a-4a4b-a1f3-092c71bf9180",
            "resourceName": "Database-1",
            "resourceType": "RelationalDatabase",
            "createdAt": 1569875211.029,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "Database-1-FinalSnapshot-1569875210793",
            "operationType": "CreateRelationalDatabaseSnapshot",
            "status": "Started",
            "statusChangedAt": 1569875211.029
        },
        {
            "id": "74d73681-30e8-4532-974e-1f23cd3f9f73",
            "resourceName": "Database-1-FinalSnapshot-1569875210793",
            "resourceType": "RelationalDatabaseSnapshot",
            "createdAt": 1569875211.029,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "Database-1",
            "operationType": "CreateRelationalDatabaseSnapshot",
            "status": "Started",
            "statusChangedAt": 1569875211.029
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRelationalDatabase。

detach-static-ip

以下程式碼範例顯示如何使用 detach-static-ip。

AWS CLI

從執行個體分離靜態 IP

下列detach-static-ip範例會從StaticIp-1任何連接的執行個體分離靜態 IP。

```
aws lightsail detach-static-ip \
    --static-ip-name StaticIp-1
```

輸出：

```
{
    "operations": [
        {
            "id": "2a43d8a3-9f2d-4fe7-bdd0-eEXAMPLE3cf3",
            "resourceName": "StaticIp-1",
            "resourceType": "StaticIp",
            "createdAt": 1571088261.999,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "MEAN-1",
            "operationType": "DetachStaticIp",
            "status": "Succeeded",
            "statusChangedAt": 1571088261.999
        },
        {
            "id": "41a7d40c-74e8-4d2e-a837-cEXAMPLEf747",
            "resourceName": "MEAN-1",
            "resourceType": "Instance",
            "createdAt": 1571088262.022,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "StaticIp-1",
            "operationType": "DetachStaticIp",
            "status": "Succeeded",
            "statusChangedAt": 1571088262.022
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetachStaticIp。

get-active-names

以下程式碼範例顯示如何使用 get-active-names。

AWS CLI

取得作用中的資源名稱

下列get-active-names範例會傳回所設定 AWS 區域中的作用中資源名稱。

```
aws lightsail get-active-names
```

輸出：

```
{
    "activeNames": [
        "WordPress-1",
        "StaticIp-1",
        "MEAN-1",
        "Plesk_Hosting_Stack_on_Ubuntu-1"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetActiveNames。

get-auto-snapshots

以下程式碼範例顯示如何使用 get-auto-snapshots。

AWS CLI

取得執行個體可用的自動快照

下列get-auto-snapshots範例會傳回執行個體 可用的自動快照WordPress-1。

```
aws lightsail get-auto-snapshots \
    --resource-name WordPress-1
```

輸出：

```
{
```

```
    "resourceName": "WordPress-1",
    "resourceType": "Instance",
    "autoSnapshots": [
        {
            "date": "2019-10-14",
            "createdAt": 1571033872.0,
            "status": "Success",
            "fromAttachedDisks": []
        },
        {
            "date": "2019-10-13",
            "createdAt": 1570947473.0,
            "status": "Success",
            "fromAttachedDisks": []
        },
        {
            "date": "2019-10-12",
            "createdAt": 1570861072.0,
            "status": "Success",
            "fromAttachedDisks": []
        },
        {
            "date": "2019-10-11",
            "createdAt": 1570774672.0,
            "status": "Success",
            "fromAttachedDisks": []
        }
    ]
}
```

如需詳細資訊，請參閱Lightsail《 開發指南》中的在 中保留執行個體或磁碟的自動快照Amazon Lightsail。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAutoSnapshots。

`get-blueprints`

以下程式碼範例顯示如何使用 `get-blueprints`。

AWS CLI

取得新執行個體的藍圖

下列get-blueprints範例顯示所有可用藍圖的詳細資訊，可用於在 中建立新執行個體Amazon Lightsail。

```
aws lightsail get-blueprints
```

輸出：

```
{
    "blueprints": [
        {
            "blueprintId": "wordpress",
            "name": "WordPress",
            "group": "wordpress",
            "type": "app",
            "description": "Bitnami, the leaders in application packaging, and
 Automattic, the experts behind WordPress, have teamed up to offer this official
 WordPress image. This image is a pre-configured, ready-to-run image for running
 WordPress on Amazon Lightsail. WordPress is the world's most popular content
 management platform. Whether it's for an enterprise or small business website, or
 a personal or corporate blog, content authors can easily create content using its
 new Gutenberg editor, and developers can extend the base platform with additional
 features. Popular plugins like Jetpack, Akismet, All in One SEO Pack, WP Mail,
 Google Analytics for WordPress, and Amazon Polly are all pre-installed in this
 image. Let's Encrypt SSL certificates are supported through an auto-configuration
 script.",
            "isActive": true,
            "minPower": 0,
            "version": "6.5.3-0",
            "versionCode": "1",
            "productUrl": "https://aws.amazon.com/marketplace/pp/B00NN8Y43U",
            "licenseUrl": "https://aws.amazon.com/marketplace/pp/B00NN8Y43U#pdp-
usage",
            "platform": "LINUX_UNIX"
        },
        {
            "blueprintId": "lamp_8_bitnami",
            "name": "LAMP (PHP 8)",
            "group": "lamp_8",
            "type": "app",
            "description": "LAMP with PHP 8.X packaged by Bitnami enables you
 to quickly start building your websites and applications by providing a coding
 framework. As a developer, it provides standalone project directories to store your
 applications. This blueprint is configured for production environments. It includes
```

```
 SSL auto-configuration with Let's Encrypt certificates, and the latest releases of
 PHP, Apache, and MariaDB on Linux. This application also includes phpMyAdmin, PHP
 main modules and Composer.",
            "isActive": true,
            "minPower": 0,
            "version": "8.2.18-4",
            "versionCode": "1",
            "productUrl": "https://aws.amazon.com/marketplace/pp/
prodview-6g3gzfcih6dvu",
            "licenseUrl": "https://aws.amazon.com/marketplace/pp/
prodview-6g3gzfcih6dvu#pdp-usage",
            "platform": "LINUX_UNIX"
        },
        {
            "blueprintId": "nodejs",
            "name": "Node.js",
            "group": "node",
            "type": "app",
            "description": "Node.js packaged by Bitnami is a pre-configured, ready
 to run image for Node.js on Amazon EC2. It includes the latest version of Node.js,
 Apache, Python and Redis. The image supports multiple Node.js applications, each
 with its own virtual host and project directory. It is configured for production
 use and is secure by default, as all ports except HTTP, HTTPS and SSH ports are
 closed. Let's Encrypt SSL certificates are supported through an auto-configuration
 script. Developers benefit from instant access to a secure, update and consistent
 Node.js environment without having to manually install and configure multiple
 components and libraries.",
            "isActive": true,
            "minPower": 0,
            "version": "18.20.2-0",
            "versionCode": "1",
            "productUrl": "https://aws.amazon.com/marketplace/pp/B00NNZUAKO",
            "licenseUrl": "https://aws.amazon.com/marketplace/pp/B00NNZUAKO#pdp-
usage",
            "platform": "LINUX_UNIX"
        },
        ...
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBlueprints。

get-bundles

以下程式碼範例顯示如何使用 get-bundles。

AWS CLI

取得新執行個體的套件

下列get-bundles範例顯示可用於在 中建立新執行個體的所有可用套件的詳細資訊Amazon Lightsail。

```
aws lightsail get-bundles
```

輸出：

```
{
    "bundles": [
        {
            "price": 5.0,
            "cpuCount": 2,
            "diskSizeInGb": 20,
            "bundleId": "nano_3_0",
            "instanceType": "nano",
            "isActive": true,
            "name": "Nano",
            "power": 298,
            "ramSizeInGb": 0.5,
            "transferPerMonthInGb": 1024,
            "supportedPlatforms": [
                "LINUX_UNIX"
            ]
        },
        {
            "price": 7.0,
            "cpuCount": 2,
            "diskSizeInGb": 40,
            "bundleId": "micro_3_0",
            "instanceType": "micro",
            "isActive": true,
            "name": "Micro",
            "power": 500,
            "ramSizeInGb": 1.0,
            "transferPerMonthInGb": 2048,
```

```
            "supportedPlatforms": [
                "LINUX_UNIX"
            ]
        },
        {
            "price": 12.0,
            "cpuCount": 2,
            "diskSizeInGb": 60,
            "bundleId": "small_3_0",
            "instanceType": "small",
            "isActive": true,
            "name": "Small",
            "power": 1000,
            "ramSizeInGb": 2.0,
            "transferPerMonthInGb": 3072,
            "supportedPlatforms": [
                "LINUX_UNIX"
            ]
        },
        ...
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBundles。

get-cloud-formation-stack-records

以下程式碼範例顯示如何使用 get-cloud-formation-stack-records。

AWS CLI

取得 CloudFormation 堆疊記錄及其相關聯的堆疊

下列get-cloud-formation-stack-records範例顯示 CloudFormation 堆疊記錄及其相關聯堆疊的詳細資訊，這些堆疊用於從匯出的Amazon Lightsail快照建立 Amazon EC2 資源。

```
aws lightsail get-cloud-formation-stack-records
```

輸出：

```
{
```

```
    "cloudFormationStackRecords": [
        {
            "name": "CloudFormationStackRecord-588a4243-
e2d1-490d-8200-3a7513ecebdf",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:CloudFormationStackRecord/28d646ab-27bc-48d9-a422-1EXAMPLE6d37",
            "createdAt": 1565301666.586,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "CloudFormationStackRecord",
            "state": "Succeeded",
            "sourceInfo": [
                {
                    "resourceType": "ExportSnapshotRecord",
                    "name": "ExportSnapshotRecord-
e02f23d7-0453-4aa9-9c95-91aa01a141dd",
                    "arn": "arn:aws:lightsail:us-
west-2:111122223333:ExportSnapshotRecord/f12b8792-f3ea-4d6f-b547-2EXAMPLE8796"
                }
            ],
            "destinationInfo": {
                "id": "arn:aws:cloudformation:us-west-2:111122223333:stack/
Lightsail-Stack-588a4243-e2d1-490d-8200-3EXAMPLEebdf/063203b0-
ba28-11e9-838b-0EXAMPLE8b00",
                "service": "Aws::CloudFormation::Stack"
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCloudFormationStackRecords。

get-disk-snapshot

以下程式碼範例顯示如何使用 get-disk-snapshot。

AWS CLI

取得磁碟快照的相關資訊

下列get-disk-snapshot範例顯示磁碟快照 的詳細資訊Disk-1-1566839161。

```
aws lightsail get-disk-snapshot \
    --disk-snapshot-name Disk-1-1566839161
```

輸出：

```
{
    "diskSnapshot": {
        "name": "Disk-1-1566839161",
        "arn": "arn:aws:lightsail:us-west-2:111122223333:DiskSnapshot/
e2d0fa53-8ee0-41a0-8e56-0EXAMPLE1051",
        "supportCode": "6EXAMPLE3362/snap-0EXAMPLE06100d09",
        "createdAt": 1566839163.749,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "resourceType": "DiskSnapshot",
        "tags": [],
        "sizeInGb": 8,
        "state": "completed",
        "progress": "100%",
        "fromDiskName": "Disk-1",
        "fromDiskArn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
        "isFromAutoSnapshot": false
    }
}
```

如需詳細資訊，請參閱 指南中的標題。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDiskSnapshot。

get-disk-snapshots

以下程式碼範例顯示如何使用 get-disk-snapshots。

AWS CLI

取得所有磁碟快照的相關資訊

下列get-disk-snapshots範例顯示已設定 AWS 區域中所有磁碟快照的詳細資訊。

```
aws lightsail get-disk-snapshots
```

輸出：

```
{
    "diskSnapshots": [
        {
            "name": "Disk-2-1571090588",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:DiskSnapshot/32e889a9-38d4-4687-9f21-eEXAMPLE7839",
            "supportCode": "6EXAMPLE3362/snap-0EXAMPLE1ca192a4",
            "createdAt": 1571090591.226,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "DiskSnapshot",
            "tags": [],
            "sizeInGb": 8,
            "state": "completed",
            "progress": "100%",
            "fromDiskName": "Disk-2",
            "fromDiskArn": "arn:aws:lightsail:us-
west-2:111122223333:Disk/6a343ff8-6341-422d-86e2-bEXAMPLE16c2",
            "isFromAutoSnapshot": false
        },
        {
            "name": "Disk-1-1566839161",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:DiskSnapshot/
e2d0fa53-8ee0-41a0-8e56-0EXAMPLE1051",
            "supportCode": "6EXAMPLE3362/snap-0EXAMPLEe06100d09",
            "createdAt": 1566839163.749,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "DiskSnapshot",
            "tags": [],
            "sizeInGb": 8,
            "state": "completed",
            "progress": "100%",
            "fromDiskName": "Disk-1",
```

```
            "fromDiskArn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
            "isFromAutoSnapshot": false
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDiskSnapshots。

get-disk

以下程式碼範例顯示如何使用 get-disk。

AWS CLI

取得區塊儲存磁碟的相關資訊

下列get-disk範例顯示磁碟 的詳細資訊Disk-1。

```
aws lightsail get-disk \
    --disk-name Disk-1
```

輸出：

```
{
    "disk": {
        "name": "Disk-1",
        "arn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
        "supportCode": "6EXAMPLE3362/vol-0EXAMPLEf2f88b32f",
        "createdAt": 1566585439.587,
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "resourceType": "Disk",
        "tags": [],
        "sizeInGb": 8,
        "isSystemDisk": false,
        "iops": 100,
        "path": "/dev/xvdf",
        "state": "in-use",
        "attachedTo": "WordPress_Multisite-1",
```

```
        "isAttached": true,
        "attachmentState": "attached"
    }
}
```

如需詳細資訊，請參閱 指南中的標題。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDisk。

get-disks

以下程式碼範例顯示如何使用 get-disks。

AWS CLI

取得所有區塊儲存磁碟的相關資訊

下列get-disks範例顯示已設定 AWS 區域中所有磁碟的詳細資訊。

```
aws lightsail get-disks
```

輸出：

```
{
    "disks": [
        {
            "name": "Disk-2",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:Disk/6a343ff8-6341-422d-86e2-bEXAMPLE16c2",
            "supportCode": "6EXAMPLE3362/vol-0EXAMPLE929602087",
            "createdAt": 1571090461.634,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "resourceType": "Disk",
            "tags": [],
            "sizeInGb": 8,
            "isSystemDisk": false,
            "iops": 100,
            "state": "available",
            "isAttached": false,
            "attachmentState": "detached"
```

```
        },
        {
            "name": "Disk-1",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
            "supportCode": "6EXAMPLE3362/vol-0EXAMPLEf2f88b32f",
            "createdAt": 1566585439.587,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "resourceType": "Disk",
            "tags": [],
            "sizeInGb": 8,
            "isSystemDisk": false,
            "iops": 100,
            "path": "/dev/xvdf",
            "state": "in-use",
            "attachedTo": "WordPress_Multisite-1",
            "isAttached": true,
            "attachmentState": "attached"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDisks。

get-domain

以下程式碼範例顯示如何使用 get-domain。

AWS CLI

取得網域的相關資訊

下列get-domain範例顯示網域 的詳細資訊example.com。

注意： Lightsail的網域相關 API 操作僅適用於 us-east-1 AWS 區域。如果您的 CLI 設定檔設定為使用不同的區域，您必須包含 `` --region us-east-1`` 參數，否則命令會失敗。

```
aws lightsail get-domain \
    --domain-name example.com \
    --region us-east-1
```

輸出：

```
{
    "domain": {
        "name": "example.com",
        "arn":
 "arn:aws:lightsail:global:111122223333:Domain/28cda903-3f15-44b2-9baf-3EXAMPLEb304",
        "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLEONGSC1",
        "createdAt": 1570728588.6,
        "location": {
            "availabilityZone": "all",
            "regionName": "global"
        },
        "resourceType": "Domain",
        "tags": [],
        "domainEntries": [
            {
                "id": "-1682899164",
                "name": "example.com",
                "target": "192.0.2.0",
                "isAlias": false,
                "type": "A"
            },
            {
                "id": "1703104243",
                "name": "example.com",
                "target": "ns-137.awsdns-17.com",
                "isAlias": false,
                "type": "NS"
            },
            {
                "id": "-1038331153",
                "name": "example.com",
                "target": "ns-1710.awsdns-21.co.uk",
                "isAlias": false,
                "type": "NS"
            },
            {
                "id": "-2107289565",
                "name": "example.com",
                "target": "ns-692.awsdns-22.net",
                "isAlias": false,
                "type": "NS"
            },
```

```
            {
                "id": "1582095705",
                "name": "example.com",
                "target": "ns-1436.awsdns-51.org",
                "isAlias": false,
                "type": "NS"
            },
            {
                "id": "-1769796132",
                "name": "example.com",
                "target": "ns-1710.awsdns-21.co.uk. awsdns-hostmaster.amazon.com. 1
 7200 900 1209600 86400",
                "isAlias": false,
                "type": "SOA"
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDomain。

get-domains

以下程式碼範例顯示如何使用 get-domains。

AWS CLI

取得所有網域的相關資訊

下列get-domains範例顯示已設定 AWS 區域中所有網域的詳細資訊。

注意： Lightsail的網域相關 API 操作僅適用於 us-east-1 AWS 區域。如果您的 CLI 設定檔設定為使用不同的區域，您必須包含 --region us-east-1 參數，否則命令會失敗。

```
aws lightsail get-domains \
    --region us-east-1
```

輸出：

```
{
    "domains": [
        {
            "name": "example.com",
```

```
            "arn":
 "arn:aws:lightsail:global:111122223333:Domain/28cda903-3f15-44b2-9baf-3EXAMPLEb304",
            "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLEONGSC1",
            "createdAt": 1570728588.6,
            "location": {
                "availabilityZone": "all",
                "regionName": "global"
            },
            "resourceType": "Domain",
            "tags": [],
            "domainEntries": [
                {
                    "id": "-1682899164",
                    "name": "example.com",
                    "target": "192.0.2.0",
                    "isAlias": false,
                    "type": "A"
                },
                {
                    "id": "1703104243",
                    "name": "example.com",
                    "target": "ns-137.awsdns-17.com",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "-1038331153",
                    "name": "example.com",
                    "target": "ns-4567.awsdns-21.co.uk",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "-2107289565",
                    "name": "example.com",
                    "target": "ns-333.awsdns-22.net",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "1582095705",
                    "name": "example.com",
                    "target": "ns-1111.awsdns-51.org",
                    "isAlias": false,
```

```
                    "type": "NS"
                },
                {
                    "id": "-1769796132",
                    "name": "example.com",
                    "target": "ns-1234.awsdns-21.co.uk. awsdns-
hostmaster.amazon.com. 1 7200 900 1209600 86400",
                    "isAlias": false,
                    "type": "SOA"
                },
                {
                    "id": "1029454894",
                    "name": "_dead6a124ede046a0319eb44a4eb3cbc.example.com",
                    "target": "_be133b0a0899fb7b6bf79d9741d1a383.hkvuiqjoua.acm-
validations.aws",
                    "isAlias": false,
                    "type": "CNAME"
                }
            ]
        },
        {
            "name": "example.net",
            "arn": "arn:aws:lightsail:global:111122223333:Domain/9c9f0d70-
c92e-4753-86c2-6EXAMPLE029d",
            "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLE5TPKMV",
            "createdAt": 1556661071.384,
            "location": {
                "availabilityZone": "all",
                "regionName": "global"
            },
            "resourceType": "Domain",
            "tags": [],
            "domainEntries": [
                {
                    "id": "-766320943",
                    "name": "example.net",
                    "target": "192.0.2.2",
                    "isAlias": false,
                    "type": "A"
                },
                {
                    "id": "-453913825",
                    "name": "example.net",
                    "target": "ns-123.awsdns-10.net",
```

```
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "1553601564",
                    "name": "example.net",
                    "target": "ns-4444.awsdns-47.co.uk",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "1653797661",
                    "name": "example.net",
                    "target": "ns-7890.awsdns-61.org",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "706414698",
                    "name": "example.net",
                    "target": "ns-123.awsdns-44.com",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "337271745",
                    "name": "example.net",
                    "target": "ns-4444.awsdns-47.co.uk. awsdns-
hostmaster.amazon.com. 1 7200 900 1209600 86400",
                    "isAlias": false,
                    "type": "SOA"
                },
                {
                    "id": "-1785431096",
                    "name": "www.example.net",
                    "target": "192.0.2.2",
                    "isAlias": false,
                    "type": "A"
                }
            ]
        },
        {
            "name": "example.org",
```

```
            "arn": "arn:aws:lightsail:global:111122223333:Domain/
f0f13ba3-3df0-4fdc-8ebb-1EXAMPLEf26e",
            "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLEAFO38",
            "createdAt": 1556661199.106,
            "location": {
                "availabilityZone": "all",
                "regionName": "global"
            },
            "resourceType": "Domain",
            "tags": [],
            "domainEntries": [
                {
                    "id": "2065301345",
                    "name": "example.org",
                    "target": "192.0.2.4",
                    "isAlias": false,
                    "type": "A"
                },
                {
                    "id": "-447198516",
                    "name": "example.org",
                    "target": "ns-123.awsdns-45.com",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "136463022",
                    "name": "example.org",
                    "target": "ns-9999.awsdns-15.co.uk",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "1395941679",
                    "name": "example.org",
                    "target": "ns-555.awsdns-01.net",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "872052569",
                    "name": "example.org",
                    "target": "ns-6543.awsdns-38.org",
                    "isAlias": false,
```

```
                    "type": "NS"
                },
                {
                    "id": "1001949377",
                    "name": "example.org",
                    "target": "ns-1234.awsdns-15.co.uk. awsdns-
hostmaster.amazon.com. 1 7200 900 1209600 86400",
                    "isAlias": false,
                    "type": "SOA"
                },
                {
                    "id": "1046191192",
                    "name": "www.example.org",
                    "target": "192.0.2.4",
                    "isAlias": false,
                    "type": "A"
                }
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDomains。

get-export-snapshot-record

以下程式碼範例顯示如何使用 get-export-snapshot-record。

AWS CLI

取得匯出至 Amazon EC2 的快照記錄

下列get-export-snapshot-record範例顯示匯出至 Amazon EC2 的Amazon Lightsail執行個體或磁碟快照詳細資訊。

```
aws lightsail get-export-snapshot-records
```

輸出：

```
{
    "exportSnapshotRecords": [
        {
```

```
            "name": "ExportSnapshotRecord-d2da10ce-0b3c-4ae1-ab3a-2EXAMPLEa586",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:ExportSnapshotRecord/076c7060-b0cc-4162-98f0-2EXAMPLEe28e",
            "createdAt": 1543534665.678,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "ExportSnapshotRecord",
            "state": "Succeeded",
            "sourceInfo": {
                "resourceType": "InstanceSnapshot",
                "createdAt": 1540339310.706,
                "name": "WordPress-512MB-Oregon-1-1540339219",
                "arn": "arn:aws:lightsail:us-
west-2:111122223333:InstanceSnapshot/5446f534-ed60-4c17-b4a5-bEXAMPLEf8b7",
                "fromResourceName": "WordPress-512MB-Oregon-1",
                "fromResourceArn": "arn:aws:lightsail:us-
west-2:111122223333:Instance/4b8f1f24-e4d1-4cf3-88ff-cEXAMPLEa397",
                "instanceSnapshotInfo": {
                    "fromBundleId": "nano_2_0",
                    "fromBlueprintId": "wordpress_4_9_8",
                    "fromDiskInfo": [
                        {
                            "path": "/dev/sda1",
                            "sizeInGb": 20,
                            "isSystemDisk": true
                        }
                    ]
                }
            },
            "destinationInfo": {
                "id": "ami-0EXAMPLEc0d65058e",
                "service": "Aws::EC2::Image"
            }
        },
        {
            "name": "ExportSnapshotRecord-1c94e884-40ff-4fe1-9302-0EXAMPLE14c2",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:ExportSnapshotRecord/
fb392ce8-6567-4013-9bfd-3EXAMPLE5b4c",
            "createdAt": 1543432110.2,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
```

```
            },
            "resourceType": "ExportSnapshotRecord",
            "state": "Succeeded",
            "sourceInfo": {
                "resourceType": "InstanceSnapshot",
                "createdAt": 1540833603.545,
                "name": "LAMP_PHP_5-512MB-Oregon-1-1540833565",
                "arn": "arn:aws:lightsail:us-
west-2:111122223333:InstanceSnapshot/82334399-b5f2-49ec-8382-0EXAMPLEe45f",
                "fromResourceName": "LAMP_PHP_5-512MB-Oregon-1",
                "fromResourceArn": "arn:aws:lightsail:us-
west-2:111122223333:Instance/863b9f35-ab1e-4418-bdd2-1EXAMPLEbab2",
                "instanceSnapshotInfo": {
                    "fromBundleId": "nano_2_0",
                    "fromBlueprintId": "lamp_5_6_37_2",
                    "fromDiskInfo": [
                        {
                            "path": "/dev/sda1",
                            "sizeInGb": 20,
                            "isSystemDisk": true
                        }
                    ]
                }
            },
            "destinationInfo": {
                "id": "ami-0EXAMPLE7c5ec84e2",
                "service": "Aws::EC2::Image"
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetExportSnapshotRecord。

get-instance-access-details

以下程式碼範例顯示如何使用 get-instance-access-details。

AWS CLI

取得執行個體的主機金鑰資訊

下列get-instance-access-details範例顯示執行個體 的主機金鑰資訊WordPress_Multisite-1。

```
aws lightsail get-instance-access-details \
    --instance-name WordPress_Multisite-1
```

輸出：

```
{
    "accessDetails": {
        "certKey": "ssh-rsa-cert-v01@openssh.com
 AEXAMPLEaC1yc2EtY2VydC12MDFAb3BlbnNzaC5jb20AAAAgNf076Dt3ppmPd0fPxZVMmS491aEAYYH9cHqAJ3fNML8
vEXAMPLE2eBWJyQvn7ol/
i0+s966h5sx8qUD79lPB7q5UESd5VZGFtytrykfQJnjiwqe7EV5agzvjblLj26Fb37EKda9HVfCOu8pWbvky7Tyn9w29
+xMfQM9xVz0rXZmqx8uJidJpRgLCMTviofwQJU/
K1EXAMPLEAAAAAAAABAAAALS00MzMzMDU4MzA4ODg1MTY2NjM4Onp6UWlndHk4UElRSG9STitOTG5QSEE9PQAAAAsAAA
+LiB+ozNbUA0cdNL9Y67x7qPv/R7XhTc21+2A+8+GuVpK/Kz9dqDMKNAEXAMPLE+YYN
+tiXm7Y8OgziK+7iDB7xUuQ4vghmn4+qgz9mKwYgWvVe2+0XLuV7cnWPB7iUlHQg
+E3LUKrV4ZFw9pj7X2dFdNKfMxwWgI1ISWKimEXAMPLEeHjrf1Rqc/
QH6TpWCvPfcx8uvwVqdwTfkE/SfA5BCzbGGI1UmIUadh8nHcb5FamQ1hK7kECy47K/x9FMn/
KwmM7pCwJbSLDMO7n9bnbvck6m8ZoB2N2YLMG5dW7BerEXAMPLEobqfdtyYJHHel1EyyEJs1fWNU3D5JIGlgzcPAV
+ZlbQyUCZXf0oslSa+HE85fO/
FRq9SVSBSHrmbeb0frlPhgMzgSmqLeyhlbr6wwWIDbREXAMPLEJZ49H7RdQxdKyYrZPWvRgcr0qI2EL0tAajnpQQ8UZq
Aqter0xN5PhFL0J49OWTacwCGRAjLhibAx7K1t/1ZXWo6c+ijq8clll327EXAMPLE/
e89GC89KcmKCxfGQniDAUgF8UqofIbq3ZOUgiAAYCVXclI4L68NhVXyoWuQXPBRQSEXAMPLEWm74tDL9tFN3c7tSe/
Oz0cTR+4sAAAIPAAAAB3NzaC1yc2EAAAIAQnG/
L0DqiSnLrWhEox4aHqMgd0m0oLLAYx6OQH9F0TM9EXAMPLE961rzSCMon7ZgsWNnL0OwZQgDG
+rtJ4N0B7HOVwns4ynUFbzNQ3qFGGeE3lKwX1L41vV1iSy7sDk8aI0LmrKJi1LE1Qc1l8uboRlwoXOYEXAMPLEaUCeX
+10+WEXAMPLEg6Y4U4ZvE2B3xyRdpvysb5TGFNtk5qPslacnVkoLOGsZZXMpLGJnG4OBpQLLtpj9sNMxAgZPCAUjhkqk
+nxO9O4NUZ2pTWbVSUaV1gm6pug9xbwNO1Im21t34JeLlKTqxcJ6zzS8W0c0KKpAm5c4hWkseMbyutS2jav/4hiS
+BhrYgptzfwe5qRXEXAMPLEHZQr3YfGzYoBJ/
lLK3NHhxOihhsfAYwMei0BFZT1F/7CT3IH4iitEkIgodi06/
Mw6UDqMPozyQCK1lEA6LFhYCOZG9drWcoRa74lM4kY9TP028Za8gDMh1WpkXLq9Gixon5OHP8aM/
sEXAMPLEr2+fnkw+1BtoO5L6+VKoPlXaGqZ/fBYEXAMPLEAMQHjnLM1JYNvtEEPhp+TNzXHzuixWf/
Ht04m0AVpXrzIDXaS1O2tXY=",
        "ipAddress": "192.0.2.0",
        "privateKey": "-----BEGIN RSA PRIVATE KEY-----
\nEXAMPLEBAAKCAQEA+AD3qeU2toBy5O5v7wnRLVo/tngVickL5+6Jf4tPrPeuoebM
\nfKlA+/ZTwe6uVBEneVWRhbcra8pH0CZ44sKnuxFeWoM7425S49uhW9+xCnWvR1Xw
\njrvKVm75Mu08p/cNvfWugrBuaPB65DspgxNnOfZWMVxpIpSqOSPWmSwQHV597d6C
\nrEXAMPLEo8hJmqz2KFQ09X7fB2lBruGgr9aXiNPmWmovYKqwFmrnFvR7odFmDecq
\n5EXAMPLE9dyU1ZsrWhGby77eYrVaFl0GNGQ8qy1HGUiscquZ9NDIL49n4mXbfsTH
\n0EXAMPLE12ZqsfLiYnSaUYCwjE74qH8ECVPytQIDAQABAoIBAHeZV9Z58JHAjifz
```

```
\nCEXAMPLEEqC3doOVDgXSlkKI92qNo4z2VcUEho878paCuVVXVHcCGgSnGeyIh2tN
\nMEXAMPLESohR427BhH3YLA+3Z5SIvnejbTgYPfLC37B8khTaYqkqMvdZiFVZK5qn
\nIEXAMPLEM93oF9eSZCjcLKB/jGHsfb0eCDMP8BshHE2beuqzVMoK1DxOnvoP3+Fp
\nAEXAMPLESq6pDpCo9YVUX8g1u3Ro9cPl2LXHDy+oVEY5KhbZQJ7VU1I72WOvppWW
\nOEXAMPLEkgYlq7p6qYtYcSgTEjz14gDiMfQ7SyHB3alkIoNONQ9ZPaWHyJvymeud
\noQTNuz0CgYEA/LFWNTEZrzdzdR1kJmyNRmAermU0B6utyNENChAlHGSHkB+1lVSh
\nbEXAMPLEQo9ooUeW5UxO3YwacZLoDT1mwxw1Ptc1+PNycZoLe1fE9UdARrdmGTob
\n8l7CPLSXp3xuR8VqSp2fnIc7hfiQs/NrPX9gm/EOrB0we0RKyDSzWScCgYEA+z/r
\niob+nJZq0YbnOSuP6oMULP4vnWniWj8MIhUJU53LwSAM8DeJdONKDdkuiOd52aAL
\nVgn7nLo88rVWKhJwVc4tu/rNgZLcR3bP4+kL6zand0KQnMLyOzNA2Ys26aa5udH1\nqWl0WTt9WEm/
h10ndC1knOMectrvsG17b38y5sMCgYEA54NiRGGz8oCPW6GN/FZA
\nKEXAMPLE5tw34GEH3Uxlc9n3CejDaQmczOATwX4nIwRZDEqWyYZcS0btg1jhGiBD\nYEXAMPLEkc8Z71L/
agZEAaVCEog9FqfSqwB
+XTfoKh8qur74X1yCu9p6gof1q6k9\neEXAMPLEchJcNNOg4ETIfMkCgYBdVORRhE4mqvWpOdzA7v66FdEz2YSkjAXKk
\naEXAMPLE8Z/8yBSmuBv1Qv03XA12my462uB92uzzGAuW
+1yBc2Kn1sXqYTy0y1z0\ngEXAMPLEBogjw4MqHKL1bPKMHyQU8/
q24PaYgzHPzy13wlH6pTYf1XqlHdE2D6Vv\nyEXAMPLEgQC3i/
kVVhky/2XRwRVlC7JO2Bg3QGTx38hpmDa5IuofKANjA+Wa3/zy\nbEXAMPLE6ytQgD9GN/YtBq+uhO
+2ZkvXPL+CWRi0ZRXpPwYDBBFU9Cw0AuWWGlL8\nwEXAMPLExMlcysRgcWB9RNgf3AuOpFd2i6XT/
riNsvvkpmJ+VooU8g==\n-----END RSA PRIVATE KEY-----\n",
        "protocol": "ssh",
        "instanceName": "WordPress_Multisite-1",
        "username": "bitnami",
        "hostKeys": [
            {
                "algorithm": "ssh-rsa",
                "publicKey":
 "AEXAMPLEaC1yc2EAAAADAQABAAABAQCoeR9ieZTjQ3pXCHczuAYZFjlF7t
+uBkXuqeGMRex78pCvmS+DiEXAMPLEuJ1Q8dcKhrQL4HpXbD9dosVCTaJnJwb4MQqsuSVFdHFzy3guP
+BKclWqtxJEXAMPLEsBGqZZlrIv6a9bTA0TCplZ8AD+hSRTaSXXqg6FT
+Qf16IktH0XlMs7xIEXAMPLEmNtjCpzZiGXDHzytoMvUgwa8uHPp44Og36EUu4VqQxoUHPJKoXvcQizyk3K8ym0hP0Tp
Ot6y9HwvykEXAMPLEAfbKjbR42+u6+OSlkr4d339q2U1sTDytJhhs8HUel1wTfGRfp",
                "witnessedAt": 1570744377.699,
                "fingerprintSHA1": "SHA1:GEXAMPLEMoYgUg0ucadqU9Bt3Lk",
                "fingerprintSHA256": "SHA256:IEXAMPLEcB5vgxnAUoJawbdZ
+MwELhIp6FUxuwq/LIU"
            },
            {
                "algorithm": "ssh-ed25519",
                "publicKey":
 "AEXAMPLEaC1lZDI1NTE5AAAAIC1gwGPDfGaONxEXAMPLEJX3UNap781QxHQmn8nzlrUv",
                "witnessedAt": 1570744377.697,
                "fingerprintSHA1": "SHA1:VEXAMPLE5ReqSmTgv03sSUw9toU",
```

```
                "fingerprintSHA256": "SHA256:0EXAMPLEdE6tI95k3TJpG
+qhJbAoknB0yz9nAEaDt3A"
            },
            {
                "algorithm": "ecdsa-sha2-nistp256",
                "publicKey":
 "AEXAMPLEZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABEXAMPLE9B4mZy8YSsZW7cixCDq5yHSAAxjJkDo5
+EnKlDCsYtUkxxEXAMPLE6VOWL2z63RTKa2AUPgd8irjxWI=",
                "witnessedAt": 1570744377.707,
                "fingerprintSHA1": "SHA1:UEXAMPLEOYCfXsCf2G6tDg+7YG0",
                "fingerprintSHA256": "SHA256:wEXAMPLEQ9a/
iEXAMPLEhRufm6U9vFU4cpkMPHnBsNA"
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInstanceAccessDetails。

get-instance-metric-data

以下程式碼範例顯示如何使用 get-instance-metric-data。

AWS CLI

取得執行個體的指標資料

下列get-instance-metric-data範例會傳回 1571342400與 之間CPUUtilization每7200秒 (2 小時） 的平均百分比1571428800，例如 MEAN-1。

建議您使用 unix 時間轉換器來識別開始和結束時間。

```
aws lightsail get-instance-metric-data \
    --instance-name MEAN-1 \
    --metric-name CPUUtilization \
    --period 7200 \
    --start-time 1571342400 \
    --end-time 1571428800 \
    --unit Percent \
    --statistics Average
```

輸出：

```
{
    "metricName": "CPUUtilization",
    "metricData": [
        {
            "average": 0.26113718770120725,
            "timestamp": 1571342400.0,
            "unit": "Percent"
        },
        {
            "average": 0.26861268928111953,
            "timestamp": 1571392800.0,
            "unit": "Percent"
        },
        {
            "average": 0.28187475104748777,
            "timestamp": 1571378400.0,
            "unit": "Percent"
        },
        {
            "average": 0.2651936960458352,
            "timestamp": 1571421600.0,
            "unit": "Percent"
        },
        {
            "average": 0.2561856213712188,
            "timestamp": 1571371200.0,
            "unit": "Percent"
        },
        {
            "average": 0.3021383254607764,
            "timestamp": 1571356800.0,
            "unit": "Percent"
        },
        {
            "average": 0.2618381649223539,
            "timestamp": 1571407200.0,
            "unit": "Percent"
        },
        {
            "average": 0.26331929394825787,
            "timestamp": 1571400000.0,
            "unit": "Percent"
        },
```

```
        {
            "average": 0.2576348407007818,
            "timestamp": 1571385600.0,
            "unit": "Percent"
        },
        {
            "average": 0.2513008454658378,
            "timestamp": 1571364000.0,
            "unit": "Percent"
        },
        {
            "average": 0.26329974562758346,
            "timestamp": 1571414400.0,
            "unit": "Percent"
        },
        {
            "average": 0.2667092536656445,
            "timestamp": 1571349600.0,
            "unit": "Percent"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInstanceMetricData。

get-instance-port-states

以下程式碼範例顯示如何使用 get-instance-port-states。

AWS CLI

取得執行個體的防火牆資訊

下列get-instance-port-states範例會傳回為執行個體 設定的防火牆連接埠MEAN-1。

```
aws lightsail get-instance-port-states \
    --instance-name MEAN-1
```

輸出：

```
{
    "portStates": [
        {
```

```
            "fromPort": 80,
            "toPort": 80,
            "protocol": "tcp",
            "state": "open"
        },
        {
            "fromPort": 22,
            "toPort": 22,
            "protocol": "tcp",
            "state": "open"
        },
        {
            "fromPort": 443,
            "toPort": 443,
            "protocol": "tcp",
            "state": "open"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInstancePortStates。

get-instance-snapshot

以下程式碼範例顯示如何使用 get-instance-snapshot。

AWS CLI

取得指定執行個體快照的相關資訊

下列get-instance-snapshot範例顯示指定執行個體快照的詳細資訊。

```
aws lightsail get-instance-snapshot \
    --instance-snapshot-name MEAN-1-1571419854
```

輸出：

```
{
    "instanceSnapshot": {
        "name": "MEAN-1-1571419854",
        "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/
ac54700c-48a8-40fd-b065-2EXAMPLEac8f",
        "supportCode": "6EXAMPLE3362/ami-0EXAMPLE67a73020d",
```

```
        "createdAt": 1571419891.927,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "resourceType": "InstanceSnapshot",
        "tags": [],
        "state": "available",
        "fromAttachedDisks": [],
        "fromInstanceName": "MEAN-1",
        "fromInstanceArn": "arn:aws:lightsail:us-west-2:111122223333:Instance/
bd470fc5-a68b-44c5-8dbc-8EXAMPLEbada",
        "fromBlueprintId": "mean",
        "fromBundleId": "medium_3_0",
        "isFromAutoSnapshot": false,
        "sizeInGb": 80
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInstanceSnapshot。

get-instance-snapshots

以下程式碼範例顯示如何使用 get-instance-snapshots。

AWS CLI

取得所有執行個體快照的相關資訊

下列get-instance-snapshots範例顯示所設定 AWS 區域中所有執行個體快照的詳細資訊。

```
aws lightsail get-instance-snapshots
```

輸出：

```
{
    "instanceSnapshots": [
        {
            "name": "MEAN-1-1571421498",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/
a20e6ebe-b0ee-4ae4-a750-3EXAMPLEcb0c",
            "supportCode": "6EXAMPLE3362/ami-0EXAMPLEe33cabfa1",
            "createdAt": 1571421527.755,
```

```
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "InstanceSnapshot",
            "tags": [
                {
                    "key": "no_delete"
                }
            ],
            "state": "available",
            "fromAttachedDisks": [],
            "fromInstanceName": "MEAN-1",
            "fromInstanceArn": "arn:aws:lightsail:us-
west-2:111122223333:Instance/1761aa0a-6038-4f25-8b94-2EXAMPLE19fd",
            "fromBlueprintId": "wordpress",
            "fromBundleId": "micro_3_0",
            "isFromAutoSnapshot": false,
            "sizeInGb": 40
        },
        {
            "name": "MEAN-1-1571419854",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/
ac54700c-48a8-40fd-b065-2EXAMPLEac8f",
            "supportCode": "6EXAMPLE3362/ami-0EXAMPLE67a73020d",
            "createdAt": 1571419891.927,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "InstanceSnapshot",
            "tags": [],
            "state": "available",
            "fromAttachedDisks": [],
            "fromInstanceName": "MEAN-1",
            "fromInstanceArn": "arn:aws:lightsail:us-west-2:111122223333:Instance/
bd470fc5-a68b-44c5-8dbc-8EXAMPLEbada",
            "fromBlueprintId": "mean",
            "fromBundleId": "medium_3_0",
            "isFromAutoSnapshot": false,
            "sizeInGb": 80
        }
    ]
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInstanceSnapshots。

get-instance-state

以下程式碼範例顯示如何使用 get-instance-state。

AWS CLI

取得執行個體狀態的相關資訊

下列get-instance-state範例會傳回指定執行個體的狀態。

```
aws lightsail get-instance-state \
    --instance-name MEAN-1
```

輸出：

```
{
    "state": {
        "code": 16,
        "name": "running"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInstanceState。

get-instance

以下程式碼範例顯示如何使用 get-instance。

AWS CLI

取得執行個體的相關資訊

下列get-instance範例顯示執行個體 的詳細資訊MEAN-1。

```
aws lightsail get-instance \
    --instance-name MEAN-1
```

輸出：

```
{
    "instance": {
        "name": "MEAN-1",
        "arn": "arn:aws:lightsail:us-west-2:111122223333:Instance/bd470fc5-
a68b-44c5-8dbc-EXAMPLE4bada",
        "supportCode": "6EXAMPLE3362/i-05EXAMPLE407c97d3",
        "createdAt": 1570635023.124,
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "resourceType": "Instance",
        "tags": [],
        "blueprintId": "mean",
        "blueprintName": "MEAN",
        "bundleId": "medium_3_0",
        "isStaticIp": false,
        "privateIpAddress": "192.0.2.0",
        "publicIpAddress": "192.0.2.0",
        "hardware": {
            "cpuCount": 2,
            "disks": [
                {
                    "createdAt": 1570635023.124,
                    "sizeInGb": 80,
                    "isSystemDisk": true,
                    "iops": 240,
                    "path": "/dev/xvda",
                    "attachedTo": "MEAN-1",
                    "attachmentState": "attached"
                }
            ],
            "ramSizeInGb": 4.0
        },
        "networking": {
            "monthlyTransfer": {
                "gbPerMonthAllocated": 4096
            },
            "ports": [
                {
                    "fromPort": 80,
                    "toPort": 80,
```

```
                    "protocol": "tcp",
                    "accessFrom": "Anywhere (0.0.0.0/0)",
                    "accessType": "public",
                    "commonName": "",
                    "accessDirection": "inbound"
                },
                {
                    "fromPort": 22,
                    "toPort": 22,
                    "protocol": "tcp",
                    "accessFrom": "Anywhere (0.0.0.0/0)",
                    "accessType": "public",
                    "commonName": "",
                    "accessDirection": "inbound"
                },
                {
                    "fromPort": 443,
                    "toPort": 443,
                    "protocol": "tcp",
                    "accessFrom": "Anywhere (0.0.0.0/0)",
                    "accessType": "public",
                    "commonName": "",
                    "accessDirection": "inbound"
                }
            ]
        },
        "state": {
            "code": 16,
            "name": "running"
        },
        "username": "bitnami",
        "sshKeyName": "MyKey"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInstance。

get-instances

以下程式碼範例顯示如何使用 get-instances。

AWS CLI

取得所有執行個體的相關資訊

下列get-instances範例顯示已設定 AWS 區域中所有執行個體的詳細資訊。

```
aws lightsail get-instances
```

輸出：

```
{
    "instances": [
        {
            "name": "Windows_Server_2022-1",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:Instance/0f44fbb9-8f55-4e47-a25e-EXAMPLE04763",
            "supportCode": "62EXAMPLE362/i-0bEXAMPLE71a686b9",
            "createdAt": 1571332358.665,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "resourceType": "Instance",
            "tags": [],
            "blueprintId": "windows_server_2022",
            "blueprintName": "Windows Server 2022",
            "bundleId": "large_win_3_0",
            "isStaticIp": false,
            "privateIpAddress": "192.0.2.0",
            "publicIpAddress": "192.0.2.0",
            "hardware": {
                "cpuCount": 1,
                "disks": [
                    {
                        "createdAt": 1571332358.665,
                        "sizeInGb": 160,
                        "isSystemDisk": true,
                        "iops": 180,
                        "path": "/dev/sda1",
                        "attachedTo": "Windows_Server_2022-1",
                        "attachmentState": "attached"
                    },
                    {
```

```
                        "name": "my-disk-for-windows-server",
                        "arn": "arn:aws:lightsail:us-
west-2:111122223333:Disk/4123a81c-484c-49ea-afea-5EXAMPLEda87",
                        "supportCode": "6EXAMPLE3362/vol-0EXAMPLEb2b99ca3d",
                        "createdAt": 1571355063.494,
                        "location": {
                            "availabilityZone": "us-west-2a",
                            "regionName": "us-west-2"
                        },
                        "resourceType": "Disk",
                        "tags": [],
                        "sizeInGb": 128,
                        "isSystemDisk": false,
                        "iops": 384,
                        "path": "/dev/xvdf",
                        "state": "in-use",
                        "attachedTo": "Windows_Server_2022-1",
                        "isAttached": true,
                        "attachmentState": "attached"
                    }
                ],
                "ramSizeInGb": 8.0
            },
            "networking": {
                "monthlyTransfer": {
                    "gbPerMonthAllocated": 3072
                },
                "ports": [
                    {
                        "fromPort": 80,
                        "toPort": 80,
                        "protocol": "tcp",
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
                        "commonName": "",
                        "accessDirection": "inbound"
                    },
                    {
                        "fromPort": 22,
                        "toPort": 22,
                        "protocol": "tcp",
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
                        "commonName": "",
```

```
                        "accessDirection": "inbound"
                    },
                    {
                        "fromPort": 3389,
                        "toPort": 3389,
                        "protocol": "tcp",
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
                        "commonName": "",
                        "accessDirection": "inbound"
                    }
                ]
            },
            "state": {
                "code": 16,
                "name": "running"
            },
            "username": "Administrator",
            "sshKeyName": "LightsailDefaultKeyPair"
        },
        {
            "name": "MEAN-1",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:Instance/bd470fc5-
a68b-44c5-8dbc-8EXAMPLEbada",
            "supportCode": "6EXAMPLE3362/i-0EXAMPLEa407c97d3",
            "createdAt": 1570635023.124,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "resourceType": "Instance",
            "tags": [],
            "blueprintId": "mean",
            "blueprintName": "MEAN",
            "bundleId": "medium_3_0",
            "isStaticIp": false,
            "privateIpAddress": "192.0.2.0",
            "publicIpAddress": "192.0.2.0",
            "hardware": {
                "cpuCount": 2,
                "disks": [
                    {
                        "name": "Disk-1",
```

```
                        "arn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
                        "supportCode": "6EXAMPLE3362/vol-0EXAMPLEf2f88b32f",
                        "createdAt": 1566585439.587,
                        "location": {
                            "availabilityZone": "us-west-2a",
                            "regionName": "us-west-2"
                        },
                        "resourceType": "Disk",
                        "tags": [
                            {
                                "key": "test"
                            }
                        ],
                        "sizeInGb": 8,
                        "isSystemDisk": false,
                        "iops": 240,
                        "path": "/dev/xvdf",
                        "state": "in-use",
                        "attachedTo": "MEAN-1",
                        "isAttached": true,
                        "attachmentState": "attached"
                    },
                    {
                        "createdAt": 1570635023.124,
                        "sizeInGb": 80,
                        "isSystemDisk": true,
                        "iops": 240,
                        "path": "/dev/sda1",
                        "attachedTo": "MEAN-1",
                        "attachmentState": "attached"
                    }
                ],
                "ramSizeInGb": 4.0
            },
            "networking": {
                "monthlyTransfer": {
                    "gbPerMonthAllocated": 4096
                },
                "ports": [
                    {
                        "fromPort": 80,
                        "toPort": 80,
                        "protocol": "tcp",
```

```
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
                        "commonName": "",
                        "accessDirection": "inbound"
                    },
                    {
                        "fromPort": 22,
                        "toPort": 22,
                        "protocol": "tcp",
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
                        "commonName": "",
                        "accessDirection": "inbound"
                    },
                    {
                        "fromPort": 443,
                        "toPort": 443,
                        "protocol": "tcp",
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
                        "commonName": "",
                        "accessDirection": "inbound"
                    }
                ]
            },
            "state": {
                "code": 16,
                "name": "running"
            },
            "username": "bitnami",
            "sshKeyName": "MyTestKey"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInstances。

get-key-pair

以下程式碼範例顯示如何使用 get-key-pair。

AWS CLI

取得金鑰對的相關資訊

下列get-key-pair範例顯示指定金鑰對的詳細資訊。

```
aws lightsail get-key-pair \
    --key-pair-name MyKey1
```

輸出：

```
{
    "keyPair": {
        "name": "MyKey1",
        "arn": "arn:aws:lightsail:us-
west-2:111122223333:KeyPair/19a4efdf-3054-43d6-91fd-eEXAMPLE21bf",
        "supportCode": "6EXAMPLE3362/MyKey1",
        "createdAt": 1571255026.975,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "resourceType": "KeyPair",
        "tags": [],
        "fingerprint": "00:11:22:33:44:55:66:77:88:99:aa:bb:cc:dd:ee:ff:gg:hh:ii:jj"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetKeyPair。

get-key-pairs

以下程式碼範例顯示如何使用 get-key-pairs。

AWS CLI

取得所有金鑰對的相關資訊

下列get-key-pairs範例顯示已設定 AWS 區域中所有金鑰對的詳細資訊。

```
aws lightsail get-key-pairs
```

輸出：

```
{
    "keyPairs": [
        {
            "name": "MyKey1",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:KeyPair/19a4efdf-3054-43d6-91fd-eEXAMPLE21bf",
            "supportCode": "6EXAMPLE3362/MyKey1",
            "createdAt": 1571255026.975,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "KeyPair",
            "tags": [],
            "fingerprint":
 "00:11:22:33:44:55:66:77:88:99:aa:bb:cc:dd:ee:ff:gg:hh:ii:jj"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetKeyPairs。

get-load-balancer-tls-certificates

以下程式碼範例顯示如何使用 get-load-balancer-tls-certificates。

AWS CLI

取得負載平衡器 TLS 憑證的相關資訊

下列get-load-balancer-tls-certificates範例顯示指定負載平衡器之 TLS 憑證的詳細資訊。

```
aws lightsail get-load-balancer-tls-certificates \
    --load-balancer-name LoadBalancer-1
```

輸出：

```
{
```

```
    "tlsCertificates": [
        {
            "name": "example-com",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:LoadBalancerTlsCertificate/d7bf4643-6a02-4cd4-b3c4-
fEXAMPLE9b4d",
            "supportCode": "6EXAMPLE3362/arn:aws:acm:us-
west-2:333322221111:certificate/9af8e32c-a54e-4a67-8c63-cEXAMPLEb314",
            "createdAt": 1571678025.3,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "LoadBalancerTlsCertificate",
            "loadBalancerName": "LoadBalancer-1",
            "isAttached": false,
            "status": "ISSUED",
            "domainName": "example.com",
            "domainValidationRecords": [
                {
                    "name": "_dEXAMPLE4ede046a0319eb44a4eb3cbc.example.com.",
                    "type": "CNAME",
                    "value": "_bEXAMPLE0899fb7b6bf79d9741d1a383.hkvuiqjoua.acm-
validations.aws.",
                    "validationStatus": "SUCCESS",
                    "domainName": "example.com"
                }
            ],
            "issuedAt": 1571678070.0,
            "issuer": "Amazon",
            "keyAlgorithm": "RSA-2048",
            "notAfter": 1605960000.0,
            "notBefore": 1571616000.0,
            "serial": "00:11:22:33:44:55:66:77:88:99:aa:bb:cc:dd:ee:ff",
            "signatureAlgorithm": "SHA256WITHRSA",
            "subject": "CN=example.com",
            "subjectAlternativeNames": [
                "example.com"
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLoadBalancerTlsCertificates。

get-load-balancer

以下程式碼範例顯示如何使用 get-load-balancer。

AWS CLI

取得負載平衡器的相關資訊

下列get-load-balancer範例顯示指定負載平衡器的詳細資訊。

```
aws lightsail get-load-balancer \
    --load-balancer-name LoadBalancer-1
```

輸出：

```
{
    "loadBalancer": {
        "name": "LoadBalancer-1",
        "arn": "arn:aws:lightsail:us-
west-2:111122223333:LoadBalancer/40486b2b-1ad0-4152-83e4-cEXAMPLE6f4b",
        "supportCode": "6EXAMPLE3362/arn:aws:elasticloadbalancing:us-
west-2:333322221111:loadbalancer/app/
bEXAMPLE128cb59d86f946a9395dd304/1EXAMPLE8dd9d77e",
        "createdAt": 1571677906.723,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "resourceType": "LoadBalancer",
        "tags": [],
        "dnsName": "bEXAMPLE128cb59d86f946a9395dd304-1486911371.us-
west-2.elb.amazonaws.com",
        "state": "active",
        "protocol": "HTTP",
        "publicPorts": [
            80
        ],
        "healthCheckPath": "/",
        "instancePort": 80,
        "instanceHealthSummary": [
            {
                "instanceName": "MEAN-3",
                "instanceHealth": "healthy"
```

```
            },
            {
                "instanceName": "MEAN-1",
                "instanceHealth": "healthy"
            },
            {
                "instanceName": "MEAN-2",
                "instanceHealth": "healthy"
            }
        ],
        "tlsCertificateSummaries": [
            {
                "name": "example-com",
                "isAttached": false
            }
        ],
        "configurationOptions": {
            "SessionStickinessEnabled": "false",
            "SessionStickiness_LB_CookieDurationSeconds": "86400"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLoadBalancer。

get-load-balancers

以下程式碼範例顯示如何使用 get-load-balancers。

AWS CLI

取得所有負載平衡器的相關資訊

下列get-load-balancers範例顯示所設定 AWS 區域中所有負載平衡器的詳細資訊。

```
aws lightsail get-load-balancers
```

輸出：

```
{
    "loadBalancers": [
        {
            "name": "LoadBalancer-1",
```

```
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:LoadBalancer/40486b2b-1ad0-4152-83e4-cEXAMPLE6f4b",
            "supportCode": "6EXAMPLE3362/arn:aws:elasticloadbalancing:us-
west-2:333322221111:loadbalancer/app/
bEXAMPLE128cb59d86f946a9395dd304/1EXAMPLE8dd9d77e",
            "createdAt": 1571677906.723,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "LoadBalancer",
            "tags": [],
            "dnsName": "bEXAMPLE128cb59d86f946a9395dd304-1486911371.us-
west-2.elb.amazonaws.com",
            "state": "active",
            "protocol": "HTTP",
            "publicPorts": [
                80
            ],
            "healthCheckPath": "/",
            "instancePort": 80,
            "instanceHealthSummary": [
                {
                    "instanceName": "MEAN-3",
                    "instanceHealth": "healthy"
                },
                {
                    "instanceName": "MEAN-1",
                    "instanceHealth": "healthy"
                },
                {
                    "instanceName": "MEAN-2",
                    "instanceHealth": "healthy"
                }
            ],
            "tlsCertificateSummaries": [
                {
                    "name": "example-com",
                    "isAttached": false
                }
            ],
            "configurationOptions": {
                "SessionStickinessEnabled": "false",
                "SessionStickiness_LB_CookieDurationSeconds": "86400"
```

```
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLoadBalancers。

get-operation

以下程式碼範例顯示如何使用 get-operation。

AWS CLI

取得單一操作的相關資訊

下列get-operation範例顯示指定操作的詳細資訊。

```
aws lightsail get-operation \
    --operation-id e5700e8a-daf2-4b49-bc01-3EXAMPLE910a
```

輸出：

```
{
    "operation": {
        "id": "e5700e8a-daf2-4b49-bc01-3EXAMPLE910a",
        "resourceName": "Instance-1",
        "resourceType": "Instance",
        "createdAt": 1571679872.404,
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "isTerminal": true,
        "operationType": "CreateInstance",
        "status": "Succeeded",
        "statusChangedAt": 1571679890.304
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetOperation。

get-operations-for-resource

以下程式碼範例顯示如何使用 get-operations-for-resource。

AWS CLI

取得資源的所有操作

下列get-operations-for-resource範例顯示指定資源之所有操作的詳細資訊。

```
aws lightsail get-operations-for-resource \
    --resource-name LoadBalancer-1
```

輸出：

```
{
    "operations": [
        {
            "id": "e2973046-43f8-4252-a4b4-9EXAMPLE69ce",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571678786.071,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "MEAN-1",
            "operationType": "DetachInstancesFromLoadBalancer",
            "status": "Succeeded",
            "statusChangedAt": 1571679087.57
        },
        {
            "id": "2d742a18-0e7f-48c8-9705-3EXAMPLEf98a",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571678782.784,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "MEAN-1",
            "operationType": "AttachInstancesToLoadBalancer",
```

```
            "status": "Succeeded",
            "statusChangedAt": 1571678798.465
        },
        {
            "id": "6c700fcc-4246-40ab-952b-1EXAMPLEdac2",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571678775.297,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "MEAN-3",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Succeeded",
            "statusChangedAt": 1571678842.806
        },
        ...
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetOperationsForResource。

get-operations

以下程式碼範例顯示如何使用 get-operations。

AWS CLI

取得所有操作的相關資訊

下列get-operations範例顯示所設定 AWS 區域中所有操作的詳細資訊。

```
aws lightsail get-operations
```

輸出：

```
{
    "operations": [
        {
            "id": "e5700e8a-daf2-4b49-bc01-3EXAMPLE910a",
```

```
            "resourceName": "Instance-1",
            "resourceType": "Instance",
            "createdAt": 1571679872.404,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "CreateInstance",
            "status": "Succeeded",
            "statusChangedAt": 1571679890.304
        },
        {
            "id": "701a3339-930e-4914-a9f9-7EXAMPLE68d7",
            "resourceName": "WordPress-1",
            "resourceType": "Instance",
            "createdAt": 1571678786.072,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "LoadBalancer-1",
            "operationType": "DetachInstancesFromLoadBalancer",
            "status": "Succeeded",
            "statusChangedAt": 1571679086.399
        },
        {
            "id": "e2973046-43f8-4252-a4b4-9EXAMPLE69ce",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571678786.071,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "WordPress-1",
            "operationType": "DetachInstancesFromLoadBalancer",
            "status": "Succeeded",
            "statusChangedAt": 1571679087.57
        },
        ...
        }
```

```
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetOperations。

get-regions

以下程式碼範例顯示如何使用 get-regions。

AWS CLI

取得 的所有 AWS 區域 Amazon Lightsail

下列get-regions範例顯示 所有 AWS 區域的詳細資訊Amazon Lightsail。

```
aws lightsail get-regions
```

輸出：

```
{
    "regions": [
        {
            "continentCode": "NA",
            "description": "This region is recommended to serve users in the eastern
 United States",
            "displayName": "Virginia",
            "name": "us-east-1",
            "availabilityZones": [],
            "relationalDatabaseAvailabilityZones": []
        },
        {
            "continentCode": "NA",
            "description": "This region is recommended to serve users in the eastern
 United States",
            "displayName": "Ohio",
            "name": "us-east-2",
            "availabilityZones": [],
            "relationalDatabaseAvailabilityZones": []
        },
        {
            "continentCode": "NA",
            "description": "This region is recommended to serve users in the
 northwestern United States, Alaska, and western Canada",
```

```
            "displayName": "Oregon",
            "name": "us-west-2",
            "availabilityZones": [],
            "relationalDatabaseAvailabilityZones": []
        },
        ...
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRegions。

get-relational-database-blueprints

以下程式碼範例顯示如何使用 get-relational-database-blueprints。

AWS CLI

取得新關聯式資料庫的藍圖

下列get-relational-database-blueprints範例顯示所有可用關聯式資料庫藍圖的詳細資訊，可用於在 中建立新的關聯式資料庫Amazon Lightsail。

```
aws lightsail get-relational-database-blueprints
```

輸出：

```
{
    "blueprints": [
        {
            "blueprintId": "mysql_5_6",
            "engine": "mysql",
            "engineVersion": "5.6.44",
            "engineDescription": "MySQL Community Edition",
            "engineVersionDescription": "MySQL 5.6.44",
            "isEngineDefault": false
        },
        {
            "blueprintId": "mysql_5_7",
            "engine": "mysql",
            "engineVersion": "5.7.26",
            "engineDescription": "MySQL Community Edition",
```

```
            "engineVersionDescription": "MySQL 5.7.26",
            "isEngineDefault": true
        },
        {
            "blueprintId": "mysql_8_0",
            "engine": "mysql",
            "engineVersion": "8.0.16",
            "engineDescription": "MySQL Community Edition",
            "engineVersionDescription": "MySQL 8.0.16",
            "isEngineDefault": false
        },
        {
            "blueprintId": "postgres_9_6",
            "engine": "postgres",
            "engineVersion": "9.6.15",
            "engineDescription": "PostgreSQL",
            "engineVersionDescription": "PostgreSQL 9.6.15-R1",
            "isEngineDefault": false
        },
        {
            "blueprintId": "postgres_10",
            "engine": "postgres",
            "engineVersion": "10.10",
            "engineDescription": "PostgreSQL",
            "engineVersionDescription": "PostgreSQL 10.10-R1",
            "isEngineDefault": false
        },
        {
            "blueprintId": "postgres_11",
            "engine": "postgres",
            "engineVersion": "11.5",
            "engineDescription": "PostgreSQL",
            "engineVersionDescription": "PostgreSQL 11.5-R1",
            "isEngineDefault": true
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRelationalDatabaseBlueprints。

get-relational-database-bundles

以下程式碼範例顯示如何使用 get-relational-database-bundles。

AWS CLI

取得新關聯式資料庫的套件

下列get-relational-database-bundles範例顯示所有可用關聯式資料庫套件的詳細資訊，可用於在 中建立新的關聯式資料庫Amazon Lightsail。請注意，回應不包含非作用中的套件，因為命令中未指定 --include-inactive旗標。您無法使用非作用中的套件建立新的關聯式資料庫。

```
aws lightsail get-relational-database-bundles
```

輸出：

```
{
    "bundles": [
        {
            "bundleId": "micro_2_0",
            "name": "Micro",
            "price": 15.0,
            "ramSizeInGb": 1.0,
            "diskSizeInGb": 40,
            "transferPerMonthInGb": 100,
            "cpuCount": 2,
            "isEncrypted": true,
            "isActive": true
        },
        {
            "bundleId": "micro_ha_2_0",
            "name": "Micro with High Availability",
            "price": 30.0,
            "ramSizeInGb": 1.0,
            "diskSizeInGb": 40,
            "transferPerMonthInGb": 100,
            "cpuCount": 2,
            "isEncrypted": true,
            "isActive": true
        },
        {
            "bundleId": "small_2_0",
            "name": "Small",
            "price": 30.0,
            "ramSizeInGb": 2.0,
            "diskSizeInGb": 80,
            "transferPerMonthInGb": 100,
```

```
            "cpuCount": 2,
            "isEncrypted": true,
            "isActive": true
        },
        {
            "bundleId": "small_ha_2_0",
            "name": "Small with High Availability",
            "price": 60.0,
            "ramSizeInGb": 2.0,
            "diskSizeInGb": 80,
            "transferPerMonthInGb": 100,
            "cpuCount": 2,
            "isEncrypted": true,
            "isActive": true
        },
        {
            "bundleId": "medium_2_0",
            "name": "Medium",
            "price": 60.0,
            "ramSizeInGb": 4.0,
            "diskSizeInGb": 120,
            "transferPerMonthInGb": 100,
            "cpuCount": 2,
            "isEncrypted": true,
            "isActive": true
        },
        {
            "bundleId": "medium_ha_2_0",
            "name": "Medium with High Availability",
            "price": 120.0,
            "ramSizeInGb": 4.0,
            "diskSizeInGb": 120,
            "transferPerMonthInGb": 100,
            "cpuCount": 2,
            "isEncrypted": true,
            "isActive": true
        },
        {
            "bundleId": "large_2_0",
            "name": "Large",
            "price": 115.0,
            "ramSizeInGb": 8.0,
            "diskSizeInGb": 240,
            "transferPerMonthInGb": 200,
```

```
            "cpuCount": 2,
            "isEncrypted": true,
            "isActive": true
        },
        {
            "bundleId": "large_ha_2_0",
            "name": "Large with High Availability",
            "price": 230.0,
            "ramSizeInGb": 8.0,
            "diskSizeInGb": 240,
            "transferPerMonthInGb": 200,
            "cpuCount": 2,
            "isEncrypted": true,
            "isActive": true
        }
    ]
}
```

如需詳細資訊，請參閱《 Amazon Lightsail開發人員指南》中的在 中建立資料庫Amazon Lightsail。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRelationalDatabaseBundles。

get-relational-database-events

以下程式碼範例顯示如何使用 get-relational-database-events。

AWS CLI

取得關聯式資料庫的事件

下列get-relational-database-events範例顯示指定關聯式資料庫過去 17 小時 (1020 分鐘） 內事件的詳細資訊。

```
aws lightsail get-relational-database-events \
    --relational-database-name Database-1 \
    --duration-in-minutes 1020
```

輸出：

```
{
    "relationalDatabaseEvents": [
        {
```

```
            "resource": "Database-1",
            "createdAt": 1571654146.553,
            "message": "Backing up Relational Database",
            "eventCategories": [
                "backup"
            ]
        },
        {
            "resource": "Database-1",
            "createdAt": 1571654249.98,
            "message": "Finished Relational Database backup",
            "eventCategories": [
                "backup"
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRelationalDatabaseEvents。

get-relational-database-log-events

以下程式碼範例顯示如何使用 get-relational-database-log-events。

AWS CLI

取得關聯式資料庫的日誌事件

下列get-relational-database-log-events範例顯示1571597176關聯式資料庫 的1570733176和 之間指定日誌的詳細資訊Database1。傳回的資訊設定為從 開始head。

我們建議您使用 unix 時間轉換器來識別開始和結束時間。

```
aws lightsail get-relational-database-log-events \
    --relational-database-name Database1 \
    --log-stream-name error \
    --start-from-head \
    --start-time 1570733176 \
    --end-time 1571597176
```

輸出：

```
{
```

```
   "resourceLogEvents": [
       {
           "createdAt": 1570820267.0,
           "message": "2019-10-11 18:57:47 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Name or service not known"
       },
       {
           "createdAt": 1570860974.0,
           "message": "2019-10-12 06:16:14 20969 [Warning] IP address '8192.0.2.0'
could not be resolved: Temporary failure in name resolution"
       },
       {
           "createdAt": 1570860977.0,
           "message": "2019-10-12 06:16:17 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Temporary failure in name resolution"
       },
       {
           "createdAt": 1570860979.0,
           "message": "2019-10-12 06:16:19 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Temporary failure in name resolution"
       },
       {
           "createdAt": 1570860981.0,
           "message": "2019-10-12 06:16:21 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Temporary failure in name resolution"
       },
       {
           "createdAt": 1570860982.0,
           "message": "2019-10-12 06:16:22 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Temporary failure in name resolution"
       },
       {
           "createdAt": 1570860984.0,
           "message": "2019-10-12 06:16:24 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Temporary failure in name resolution"
       },
       {
           "createdAt": 1570860986.0,
           "message": "2019-10-12 06:16:26 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Temporary failure in name resolution"
       },
       ...
       }
   ],
```

```
    "nextBackwardToken":
 "eEXAMPLEZXJUZXh0IjoiZnRWb3F3cUpRSlQ5NndMYThxelRUZlFhR3J6c2dKWEEvM2kvajZMZzVVVWpqRDN0YjFXTj
    "nextForwardToken":
 "eEXAMPLEZXJUZXh0IjoiT09Lb0Z6ZFRJbHhaNEQ5N2tPbkkwRmwwNUxPZjFTbFFwUklQbzlSaWgvMWVXbEk4aG56VH
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRelationalDatabaseLogEvents。

get-relational-database-log-streams

以下程式碼範例顯示如何使用 get-relational-database-log-streams。

AWS CLI

取得關聯式資料庫的日誌串流

下列get-relational-database-log-streams範例會傳回指定關聯式資料庫的所有可用日誌串流。

```
aws lightsail get-relational-database-log-streams \
--relational-database-name Database1
```

輸出：

```
{
    "logStreams": [
        "audit",
        "error",
        "general",
        "slowquery"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRelationalDatabaseLogStreams。

get-relational-database-master-user-password

以下程式碼範例顯示如何使用 get-relational-database-master-user-password。

AWS CLI

取得關聯式資料庫的主要使用者密碼

下列get-relational-database-master-user-password範例會傳回指定關聯式資料庫之主要使用者密碼的相關資訊。

```
aws lightsail get-relational-database-master-user-password \
    --relational-database-name Database-1
```

輸出：

```
{
    "masterUserPassword": "VEXAMPLEec.9qvx,_t<)Wkf)kwboM,>2",
    "createdAt": 1571259453.959
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRelationalDatabaseMasterUserPassword。

get-relational-database-metric-data

以下程式碼範例顯示如何使用 get-relational-database-metric-data。

AWS CLI

取得關聯式資料庫的指標資料

下列get-relational-database-metric-data範例會傳回 1570733176與1571597176關聯式資料庫 之間 24 小時 (86400 秒） DatabaseConnections期間的指標計數總和Database1。

我們建議您使用 unix 時間轉換器來識別開始和結束時間。

```
aws lightsail get-relational-database-metric-data \
    --relational-database-name Database1 \
    --metric-name DatabaseConnections \
    --period 86400 \
    --start-time 1570733176 \
    --end-time 1571597176 \
    --unit Count \
    --statistics Sum
```

輸出：

```
{
```

```
    "metricName": "DatabaseConnections",
    "metricData": [
        {
            "sum": 1.0,
            "timestamp": 1571510760.0,
            "unit": "Count"
        },
        {
            "sum": 1.0,
            "timestamp": 1570733160.0,
            "unit": "Count"
        },
        {
            "sum": 1.0,
            "timestamp": 1570992360.0,
            "unit": "Count"
        },
        {
            "sum": 0.0,
            "timestamp": 1571251560.0,
            "unit": "Count"
        },
        {
            "sum": 721.0,
            "timestamp": 1570819560.0,
            "unit": "Count"
        },
        {
            "sum": 1.0,
            "timestamp": 1571078760.0,
            "unit": "Count"
        },
        {
            "sum": 2.0,
            "timestamp": 1571337960.0,
            "unit": "Count"
        },
        {
            "sum": 684.0,
            "timestamp": 1570905960.0,
            "unit": "Count"
        },
        {
            "sum": 0.0,
```

```
            "timestamp": 1571165160.0,
            "unit": "Count"
        },
        {
            "sum": 1.0,
            "timestamp": 1571424360.0,
            "unit": "Count"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRelationalDatabaseMetricData。

get-relational-database-parameters

以下程式碼範例顯示如何使用 get-relational-database-parameters。

AWS CLI

取得關聯式資料庫的參數

下列get-relational-database-parameters範例會傳回指定關聯式資料庫所有可用參數的相關資訊。

```
aws lightsail get-relational-database-parameters \
    --relational-database-name Database-1
```

輸出：

```
{
    "parameters": [
        {
            "allowedValues": "0,1",
            "applyMethod": "pending-reboot",
            "applyType": "dynamic",
            "dataType": "boolean",
            "description": "Automatically set all granted roles as active after the
 user has authenticated successfully.",
            "isModifiable": true,
            "parameterName": "activate_all_roles_on_login",
            "parameterValue": "0"
```

```
        },
        {
            "allowedValues": "0,1",
            "applyMethod": "pending-reboot",
            "applyType": "static",
            "dataType": "boolean",
            "description": "Controls whether user-defined functions that have only
 an xxx symbol for the main function can be loaded",
            "isModifiable": false,
            "parameterName": "allow-suspicious-udfs"
        },
        {
            "allowedValues": "0,1",
            "applyMethod": "pending-reboot",
            "applyType": "dynamic",
            "dataType": "boolean",
            "description": "Sets the autocommit mode",
            "isModifiable": true,
            "parameterName": "autocommit"
        },
        {
            "allowedValues": "0,1",
            "applyMethod": "pending-reboot",
            "applyType": "static",
            "dataType": "boolean",
            "description": "Controls whether the server autogenerates SSL key and
 certificate files in the data directory, if they do not already exist.",
            "isModifiable": false,
            "parameterName": "auto_generate_certs"
        },
        ...
        }
    ]
}
```

如需詳細資訊，請參閱《 Lightsail 開發指南》中的在 中更新資料庫參數Amazon Lightsail。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRelationalDatabaseParameters。

get-relational-database-snapshot

以下程式碼範例顯示如何使用 get-relational-database-snapshot。

AWS CLI

取得關聯式資料庫快照的相關資訊

下列get-relational-database-snapshot範例顯示指定關聯式資料庫快照的詳細資訊。

```
aws lightsail get-relational-database-snapshot \
    --relational-database-snapshot-name Database-1-1571350042
```

輸出：

```
{
    "relationalDatabaseSnapshot": {
        "name": "Database-1-1571350042",
        "arn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabaseSnapshot/0389bbad-4b85-4c3d-9EXAMPLEaee3643d2",
        "supportCode": "6EXAMPLE3362/ls-8EXAMPLE2ba7ad041451946fafc2ad19cfbd9eb2",
        "createdAt": 1571350046.238,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "resourceType": "RelationalDatabaseSnapshot",
        "tags": [],
        "engine": "mysql",
        "engineVersion": "8.0.16",
        "sizeInGb": 40,
        "state": "available",
        "fromRelationalDatabaseName": "Database-1",
        "fromRelationalDatabaseArn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabase/7ea932b1-b85a-4bd5-9b3e-bEXAMPLE8cc4",
        "fromRelationalDatabaseBundleId": "micro_1_0",
        "fromRelationalDatabaseBlueprintId": "mysql_8_0"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRelationalDatabaseSnapshot。

get-relational-database-snapshots

以下程式碼範例顯示如何使用 get-relational-database-snapshots。

AWS CLI

取得所有關聯式資料庫快照的相關資訊

下列get-relational-database-snapshots範例顯示所設定 AWS 區域中所有關聯式資料庫快照的詳細資訊。

```
aws lightsail get-relational-database-snapshots
```

輸出：

```
{
    "relationalDatabaseSnapshots": [
        {
            "name": "Database-1-1571350042",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabaseSnapshot/0389bbad-4b85-4c3d-9861-6EXAMPLE43d2",
            "supportCode": "6EXAMPLE3362/
ls-8EXAMPLE2ba7ad041451946fafc2ad19cfbd9eb2",
            "createdAt": 1571350046.238,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "RelationalDatabaseSnapshot",
            "tags": [],
            "engine": "mysql",
            "engineVersion": "8.0.16",
            "sizeInGb": 40,
            "state": "available",
            "fromRelationalDatabaseName": "Database-1",
            "fromRelationalDatabaseArn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabase/7ea932b1-b85a-4bd5-9b3e-bEXAMPLE8cc4",
            "fromRelationalDatabaseBundleId": "micro_1_0",
            "fromRelationalDatabaseBlueprintId": "mysql_8_0"
        },
        {
            "name": "Database1-Console",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabaseSnapshot/8b94136e-06ec-4b1a-
a3fb-5EXAMPLEe1e9",
            "supportCode": "6EXAMPLE3362/
ls-9EXAMPLE14b000d34c8d1c432734e137612d5b5c",
```

```
            "createdAt": 1571249981.025,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "RelationalDatabaseSnapshot",
            "tags": [
                {
                    "key": "test"
                }
            ],
            "engine": "mysql",
            "engineVersion": "5.6.44",
            "sizeInGb": 40,
            "state": "available",
            "fromRelationalDatabaseName": "Database1",
            "fromRelationalDatabaseArn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabase/a6161cb7-4535-4f16-9dcf-8EXAMPLE3d4e",
            "fromRelationalDatabaseBundleId": "micro_1_0",
            "fromRelationalDatabaseBlueprintId": "mysql_5_6"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRelationalDatabaseSnapshots。

get-relational-database

以下程式碼範例顯示如何使用 get-relational-database。

AWS CLI

取得關聯式資料庫的相關資訊

下列get-relational-database範例顯示指定關聯式資料庫的詳細資訊。

```
aws lightsail get-relational-database \
    --relational-database-name Database-1
```

輸出：

```
{
    "relationalDatabase": {
```

```
        "name": "Database-1",
        "arn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabase/7ea932b1-b85a-4bd5-9b3e-bEXAMPLE8cc4",
        "supportCode": "6EXAMPLE3362/ls-9EXAMPLE8ad863723b62cc8901a8aa6e794ae0d2",
        "createdAt": 1571259453.795,
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "resourceType": "RelationalDatabase",
        "tags": [],
        "relationalDatabaseBlueprintId": "mysql_8_0",
        "relationalDatabaseBundleId": "micro_1_0",
        "masterDatabaseName": "dbmaster",
        "hardware": {
            "cpuCount": 1,
            "diskSizeInGb": 40,
            "ramSizeInGb": 1.0
        },
        "state": "available",
        "backupRetentionEnabled": false,
        "pendingModifiedValues": {},
        "engine": "mysql",
        "engineVersion": "8.0.16",
        "masterUsername": "dbmasteruser",
        "parameterApplyStatus": "in-sync",
        "preferredBackupWindow": "10:01-10:31",
        "preferredMaintenanceWindow": "sat:11:14-sat:11:44",
        "publiclyAccessible": true,
        "masterEndpoint": {
            "port": 3306,
            "address": "ls-9EXAMPLE8ad863723b62ccEXAMPLEa6e794ae0d2.czowadgeezqi.us-
west-2.rds.amazonaws.com"
        },
        "pendingMaintenanceActions": []
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRelationalDatabase。

get-relational-databases

以下程式碼範例顯示如何使用 get-relational-databases。

AWS CLI

取得所有關聯式資料庫的相關資訊

下列get-relational-databases範例顯示已設定 AWS 區域中所有關聯式資料庫的詳細資訊。

```
aws lightsail get-relational-databases
```

輸出：

```
{
    "relationalDatabases": [
        {
            "name": "MySQL",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabase/8529020c-3ab9-4d51-92af-5EXAMPLE8979",
            "supportCode": "6EXAMPLE3362/
ls-3EXAMPLEa995d8c3b06b4501356e5f2f28e1aeba",
            "createdAt": 1554306019.155,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "resourceType": "RelationalDatabase",
            "tags": [],
            "relationalDatabaseBlueprintId": "mysql_8_0",
            "relationalDatabaseBundleId": "micro_1_0",
            "masterDatabaseName": "dbmaster",
            "hardware": {
                "cpuCount": 1,
                "diskSizeInGb": 40,
                "ramSizeInGb": 1.0
            },
            "state": "available",
            "backupRetentionEnabled": true,
            "pendingModifiedValues": {},
            "engine": "mysql",
            "engineVersion": "8.0.15",
            "latestRestorableTime": 1571686200.0,
            "masterUsername": "dbmasteruser",
            "parameterApplyStatus": "in-sync",
            "preferredBackupWindow": "07:51-08:21",
            "preferredMaintenanceWindow": "tue:12:18-tue:12:48",
```

```
            "publiclyAccessible": true,
            "masterEndpoint": {
                "port": 3306,
                "address":
 "ls-3EXAMPLEa995d8c3b06b4501356e5f2fEXAMPLEa.czowadgeezqi.us-
west-2.rds.amazonaws.com"
            },
            "pendingMaintenanceActions": []
        },
        {
            "name": "Postgres",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabase/
e9780b6b-d0ab-4af2-85f1-1EXAMPLEac68",
            "supportCode": "6EXAMPLE3362/
ls-3EXAMPLEb4fffb5cec056220c734713e14bd5fcd",
            "createdAt": 1554306000.814,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "resourceType": "RelationalDatabase",
            "tags": [],
            "relationalDatabaseBlueprintId": "postgres_11",
            "relationalDatabaseBundleId": "micro_1_0",
            "masterDatabaseName": "dbmaster",
            "hardware": {
                "cpuCount": 1,
                "diskSizeInGb": 40,
                "ramSizeInGb": 1.0
            },
            "state": "available",
            "backupRetentionEnabled": true,
            "pendingModifiedValues": {},
            "engine": "postgres",
            "engineVersion": "11.1",
            "latestRestorableTime": 1571686339.0,
            "masterUsername": "dbmasteruser",
            "parameterApplyStatus": "in-sync",
            "preferredBackupWindow": "06:19-06:49",
            "preferredMaintenanceWindow": "sun:10:19-sun:10:49",
            "publiclyAccessible": false,
            "masterEndpoint": {
                "port": 5432,
```

```
                "address":
 "ls-3EXAMPLEb4fffb5cec056220c734713eEXAMPLEd.czowadgeezqi.us-
west-2.rds.amazonaws.com"
            },
            "pendingMaintenanceActions": []
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRelationalDatabases。

get-static-ip

以下程式碼範例顯示如何使用 get-static-ip。

AWS CLI

取得靜態 IP 的相關資訊

下列get-static-ip範例顯示指定靜態 IP 的詳細資訊。

```
aws lightsail get-static-ip \
    --static-ip-name StaticIp-1
```

輸出：

```
{
    "staticIp": {
        "name": "StaticIp-1",
        "arn": "arn:aws:lightsail:us-
west-2:111122223333:StaticIp/2257cd76-1f0e-4ac0-82e2-2EXAMPLE23ad",
        "supportCode": "6EXAMPLE3362/192.0.2.0",
        "createdAt": 1571071325.076,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "resourceType": "StaticIp",
        "ipAddress": "192.0.2.0",
        "isAttached": false
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetStaticIp。

get-static-ips

以下程式碼範例顯示如何使用 get-static-ips。

AWS CLI

取得所有靜態 IPs 的相關資訊

下列get-static-ips範例顯示所設定 AWS 區域中所有靜態 IPs 的詳細資訊。

```
aws lightsail get-static-ips
```

輸出：

```
{
    "staticIps": [
        {
            "name": "StaticIp-1",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:StaticIp/2257cd76-1f0e-4ac0-8EXAMPLE16f9423ad",
            "supportCode": "6EXAMPLE3362/192.0.2.0",
            "createdAt": 1571071325.076,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "StaticIp",
            "ipAddress": "192.0.2.0",
            "isAttached": false
        },
        {
            "name": "StaticIP-2",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:StaticIp/c61edb40-
e5f0-4fd6-ae7c-8EXAMPLE19f8",
            "supportCode": "6EXAMPLE3362/192.0.2.2",
            "createdAt": 1568305385.681,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
```

```
            "resourceType": "StaticIp",
            "ipAddress": "192.0.2.2",
            "attachedTo": "WordPress-1",
            "isAttached": true
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetStaticIps。

is-vpc-peered

以下程式碼範例顯示如何使用 is-vpc-peered。

AWS CLI

識別您的Amazon Lightsail虛擬私有雲端是否對等互連

下列is-vpc-peered範例會傳回指定 AWS 區域的Amazon Lightsail虛擬私有雲端 (VPC) 對等互連狀態。

```
aws lightsail is-vpc-peered \
    --region us-west-2
```

輸出：

```
{
    "isPeered": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 IsVpcPeered。

open-instance-public-ports

以下程式碼範例顯示如何使用 open-instance-public-ports。

AWS CLI

開啟執行個體的防火牆連接埠

下列open-instance-public-ports範例會在指定的執行個體上開啟 TCP 連接埠 22。

```
aws lightsail open-instance-public-ports \
    --instance-name MEAN-2 \
    --port-info fromPort=22,protocol=TCP,toPort=22
```

輸出：

```
{
    "operation": {
        "id": "719744f0-a022-46f2-9f11-6EXAMPLE4642",
        "resourceName": "MEAN-2",
        "resourceType": "Instance",
        "createdAt": 1571072906.849,
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "isTerminal": true,
        "operationDetails": "22/tcp",
        "operationType": "OpenInstancePublicPorts",
        "status": "Succeeded",
        "statusChangedAt": 1571072906.849
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 OpenInstancePublicPorts。

peer-vpc

以下程式碼範例顯示如何使用 peer-vpc。

AWS CLI

對等Amazon Lightsail虛擬私有雲端

下列peer-vpc範例會比對指定 AWS 區域的Amazon Lightsail虛擬私有雲端 (VPC)。

```
aws lightsail peer-vpc \
    --region us-west-2
```

輸出：

```
{
```

```
    "operation": {
        "id": "787e846a-54ac-497f-bce2-9EXAMPLE5d91",
        "resourceName": "vpc-0EXAMPLEa5261efb3",
        "resourceType": "PeeredVpc",
        "createdAt": 1571694233.104,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "isTerminal": true,
        "operationDetails": "vpc-e2b3eb9b",
        "operationType": "PeeredVpc",
        "status": "Succeeded",
        "statusChangedAt": 1571694233.104
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PeerVpc。

reboot-instance

以下程式碼範例顯示如何使用 reboot-instance。

AWS CLI

重新啟動執行個體

下列reboot-instance範例會重新啟動指定的執行個體。

```
aws lightsail reboot-instance \
    --instance-name MEAN-1
```

輸出：

```
{
    "operations": [
        {
            "id": "2b679f1c-8b71-4bb4-8e97-8EXAMPLEed93",
            "resourceName": "MEAN-1",
            "resourceType": "Instance",
            "createdAt": 1571694445.49,
            "location": {
```

```
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "",
            "operationType": "RebootInstance",
            "status": "Succeeded",
            "statusChangedAt": 1571694445.49
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RebootInstance。

reboot-relational-database

以下程式碼範例顯示如何使用 reboot-relational-database。

AWS CLI

重新啟動關聯式資料庫

下列reboot-relational-database範例會重新啟動指定的關聯式資料庫。

```
aws lightsail reboot-relational-database \
    --relational-database-name Database-1
```

輸出：

```
{
    "operations": [
        {
            "id": "e4c980c0-3137-496c-9c91-1EXAMPLEdec2",
            "resourceName": "Database-1",
            "resourceType": "RelationalDatabase",
            "createdAt": 1571694532.91,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "",
```

```
            "operationType": "RebootRelationalDatabase",
            "status": "Started",
            "statusChangedAt": 1571694532.91
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RebootRelationalDatabase。

release-static-ip

以下程式碼範例顯示如何使用 release-static-ip。

AWS CLI

刪除靜態 IP

下列release-static-ip範例會刪除指定的靜態 IP。

```
aws lightsail release-static-ip \
    --static-ip-name StaticIp-1
```

輸出：

```
{
    "operations": [
        {
            "id": "e374c002-dc6d-4c7f-919f-2EXAMPLE13ce",
            "resourceName": "StaticIp-1",
            "resourceType": "StaticIp",
            "createdAt": 1571694962.003,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "ReleaseStaticIp",
            "status": "Succeeded",
            "statusChangedAt": 1571694962.003
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReleaseStaticIp。

start-instance

以下程式碼範例顯示如何使用 start-instance。

AWS CLI

啟動執行個體

下列start-instance範例會啟動指定的執行個體。

```
aws lightsail start-instance \
    --instance-name WordPress-1
```

輸出：

```
{
    "operations": [
        {
            "id": "f88d2a93-7cea-4165-afce-2d688cb18f23",
            "resourceName": "WordPress-1",
            "resourceType": "Instance",
            "createdAt": 1571695583.463,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "StartInstance",
            "status": "Started",
            "statusChangedAt": 1571695583.463
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartInstance。

start-relational-database

以下程式碼範例顯示如何使用 start-relational-database。

AWS CLI

啟動關聯式資料庫

下列start-relational-database範例會啟動指定的關聯式資料庫。

```
aws lightsail start-relational-database \
    --relational-database-name Database-1
```

輸出：

```
{
    "operations": [
        {
            "id": "4d5294ec-a38a-4fda-9e37-aEXAMPLE0d24",
            "resourceName": "Database-1",
            "resourceType": "RelationalDatabase",
            "createdAt": 1571695998.822,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "StartRelationalDatabase",
            "status": "Started",
            "statusChangedAt": 1571695998.822
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartRelationalDatabase。

stop-instance

以下程式碼範例顯示如何使用 stop-instance。

AWS CLI

停止執行個體

下列stop-instance範例會停止指定的執行個體。

```
aws lightsail stop-instance \
```

```
--instance-name WordPress-1
```

輸出：

```
{
    "operations": [
        {
            "id": "265357e2-2943-4d51-888a-1EXAMPLE7585",
            "resourceName": "WordPress-1",
            "resourceType": "Instance",
            "createdAt": 1571695471.134,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "StopInstance",
            "status": "Started",
            "statusChangedAt": 1571695471.134
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopInstance。

stop-relational-database

以下程式碼範例顯示如何使用 stop-relational-database。

AWS CLI

停止關聯式資料庫

下列stop-relational-database範例會停止指定的關聯式資料庫。

```
aws lightsail stop-relational-database \
    --relational-database-name Database-1
```

輸出：

```
{
```

```
    "operations": [
        {
            "id": "cc559c19-4adb-41e4-b75b-5EXAMPLE4e61",
            "resourceName": "Database-1",
            "resourceType": "RelationalDatabase",
            "createdAt": 1571695526.29,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "StopRelationalDatabase",
            "status": "Started",
            "statusChangedAt": 1571695526.29
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopRelationalDatabase。

unpeer-vpc

以下程式碼範例顯示如何使用 unpeer-vpc。

AWS CLI

取消對等Amazon Lightsail虛擬私有雲端

下列unpeer-vpc範例會取消對指定 AWS 區域的Amazon Lightsail虛擬私有雲端 (VPC)。

```
aws lightsail unpeer-vpc \
    --region us-west-2
```

輸出：

```
{
    "operation": {
        "id": "531aca64-7157-47ab-84c6-eEXAMPLEd898",
        "resourceName": "vpc-0EXAMPLEa5261efb3",
        "resourceType": "PeeredVpc",
        "createdAt": 1571694109.945,
        "location": {
```

```
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "isTerminal": true,
        "operationDetails": "vpc-e2b3eb9b",
        "operationType": "UnpeeredVpc",
        "status": "Succeeded",
        "statusChangedAt": 1571694109.945
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UnpeerVpc。

使用 的 Macie 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Macie 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`describe-buckets`

以下程式碼範例顯示如何使用 `describe-buckets`。

AWS CLI

查詢 Amazon Macie 為您的帳戶監控和分析的一或多個 S3 儲存貯體的資料

下列`describe-buckets`範例會查詢名稱開頭為 amzn-s3-demo-bucket 且位於目前區域之所有 S3 儲存貯體的中繼資料。 AWS

```
aws macie2 describe-buckets \
    --criteria '{"bucketName":{"prefix":"amzn-s3-demo-bucket"}}'
```

輸出：


```
{
    "buckets": [
        {
            "accountId": "123456789012",
            "allowsUnencryptedObjectUploads": "FALSE",
            "automatedDiscoveryMonitoringStatus": "MONITORED",
            "bucketArn": "arn:aws:s3:::amzn-s3-demo-bucket1",
            "bucketCreatedAt": "2020-05-18T19:54:00+00:00",
            "bucketName": "amzn-s3-demo-bucket1",
            "classifiableObjectCount": 13,
            "classifiableSizeInBytes": 1592088,
            "jobDetails": {
                "isDefinedInJob": "TRUE",
                "isMonitoredByJob": "TRUE",
                "lastJobId": "08c81dc4a2f3377fae45c9ddaEXAMPLE",
                "lastJobRunTime": "2024-08-19T14:55:30.270000+00:00"
            },
            "lastAutomatedDiscoveryTime": "2024-10-22T19:11:25.364000+00:00",
            "lastUpdated": "2024-10-25T07:33:06.337000+00:00",
            "objectCount": 13,
            "objectCountByEncryptionType": {
                "customerManaged": 0,
                "kmsManaged": 2,
                "s3Managed": 7,
                "unencrypted": 4,
                "unknown": 0
            },
            "publicAccess": {
                "effectivePermission": "NOT_PUBLIC",
                "permissionConfiguration": {
                    "accountLevelPermissions": {
                        "blockPublicAccess": {
                            "blockPublicAcls": true,
                            "blockPublicPolicy": true,
                            "ignorePublicAcls": true,
                            "restrictPublicBuckets": true
                        }
                    },
                    "bucketLevelPermissions": {
                        "accessControlList": {
                            "allowsPublicReadAccess": false,
                            "allowsPublicWriteAccess": false
```

```
                    },
                    "blockPublicAccess": {
                        "blockPublicAcls": true,
                        "blockPublicPolicy": true,
                        "ignorePublicAcls": true,
                        "restrictPublicBuckets": true
                    },
                    "bucketPolicy": {
                        "allowsPublicReadAccess": false,
                        "allowsPublicWriteAccess": false
                    }
                }
            }
        },
        "region": "us-west-2",
        "replicationDetails": {
            "replicated": false,
            "replicatedExternally": false,
            "replicationAccounts": []
        },
        "sensitivityScore": 78,
        "serverSideEncryption": {
            "kmsMasterKeyId": null,
            "type": "NONE"
        },
        "sharedAccess": "NOT_SHARED",
        "sizeInBytes": 4549746,
        "sizeInBytesCompressed": 0,
        "tags": [
            {
                "key": "Division",
                "value": "HR"
            },
            {
                "key": "Team",
                "value": "Recruiting"
            }
        ],
        "unclassifiableObjectCount": {
            "fileType": 0,
            "storageClass": 0,
            "total": 0
        },
        "unclassifiableObjectSizeInBytes": {
```

```
                "fileType": 0,
                "storageClass": 0,
                "total": 0
            },
            "versioning": true
        },
        {
            "accountId": "123456789012",
            "allowsUnencryptedObjectUploads": "TRUE",
            "automatedDiscoveryMonitoringStatus": "MONITORED",
            "bucketArn": "arn:aws:s3:::amzn-s3-demo-bucket2",
            "bucketCreatedAt": "2020-11-25T18:24:38+00:00",
            "bucketName": "amzn-s3-demo-bucket2",
            "classifiableObjectCount": 8,
            "classifiableSizeInBytes": 133810,
            "jobDetails": {
                "isDefinedInJob": "TRUE",
                "isMonitoredByJob": "FALSE",
                "lastJobId": "188d4f6044d621771ef7d65f2EXAMPLE",
                "lastJobRunTime": "2024-07-09T19:37:11.511000+00:00"
            },
            "lastAutomatedDiscoveryTime": "2024-10-24T19:11:25.364000+00:00",
            "lastUpdated": "2024-10-25T07:33:06.337000+00:00",
            "objectCount": 8,
            "objectCountByEncryptionType": {
                "customerManaged": 0,
                "kmsManaged": 0,
                "s3Managed": 8,
                "unencrypted": 0,
                "unknown": 0
            },
            "publicAccess": {
                "effectivePermission": "NOT_PUBLIC",
                "permissionConfiguration": {
                    "accountLevelPermissions": {
                        "blockPublicAccess": {
                            "blockPublicAcls": true,
                            "blockPublicPolicy": true,
                            "ignorePublicAcls": true,
                            "restrictPublicBuckets": true
                        }
                    },
                    "bucketLevelPermissions": {
                        "accessControlList": {
```

```
                        "allowsPublicReadAccess": false,
                        "allowsPublicWriteAccess": false
                    },
                    "blockPublicAccess": {
                        "blockPublicAcls": true,
                        "blockPublicPolicy": true,
                        "ignorePublicAcls": true,
                        "restrictPublicBuckets": true
                    },
                    "bucketPolicy": {
                        "allowsPublicReadAccess": false,
                        "allowsPublicWriteAccess": false
                    }
                }
            }
        },
        "region": "us-west-2",
        "replicationDetails": {
            "replicated": false,
            "replicatedExternally": false,
            "replicationAccounts": []
        },
        "sensitivityScore": 95,
        "serverSideEncryption": {
            "kmsMasterKeyId": null,
            "type": "AES256"
        },
        "sharedAccess": "EXTERNAL",
        "sizeInBytes": 175978,
        "sizeInBytesCompressed": 0,
        "tags": [
            {
                "key": "Division",
                "value": "HR"
            },
            {
                "key": "Team",
                "value": "Recruiting"
            }
        ],
        "unclassifiableObjectCount": {
            "fileType": 3,
            "storageClass": 0,
            "total": 3
```

```
            },
            "unclassifiableObjectSizeInBytes": {
                "fileType": 2999826,
                "storageClass": 0,
                "total": 2999826
            },
            "versioning": true
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Macie 使用者指南》中的篩選 S3 儲存貯體庫存。 Amazon Macie

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeBuckets。

使用 的 Amazon Managed Grafana 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon Managed Grafana 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`list-workspaces`

以下程式碼範例顯示如何使用 `list-workspaces`。

AWS CLI

在使用者登入資料指定的區域中列出帳戶的工作區

下列`list-workspaces`範例列出帳戶 區域的 Grafana 工作區。

```
aws grafana list-workspaces
```

輸出：

```
{
    "workspaces": [
        {
            "authentication": {
                "providers": [
                    "AWS_SSO"
                ]
            },
            "created": "2022-04-04T16:20:21.796000-07:00",
            "description": "to test tags",
            "endpoint": "g-949e7b44df.grafana-workspace.us-east-1.amazonaws.com",
            "grafanaVersion": "8.2",
            "id": "g-949e7b44df",
            "modified": "2022-04-04T16:20:21.796000-07:00",
            "name": "testtag2",
            "notificationDestinations": [
                "SNS"
            ],
            "status": "ACTIVE"
        },
        {
            "authentication": {
                "providers": [
                    "AWS_SSO"
                ]
            },
            "created": "2022-04-20T10:22:15.115000-07:00",
            "description": "ww",
            "endpoint": "g-bffa51ed1b.grafana-workspace.us-east-1.amazonaws.com",
            "grafanaVersion": "8.2",
            "id": "g-bffa51ed1b",
            "modified": "2022-04-20T10:22:15.115000-07:00",
            "name": "ww",
            "notificationDestinations": [
                "SNS"
            ],
            "status": "ACTIVE"
        }
    ]
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListWorkspaces。

使用 的 MediaConnect 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 MediaConnect 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`add-flow-outputs`

以下程式碼範例顯示如何使用 `add-flow-outputs`。

AWS CLI

將輸出新增至流程

下列`add-flow-outputs`範例會將輸出新增至指定的流程。

```
aws mediaconnect add-flow-outputs \
--flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
--outputs Description='NYC
 stream',Destination=192.0.2.12,Name=NYC,Port=3333,Protocol=rtp-
fec,SmoothingLatency=100 Description='LA
 stream',Destination=203.0.113.9,Name=LA,Port=4444,Protocol=rtp-
fec,SmoothingLatency=100
```

輸出：

```
{
```

```
    "Outputs": [
        {
            "Port": 3333,
            "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC",
            "Name": "NYC",
            "Description": "NYC stream",
            "Destination": "192.0.2.12",
            "Transport": {
                "Protocol": "rtp-fec",
                "SmoothingLatency": 100
            }
        },
        {
            "Port": 4444,
            "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-987655dEF67hiJ89-c34de5fG678h:LA",
            "Name": "LA",
            "Description": "LA stream",
            "Destination": "203.0.113.9",
            "Transport": {
                "Protocol": "rtp-fec",
                "SmoothingLatency": 100
            }
        }
    ],
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame"
}
```

如需詳細資訊，請參閱 AWS Elemental MediaConnect 使用者指南中的將輸出新增至流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddFlowOutputs。

create-flow

以下程式碼範例顯示如何使用 create-flow。

AWS CLI

建立流程

下列create-flow範例會建立具有指定組態的流程。

```
aws mediaconnect create-flow \
    --availability-zone us-west-2c \
    --name ExampleFlow \
    --source Description='Example source,
 backup',IngestPort=1055,Name=BackupSource,Protocol=rtp,WhitelistCidr=10.24.34.0/23
```

輸出：

```
{
    "Flow": {
        "FlowArn": "arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:ExampleFlow",
        "AvailabilityZone": "us-west-2c",
        "EgressIp": "54.245.71.21",
        "Source": {
            "IngestPort": 1055,
            "SourceArn": "arn:aws:mediaconnect:us-
east-1:123456789012:source:2-3aBC45dEF67hiJ89-c34de5fG678h:BackupSource",
            "Transport": {
                "Protocol": "rtp",
                "MaxBitrate": 80000000
            },
            "Description": "Example source, backup",
            "IngestIp": "54.245.71.21",
            "WhitelistCidr": "10.24.34.0/23",
            "Name": "mySource"
        },
        "Entitlements": [],
        "Name": "ExampleFlow",
        "Outputs": [],
        "Status": "STANDBY",
        "Description": "Example source, backup"
    }
}
```

如需詳細資訊，請參閱 AWS Elemental MediaConnect 使用者指南中的建立流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFlow。

delete-flow

以下程式碼範例顯示如何使用 delete-flow。

AWS CLI

刪除流程

下列delete-flow範例會刪除指定的流程。

```
aws mediaconnect delete-flow \
    --flow-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

輸出：

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
    "Status": "DELETING"
}
```

如需詳細資訊，請參閱 AWS Elemental MediaConnect 使用者指南中的刪除流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFlow。

describe-flow

以下程式碼範例顯示如何使用 describe-flow。

AWS CLI

檢視流程的詳細資訊

下列describe-flow範例顯示指定流程的詳細資訊，例如 ARN、可用區域、狀態、來源、權利和輸出。

```
aws mediaconnect describe-flow \
    --flow-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

輸出：

```
{
    "Flow": {
        "EgressIp": "54.201.4.39",
```

```
        "AvailabilityZone": "us-west-2c",
        "Status": "ACTIVE",
        "FlowArn": "arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
        "Entitlements": [
            {
                "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:123456789012:entitlement:1-AaBb11CcDd22EeFf-34DE5fG12AbC:MyEntitlement",
                "Description": "Assign to this account",
                "Name": "MyEntitlement",
                "Subscribers": [
                    "444455556666"
                ]
            }
        ],
        "Description": "NYC awards show",
        "Name": "AwardsShow",
        "Outputs": [
            {
                "Port": 2355,
                "Name": "NYC",
                "Transport": {
                    "SmoothingLatency": 0,
                    "Protocol": "rtp-fec"
                },
                "OutputArn": "arn:aws:mediaconnect:us-
east-1:123456789012:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC",
                "Destination": "192.0.2.0"
            },
            {
                "Port": 3025,
                "Name": "LA",
                "Transport": {
                    "SmoothingLatency": 0,
                    "Protocol": "rtp-fec"
                },
                "OutputArn": "arn:aws:mediaconnect:us-
east-1:123456789012:output:2-987655dEF67hiJ89-c34de5fG678h:LA",
                "Destination": "192.0.2.0"
            }
        ],
        "Source": {
            "IngestIp": "54.201.4.39",
```

```
            "SourceArn": "arn:aws:mediaconnect:us-
east-1:123456789012:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource",
            "Transport": {
                "MaxBitrate": 80000000,
                "Protocol": "rtp"
            },
            "IngestPort": 1069,
            "Description": "Saturday night show",
            "Name": "ShowSource",
            "WhitelistCidr": "10.24.34.0/23"
        }
    }
}
```

如需詳細資訊，請參閱 AWS Elemental MediaConnect 使用者指南中的檢視流程的詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFlow。

grant-flow-entitlements

以下程式碼範例顯示如何使用 grant-flow-entitlements。

AWS CLI

授予流程的權限

下列grant-flow-entitlements範例授予指定現有流程的權限，以與其他 AWS 帳戶共用您的內容。

```
aws mediaconnect grant-flow-entitlements \
    --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
    --entitlements Description='For
 AnyCompany',Encryption={"Algorithm=aes128,KeyType=static-
key,RoleArn=arn:aws:iam::111122223333:role/MediaConnect-
ASM,SecretArn=arn:aws:secretsmanager:us-
west-2:111122223333:secret:mySecret1"},Name=AnyCompany_Entitlement,Subscribers=444455556666
 Description='For Example Corp',Name=ExampleCorp,Subscribers=777788889999
```

輸出：

```
{
```

```
    "Entitlements": [
        {
            "Name": "AnyCompany_Entitlement",
            "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement",
            "Subscribers": [
                "444455556666"
            ],
            "Description": "For AnyCompany",
            "Encryption": {
                "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret:mySecret1",
                "Algorithm": "aes128",
                "RoleArn": "arn:aws:iam::111122223333:role/MediaConnect-ASM",
                "KeyType": "static-key"
            }
        },
        {
            "Name": "ExampleCorp",
            "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-3333cccc4444dddd-1111aaaa2222:ExampleCorp",
            "Subscribers": [
                "777788889999"
            ],
            "Description": "For Example Corp"
        }
    ],
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame"
}
```

如需詳細資訊，請參閱 AWS Elemental MediaConnect 使用者指南中的授予流程的權限。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GrantFlowEntitlements。

`list-entitlements`

以下程式碼範例顯示如何使用 `list-entitlements`。

AWS CLI

檢視權利清單

下列`list-entitlements`範例顯示已授予帳戶的所有權利清單。

```
aws mediaconnect list-entitlements
```

輸出：

```
{
    "Entitlements": [
        {
            "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:MyEntitlement",
            "EntitlementName": "MyEntitlement"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Elemental MediaConnect API 參考中的 ListEntitlements。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEntitlements。

list-flows

以下程式碼範例顯示如何使用 list-flows。

AWS CLI

檢視流程清單

下列list-flows範例顯示流程清單。

```
aws mediaconnect list-flows
```

輸出：

```
{
    "Flows": [
        {
            "Status": "STANDBY",
            "SourceType": "OWNED",
            "AvailabilityZone": "us-west-2a",
            "Description": "NYC awards show",
            "Name": "AwardsShow",
            "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow"
```

```
        },
        {
            "Status": "STANDBY",
            "SourceType": "OWNED",
            "AvailabilityZone": "us-west-2c",
            "Description": "LA basketball game",
            "Name": "BasketballGame",
            "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Elemental MediaConnect 使用者指南中的檢視流程清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFlows。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出 MediaConnect 資源的標籤

下列`list-tags-for-resource`範例顯示與指定 MediaConnect 資源相關聯的標籤索引鍵和值。

```
aws mediaconnect list-tags-for-resource \
    --resource-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame
```

輸出：

```
{
    "Tags": {
        "region": "west",
        "stage": "prod"
    }
}
```

如需詳細資訊，請參閱 AWS Elemental MediaConnect API 參考中的 ListTagsForResource、TagResource、UntagResource。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

remove-flow-output

以下程式碼範例顯示如何使用 remove-flow-output。

AWS CLI

從流程移除輸出

下列remove-flow-output範例會從指定的流程移除輸出。

```
aws mediaconnect remove-flow-output \
    --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
    --output-arn arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC
```

輸出：

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
    "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC"
}
```

如需詳細資訊，請參閱 AWS Elemental MediaConnect 使用者指南中的從流程移除輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveFlowOutput。

revoke-flow-entitlement

以下程式碼範例顯示如何使用 revoke-flow-entitlement。

AWS CLI

撤銷 權利

下列revoke-flow-entitlement範例會撤銷指定流程的權限。

```
aws mediaconnect revoke-flow-entitlement \
```

```
    --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
    --entitlement-arn arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement
```

輸出：

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
    "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement"
}
```

如需詳細資訊，請參閱 AWS Elemental MediaConnect 使用者指南中的撤銷 權限。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RevokeFlowEntitlement。

start-flow

以下程式碼範例顯示如何使用 start-flow。

AWS CLI

啟動流程

下列start-flow範例會啟動指定的流程。

```
aws mediaconnect start-flow \
    --flow-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

此命令不會產生輸出。輸出：

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
    "Status": "STARTING"
}
```

如需詳細資訊，請參閱 AWS Elemental MediaConnect 使用者指南中的啟動流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartFlow。

stop-flow

以下程式碼範例顯示如何使用 stop-flow。

AWS CLI

停止流程

下列stop-flow範例會停止指定的流程。

```
aws mediaconnect stop-flow \
    --flow-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

輸出：

```
{
    "Status": "STOPPING",
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow"
}
```

如需詳細資訊，請參閱 AWS Elemental MediaConnect 使用者指南中的停止流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopFlow。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤新增至 MediaConnect 資源

下列tag-resource範例會將索引鍵名稱和值為 的標籤新增至指定的 MediaConnect 資源。

```
aws mediaconnect tag-resource \
    --resource-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame
    --tags region=west
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaConnect API 參考中的 ListTagsForResource、TagResource、UntagResource。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從 MediaConnect 資源移除標籤

下列untag-resource範例會從 MediaConnect 資源移除具有指定金鑰名稱及其相關聯值的標籤。

```
aws mediaconnect untag-resource \
    --resource-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame \
    --tag-keys region
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaConnect API 參考中的 ListTagsForResource、TagResource、UntagResource。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-flow-entitlement

以下程式碼範例顯示如何使用 update-flow-entitlement。

AWS CLI

更新權利

下列update-flow-entitlement範例會使用新的描述和訂閱者更新指定的權利。

```
aws mediaconnect update-flow-entitlement \
    --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
```

```
    --entitlement-arn arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement
 \
    --description 'For AnyCompany Affiliate' \
    --subscribers 777788889999
```

輸出：

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
    "Entitlement": {
        "Name": "AnyCompany_Entitlement",
        "Description": "For AnyCompany Affiliate",
        "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement",
        "Encryption": {
            "KeyType": "static-key",
            "Algorithm": "aes128",
            "RoleArn": "arn:aws:iam::111122223333:role/MediaConnect-ASM",
            "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret:mySecret1"
        },
        "Subscribers": [
            "777788889999"
        ]
    }
}
```

如需詳細資訊，請參閱 AWS Elemental MediaConnect 使用者指南中的更新 權限。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateFlowEntitlement。

update-flow-output

以下程式碼範例顯示如何使用 update-flow-output。

AWS CLI

更新流程的輸出

下列update-flow-output範例會更新指定流程的輸出。

```
aws mediaconnect update-flow-output \
    --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
    --output-arn arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC \
    --port 3331
```

輸出：

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
    "Output": {
        "Name": "NYC",
        "Port": 3331,
        "Description": "NYC stream",
        "Transport": {
            "Protocol": "rtp-fec",
            "SmoothingLatency": 100
        },
        "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC",
        "Destination": "192.0.2.12"
    }
}
```

如需詳細資訊，請參閱 AWS Elemental MediaConnect 使用者指南中的更新流程的輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateFlowOutput。

update-flow-source

以下程式碼範例顯示如何使用 update-flow-source。

AWS CLI

更新現有流程的來源

下列update-flow-source範例會更新現有流程的來源。

```
aws mediaconnect update-flow-source \
    --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow \
```

```
    --source-arn arn:aws:mediaconnect:us-
east-1:111122223333:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource \
    --description 'Friday night show' \
    --ingest-port 3344 \
    --protocol rtp-fec \
    --whitelist-cidr 10.24.34.0/23
```

輸出：

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
    "Source": {
        "IngestIp": "34.210.136.56",
        "WhitelistCidr": "10.24.34.0/23",
        "Transport": {
            "Protocol": "rtp-fec"
        },
        "IngestPort": 3344,
        "Name": "ShowSource",
        "Description": "Friday night show",
        "SourceArn": "arn:aws:mediaconnect:us-
east-1:111122223333:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource"
    }
}
```

如需詳細資訊，請參閱 AWS Elemental MediaConnect 使用者指南中的更新流程的來源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateFlowSource。

使用 的 MediaConvert 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 MediaConvert 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`cancel-job`

以下程式碼範例顯示如何使用 `cancel-job`。

AWS CLI

取消佇列中的任務

下列`cancel-job`範例會取消 ID 為 的任務`1234567891234-abc123`。您無法取消服務已開始處理的任務。

```
aws mediaconvert cancel-job \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
    --region region-name-1 \
    --id 1234567891234-abc123
```

若要取得您的帳戶特定端點，請使用 `describe-endpoints`，或傳送不含端點的 命令。服務會傳回錯誤和您的端點。

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的使用 Elemental MediaConvert 任務。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelJob。

`create-job-template`

以下程式碼範例顯示如何使用 `create-job-template`。

AWS CLI

建立作業範本

下列`create-job-template`範例會使用系統所在檔案中指定的轉碼設定`job-template.json`來建立任務範本。

```
aws mediaconvert create-job-template \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
    --region region-name-1 \
    --name JobTemplate1 \
    --cli-input-json file://~/job-template.json
```

如果您使用 建立任務範本 JSON 檔案，get-job-template然後修改檔案，請移除JobTemplate物件，但將設定子物件保留在其中。此外，請務必移除下列鍵值對：LastUpdated、Type、 Arn和 CreatedAt。您可以在 JSON 檔案或命令列中指定類別、描述、名稱和佇列。

若要取得您的帳戶特定端點，請使用 describe-endpoints，或傳送不含端點的 命令。服務會傳回錯誤和您的端點。

如果您的請求成功，服務會針對您建立的任務範本傳回 JSON 規格。

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的使用 Elemental MediaConvert 任務範本。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateJobTemplate。

create-job

以下程式碼範例顯示如何使用 create-job。

AWS CLI

建立任務

下列create-job範例會建立轉碼任務，其中包含檔案中所指定的設定job.json，該檔案位於您傳送命令的系統上。此 JSON 任務規格可能會個別指定每個設定、參考任務範本或參考輸出預設集。

```
aws mediaconvert create-job \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
    --region region-name-1 \
    --cli-input-json file://~/job.json
```

您可以使用 AWS Elemental MediaConvert 主控台來產生 JSON 任務規格，方法是選擇您的任務設定，然後選擇任務區段底部的顯示任務 JSON。

若要取得您的帳戶特定端點，請使用 describe-endpoints，或傳送不含端點的 命令。服務會傳回錯誤和您的端點。

如果您的請求成功，服務會傳回您隨請求傳送的 JSON 任務規格。

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的使用 Elemental MediaConvert 任務。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateJob。

create-preset

以下程式碼範例顯示如何使用 create-preset。

AWS CLI

建立自訂輸出預設集

下列create-preset範例會根據檔案 中指定的輸出設定建立自訂輸出預設集preset.json。您可以在 JSON 檔案或命令列指定類別、描述和名稱。

```
aws mediaconvert create-preset \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com
    --region region-name-1 \
    --cli-input-json file://~/preset.json
```

如果您使用 建立預設 JSON 檔案，get-preset然後修改輸出檔案，請務必移除下列鍵/值對：LastUpdated、Type、 Arn和 CreatedAt。

若要取得您的帳戶特定端點，請使用 describe-endpoints，或傳送不含端點的 命令。服務會傳回錯誤和您的端點。

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的使用 Elemental MediaConvert 輸出預設集。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePreset。

create-queue

以下程式碼範例顯示如何使用 create-queue。

AWS CLI

建立自訂佇列

下列create-queue範例會建立自訂轉碼佇列。

```
aws mediaconvert create-queue \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
```

```
    --region region-name-1 \
    --name Queue1 \
    --description "Keep this queue empty unless job is urgent."
```

若要取得您的帳戶特定端點，請使用 `describe-endpoints`，或傳送不含端點的 命令。服務會傳回錯誤和您的端點。

輸出：

```
{
    "Queue": {
        "Status": "ACTIVE",
        "Name": "Queue1",
        "LastUpdated": 1518034928,
        "Arn": "arn:aws:mediaconvert:region-name-1:012345678998:queues/Queue1",
        "Type": "CUSTOM",
        "CreatedAt": 1518034928,
        "Description": "Keep this queue empty unless job is urgent."
    }
}
```

如需詳細資訊，請參閱 AWS Elemental MediaConvert 使用者指南中的使用 Elemental MediaConvert 佇列。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateQueue。

delete-job-template

以下程式碼範例顯示如何使用 `delete-job-template`。

AWS CLI

刪除任務範本

下列`delete-job-template`範例會刪除指定的自訂任務範本。

```
aws mediaconvert delete-job-template \
    --name "DASH Streaming" \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

此命令不會產生輸出。執行 `aws mediaconvert list-job-templates` 以確認您的範本已刪除。

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的使用 Elemental MediaConvert 任務範本。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteJobTemplate。

delete-preset

以下程式碼範例顯示如何使用 delete-preset。

AWS CLI

刪除自訂隨需佇列

下列delete-preset範例會刪除指定的自訂預設集。

```
aws mediaconvert delete-preset \
    --name SimpleMP4 \
    --endpoint-url  https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

此命令不會產生輸出。執行 aws mediaconvert list-presets 以確認您的預設集已刪除。

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的使用 Elemental MediaConvert 輸出預設集。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePreset。

delete-queue

以下程式碼範例顯示如何使用 delete-queue。

AWS CLI

刪除自訂隨需佇列

下列delete-queue範例會刪除指定的自訂隨需佇列。

您無法刪除預設佇列。在定價計劃期間內或預留佇列中仍有未處理的任務時，您無法刪除預留佇列。

```
aws mediaconvert delete-queue \
    --name Customer1 \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

此命令不會產生輸出。執行 aws mediaconvert list-queues 以確認您的佇列已刪除。

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的使用 Elemental MediaConvert 佇列。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteQueue。

describe-endpoints

以下程式碼範例顯示如何使用 describe-endpoints。

AWS CLI

取得您的帳戶特定端點

下列describe-endpoints範例會擷取您需要將任何其他請求傳送至服務的端點。

```
aws mediaconvert describe-endpoints
```

輸出：

```
{
    "Endpoints": [
        {
            "Url": "https://abcd1234.mediaconvert.region-name-1.amazonaws.com"
        }
    ]
}
```

如需詳細資訊，請參閱 Elemental MediaConvert API 參考中的使用 API 的 MediaConvert 入門。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEndpoints。

get-job-template

以下程式碼範例顯示如何使用 get-job-template。

AWS CLI

取得任務範本的詳細資訊

下列get-job-template範例顯示指定自訂任務範本的 JSON 定義。

```
aws mediaconvert get-job-template \
    --name "DASH Streaming" \
    --endpoint-url https://abcd1234.mediaconvert.us-east-1.amazonaws.com
```

輸出：

```
{
    "JobTemplate": {
        "StatusUpdateInterval": "SECONDS_60",
        "LastUpdated": 1568652998,
        "Description": "Create a DASH streaming ABR stack",
        "CreatedAt": 1568652998,
        "Priority": 0,
        "Name": "DASH Streaming",
        "Settings": {
            ...<truncatedforbrevity>...
        },
        "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:jobTemplates/DASH
 Streaming",
        "Type": "CUSTOM"
    }
}
```

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的使用 Elemental MediaConvert 任務範本。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetJobTemplate。

get-job

以下程式碼範例顯示如何使用 get-job。

AWS CLI

取得特定任務的詳細資訊

下列範例會請求 ID 為 之任務的資訊1234567890987-1ab2c3，在此範例中以錯誤結尾。

```
aws mediaconvert get-job \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
    --region region-name-1 \
    --id 1234567890987-1ab2c3
```

若要取得您的帳戶特定端點，請使用 `describe-endpoints`，或傳送不含端點的 命令。服務會傳回錯誤和您的端點。

如果您的請求成功，服務會傳回包含任務資訊的 JSON 檔案，包括任務設定、任何傳回的錯誤和其他任務資料，如下所示：

```
{
    "Job": {
        "Status": "ERROR",
        "Queue": "arn:aws:mediaconvert:region-name-1:012345678998:queues/Queue1",
        "Settings": {
            ...<truncated for brevity>...
        },
        "ErrorMessage": "Unable to open input file [s3://my-input-bucket/file-
name.mp4]: [Failed probe/open: [Failed to read data: AssumeRole failed]]",
        "ErrorCode": 1434,
        "Role": "arn:aws:iam::012345678998:role/MediaConvertServiceRole",
        "Arn": "arn:aws:mediaconvert:us-
west-1:012345678998:jobs/1234567890987-1ab2c3",
        "UserMetadata": {},
        "Timing": {
            "FinishTime": 1517442131,
            "SubmitTime": 1517442103,
            "StartTime": 1517442104
        },
        "Id": "1234567890987-1ab2c3",
        "CreatedAt": 1517442103
    }
}
```

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的使用 Elemental MediaConvert 任務。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetJob。

get-preset

以下程式碼範例顯示如何使用 `get-preset`。

AWS CLI

取得特定預設集的詳細資訊

下列get-preset範例會請求指定自訂預設集的 JSON 定義。

```
aws mediaconvert get-preset \
    --name SimpleMP4 \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

輸出：

```
{
    "Preset": {
        "Description": "Creates basic MP4 file. No filtering or preproccessing.",
        "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:presets/SimpleMP4",
        "LastUpdated": 1568843141,
        "Name": "SimpleMP4",
        "Settings": {
            "ContainerSettings": {
                "Mp4Settings": {
                    "FreeSpaceBox": "EXCLUDE",
                    "CslgAtom": "INCLUDE",
                    "MoovPlacement": "PROGRESSIVE_DOWNLOAD"
                },
                "Container": "MP4"
            },
            "AudioDescriptions": [
                {
                    "LanguageCodeControl": "FOLLOW_INPUT",
                    "AudioTypeControl": "FOLLOW_INPUT",
                    "CodecSettings": {
                        "AacSettings": {
                            "RawFormat": "NONE",
                            "CodecProfile": "LC",
                            "AudioDescriptionBroadcasterMix": "NORMAL",
                            "SampleRate": 48000,
                            "Bitrate": 96000,
                            "RateControlMode": "CBR",
                            "Specification": "MPEG4",
                            "CodingMode": "CODING_MODE_2_0"
                        },
                        "Codec": "AAC"
                    }
                }
            ],
            "VideoDescription": {
```

```
                "RespondToAfd": "NONE",
                "TimecodeInsertion": "DISABLED",
                "Sharpness": 50,
                "ColorMetadata": "INSERT",
                "CodecSettings": {
                    "H264Settings": {
                        "FramerateControl": "INITIALIZE_FROM_SOURCE",
                        "SpatialAdaptiveQuantization": "ENABLED",
                        "Softness": 0,
                        "Telecine": "NONE",
                        "CodecLevel": "AUTO",
                        "QualityTuningLevel": "SINGLE_PASS",
                        "UnregisteredSeiTimecode": "DISABLED",
                        "Slices": 1,
                        "Syntax": "DEFAULT",
                        "GopClosedCadence": 1,
                        "AdaptiveQuantization": "HIGH",
                        "EntropyEncoding": "CABAC",
                        "InterlaceMode": "PROGRESSIVE",
                        "ParControl": "INITIALIZE_FROM_SOURCE",
                        "NumberBFramesBetweenReferenceFrames": 2,
                        "GopSizeUnits": "FRAMES",
                        "RepeatPps": "DISABLED",
                        "CodecProfile": "MAIN",
                        "FieldEncoding": "PAFF",
                        "GopSize": 90.0,
                        "SlowPal": "DISABLED",
                        "SceneChangeDetect": "ENABLED",
                        "GopBReference": "DISABLED",
                        "RateControlMode": "CBR",
                        "FramerateConversionAlgorithm": "DUPLICATE_DROP",
                        "FlickerAdaptiveQuantization": "DISABLED",
                        "DynamicSubGop": "STATIC",
                        "MinIInterval": 0,
                        "TemporalAdaptiveQuantization": "ENABLED",
                        "Bitrate": 400000,
                        "NumberReferenceFrames": 3
                    },
                    "Codec": "H_264"
                },
                "AfdSignaling": "NONE",
                "AntiAlias": "ENABLED",
                "ScalingBehavior": "DEFAULT",
                "DropFrameTimecode": "ENABLED"
```

```
            }
        },
        "Type": "CUSTOM",
        "CreatedAt": 1568841521
    }
}
```

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的使用 Elemental MediaConvert 輸出預設集。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPreset。

`get-queue`

以下程式碼範例顯示如何使用 `get-queue`。

AWS CLI

取得佇列的詳細資訊

下列`get-queue`範例會擷取指定自訂佇列的詳細資訊。

```
aws mediaconvert get-queue \
    --name Customer1 \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

輸出：

```
{
    "Queue": {
        "LastUpdated": 1526428502,
        "Type": "CUSTOM",
        "SubmittedJobsCount": 0,
        "Status": "ACTIVE",
        "PricingPlan": "ON_DEMAND",
        "CreatedAt": 1526428502,
        "ProgressingJobsCount": 0,
        "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:queues/Customer1",
        "Name": "Customer1"
    }
}
```

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的使用 Elemental MediaConvert 佇列。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetQueue。

`list-job-templates`

以下程式碼範例顯示如何使用 `list-job-templates`。

AWS CLI

範例 1：列出您的自訂任務範本

下列`list-job-templates`範例列出目前區域中的所有自訂任務範本。若要列出系統任務範本，請參閱下一個範例。

```
aws mediaconvert list-job-templates \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

輸出：

```
{
    "JobTemplates": [
        {
            "Description": "Create a DASH streaming ABR stack",
            "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:jobTemplates/DASH
 Streaming",
            "Name": "DASH Streaming",
            "LastUpdated": 1568653007,
            "Priority": 0,
            "Settings": {
                ...<truncatedforbrevity>...
            },
            "Type": "CUSTOM",
            "StatusUpdateInterval": "SECONDS_60",
            "CreatedAt": 1568653007
        },
        {
            "Description": "Create a high-res file",
            "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:jobTemplates/File",
            "Name": "File",
            "LastUpdated": 1568653007,
            "Priority": 0,
```

```
            "Settings": {
                ...<truncatedforbrevity>...
            },
            "Type": "CUSTOM",
            "StatusUpdateInterval": "SECONDS_60",
            "CreatedAt": 1568653023
        }
    ]
}
```

範例 2：列出 MediaConvert 系統任務範本

下列list-job-templates範例列出所有系統任務範本。

```
aws mediaconvert list-job-templates \
    --endpoint-url https://abcd1234.mediaconvert.us-east-1.amazonaws.com \
    --list-by SYSTEM
```

輸出：

```
{
    "JobTemplates": [
        {
            "CreatedAt": 1568321779,
            "Arn": "arn:aws:mediaconvert:us-east-1:123456789012:jobTemplates/System-
Generic_Mp4_Hev1_Avc_Aac_Sdr_Qvbr",
            "Name": "System-Generic_Mp4_Hev1_Avc_Aac_Sdr_Qvbr",
            "Description": "GENERIC, MP4, AVC + HEV1(HEVC,SDR), AAC, SDR, QVBR",
            "Category": "GENERIC",
            "Settings": {
                "AdAvailOffset": 0,
                "OutputGroups": [
                    {
                        "Outputs": [
                            {
                                "Extension": "mp4",
                                "Preset": "System-
Generic_Hd_Mp4_Avc_Aac_16x9_Sdr_1280x720p_30Hz_5Mbps_Qvbr_Vq9",
                                "NameModifier":
 "_Generic_Hd_Mp4_Avc_Aac_16x9_Sdr_1280x720p_30Hz_5000Kbps_Qvbr_Vq9"
                            },
                            {
                                "Extension": "mp4",
```

```
                                "Preset": "System-
Generic_Hd_Mp4_Avc_Aac_16x9_Sdr_1920x1080p_30Hz_10Mbps_Qvbr_Vq9",
                                "NameModifier":
 "_Generic_Hd_Mp4_Avc_Aac_16x9_Sdr_1920x1080p_30Hz_10000Kbps_Qvbr_Vq9"
                            },
                            {
                                "Extension": "mp4",
                                "Preset": "System-
Generic_Sd_Mp4_Avc_Aac_16x9_Sdr_640x360p_30Hz_0.8Mbps_Qvbr_Vq7",
                                "NameModifier":
 "_Generic_Sd_Mp4_Avc_Aac_16x9_Sdr_640x360p_30Hz_800Kbps_Qvbr_Vq7"
                            },
                            {
                                "Extension": "mp4",
                                "Preset": "System-
Generic_Hd_Mp4_Hev1_Aac_16x9_Sdr_1280x720p_30Hz_4Mbps_Qvbr_Vq9",
                                "NameModifier":
 "_Generic_Hd_Mp4_Hev1_Aac_16x9_Sdr_1280x720p_30Hz_4000Kbps_Qvbr_Vq9"
                            },
                            {
                                "Extension": "mp4",
                                "Preset": "System-
Generic_Hd_Mp4_Hev1_Aac_16x9_Sdr_1920x1080p_30Hz_8Mbps_Qvbr_Vq9",
                                "NameModifier":
 "_Generic_Hd_Mp4_Hev1_Aac_16x9_Sdr_1920x1080p_30Hz_8000Kbps_Qvbr_Vq9"
                            },
                            {
                                "Extension": "mp4",
                                "Preset": "System-
Generic_Uhd_Mp4_Hev1_Aac_16x9_Sdr_3840x2160p_30Hz_12Mbps_Qvbr_Vq9",
                                "NameModifier":
 "_Generic_Uhd_Mp4_Hev1_Aac_16x9_Sdr_3840x2160p_30Hz_12000Kbps_Qvbr_Vq9"
                            }
                        ],
                        "OutputGroupSettings": {
                            "FileGroupSettings": {

                            },
                            "Type": "FILE_GROUP_SETTINGS"
                        },
                        "Name": "File Group"
                    }
                ]
            },
```

```
            "Type": "SYSTEM",
            "LastUpdated": 1568321779
        },
        ...<truncatedforbrevity>...
    ]
}
```

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的使用 Elemental MediaConvert 任務範本。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListJobTemplates。

`list-jobs`

以下程式碼範例顯示如何使用 `list-jobs`。

AWS CLI

取得區域中所有任務的詳細資訊

下列範例會請求指定區域中所有任務的資訊。

```
aws mediaconvert list-jobs \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
    --region region-name-1
```

若要取得您的帳戶特定端點，請使用 `describe-endpoints`，或傳送不含端點的 命令。服務會傳回錯誤和您的端點。

如需詳細資訊，請參閱 AWS Elemental MediaConvert 使用者指南中的使用 Elemental MediaConvert 任務。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListJobs。

`list-presets`

以下程式碼範例顯示如何使用 `list-presets`。

AWS CLI

範例 1：列出您的自訂輸出預設集

下列`list-presets`範例列出您的自訂輸出預設集。若要列出系統預設集，請參閱下一個範例。

```
aws mediaconvert list-presets \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

輸出：

```
{
    "Presets": [
        {
            "Name": "SimpleMP4",
            "CreatedAt": 1568841521,
            "Settings": {
                ......
            },
            "Arn": "arn:aws:mediaconvert:us-east-1:003235472598:presets/SimpleMP4",
            "Type": "CUSTOM",
            "LastUpdated": 1568843141,
            "Description": "Creates basic MP4 file. No filtering or preproccessing."
        },
        {
            "Name": "SimpleTS",
            "CreatedAt": 1568843113,
            "Settings": {
                ... truncated for brevity ...
            },
            "Arn": "arn:aws:mediaconvert:us-east-1:003235472598:presets/SimpleTS",
            "Type": "CUSTOM",
            "LastUpdated": 1568843113,
            "Description": "Create a basic transport stream."
        }
    ]
}
```

範例 2：列出系統輸出預設集

下列`list-presets`範例列出可用的 MediaConvert 系統預設集。若要列出您的自訂預設集，請參閱先前的範例。

```
aws mediaconvert list-presets \
    --list-by SYSTEM \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

輸出：

```
{
    "Presets": [
        {
            "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:presets/System-
Avc_16x9_1080p_29_97fps_8500kbps",
            "Name": "System-Avc_16x9_1080p_29_97fps_8500kbps",
            "CreatedAt": 1568321789,
            "Description": "Wifi, 1920x1080, 16:9, 29.97fps, 8500kbps",
            "LastUpdated": 1568321789,
            "Type": "SYSTEM",
            "Category": "HLS",
            "Settings": {
            ...<output settings removed for brevity>...
            }
        },

        ...<list of presets shortened for brevity>...

        {
            "Arn": "arn:aws:mediaconvert:us-east-1:123456789012:presets/System-
Xdcam_HD_1080i_29_97fps_35mpbs",
            "Name": "System-Xdcam_HD_1080i_29_97fps_35mpbs",
            "CreatedAt": 1568321790,
            "Description": "XDCAM MPEG HD, 1920x1080i, 29.97fps, 35mbps",
            "LastUpdated": 1568321790,
            "Type": "SYSTEM",
            "Category": "MXF",
            "Settings": {
            ...<output settings removed for brevity>...
            }
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的使用 Elemental MediaConvert 輸出預設集。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPresets。

`list-queues`

以下程式碼範例顯示如何使用 `list-queues`。

AWS CLI

列出您的佇列

下列`list-queues`範例列出所有 MediaConvert 佇列。

```
aws mediaconvert list-queues \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

輸出：

```
{
    "Queues": [
        {
            "PricingPlan": "ON_DEMAND",
            "Type": "SYSTEM",
            "Status": "ACTIVE",
            "CreatedAt": 1503451595,
            "Name": "Default",
            "SubmittedJobsCount": 0,
            "ProgressingJobsCount": 0,
            "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:queues/Default",
            "LastUpdated": 1534549158
        },
        {
            "PricingPlan": "ON_DEMAND",
            "Type": "CUSTOM",
            "Status": "ACTIVE",
            "CreatedAt": 1537460025,
            "Name": "Customer1",
            "SubmittedJobsCount": 0,
            "Description": "Jobs we run for our cusotmer.",
            "ProgressingJobsCount": 0,
            "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:queues/Customer1",
            "LastUpdated": 1537460025
        },
        {
            "ProgressingJobsCount": 0,
            "Status": "ACTIVE",
            "Name": "transcode-library",
            "SubmittedJobsCount": 0,
            "LastUpdated": 1564066204,
            "ReservationPlan": {
```

```
                "Status": "ACTIVE",
                "ReservedSlots": 1,
                "PurchasedAt": 1564066203,
                "Commitment": "ONE_YEAR",
                "ExpiresAt": 1595688603,
                "RenewalType": "EXPIRE"
            },
            "PricingPlan": "RESERVED",
            "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:queues/transcode-
library",
            "Type": "CUSTOM",
            "CreatedAt": 1564066204
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的使用 Elemental MediaConvert 佇列。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListQueues。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出 MediaConvert 佇列、任務範本或輸出預設集上的標籤

下列`list-tags-for-resource`範例列出指定輸出預設集上的標籤。

```
aws mediaconvert list-tags-for-resource \
    --arn arn:aws:mediaconvert:us-west-2:123456789012:presets/SimpleMP4 \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

輸出：

```
{
    "ResourceTags": {
        "Tags": {
            "customer": "zippyVideo"
        },
        "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:presets/SimpleMP4"
```

```
    }
}
```

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的標記 Elemental MediaConvert 佇列、任務範本和輸出預設集。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

update-job-template

以下程式碼範例顯示如何使用 update-job-template。

AWS CLI

變更任務範本

下列update-job-template範例會將指定自訂任務範本的 JSON 定義取代為所提供檔案中的 JSON 定義。

aws mediaconvert update-job-template --name File1 --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com --cli-input-json file：//~/job-template-update.json

job-template-update.json 的內容：

```
{
    "Description": "A simple job template that generates a single file output.",
    "Queue": "arn:aws:mediaconvert:us-east-1:012345678998:queues/Default",
    "Name": "SimpleFile",
    "Settings": {
      "OutputGroups": [
        {
          "Name": "File Group",
          "Outputs": [
            {
              "ContainerSettings": {
                "Container": "MP4",
                "Mp4Settings": {
                  "CslgAtom": "INCLUDE",
                  "FreeSpaceBox": "EXCLUDE",
                  "MoovPlacement": "PROGRESSIVE_DOWNLOAD"
                }
              },
```

```
                "VideoDescription": {
                  "ScalingBehavior": "DEFAULT",
                  "TimecodeInsertion": "DISABLED",
                  "AntiAlias": "ENABLED",
                  "Sharpness": 50,
                  "CodecSettings": {
                    "Codec": "H_264",
                    "H264Settings": {
                      "InterlaceMode": "PROGRESSIVE",
                      "NumberReferenceFrames": 3,
                      "Syntax": "DEFAULT",
                      "Softness": 0,
                      "GopClosedCadence": 1,
                      "GopSize": 90,
                      "Slices": 1,
                      "GopBReference": "DISABLED",
                      "SlowPal": "DISABLED",
                      "SpatialAdaptiveQuantization": "ENABLED",
                      "TemporalAdaptiveQuantization": "ENABLED",
                      "FlickerAdaptiveQuantization": "DISABLED",
                      "EntropyEncoding": "CABAC",
                      "Bitrate": 400000,
                      "FramerateControl": "INITIALIZE_FROM_SOURCE",
                      "RateControlMode": "CBR",
                      "CodecProfile": "MAIN",
                      "Telecine": "NONE",
                      "MinIInterval": 0,
                      "AdaptiveQuantization": "HIGH",
                      "CodecLevel": "AUTO",
                      "FieldEncoding": "PAFF",
                      "SceneChangeDetect": "ENABLED",
                      "QualityTuningLevel": "SINGLE_PASS",
                      "FramerateConversionAlgorithm": "DUPLICATE_DROP",
                      "UnregisteredSeiTimecode": "DISABLED",
                      "GopSizeUnits": "FRAMES",
                      "ParControl": "INITIALIZE_FROM_SOURCE",
                      "NumberBFramesBetweenReferenceFrames": 2,
                      "RepeatPps": "DISABLED",
                      "DynamicSubGop": "STATIC"
                    }
                  },
                  "AfdSignaling": "NONE",
                  "DropFrameTimecode": "ENABLED",
                  "RespondToAfd": "NONE",
```

```
                "ColorMetadata": "INSERT"
              },
              "AudioDescriptions": [
                {
                  "AudioTypeControl": "FOLLOW_INPUT",
                  "CodecSettings": {
                    "Codec": "AAC",
                    "AacSettings": {
                      "AudioDescriptionBroadcasterMix": "NORMAL",
                      "Bitrate": 96000,
                      "RateControlMode": "CBR",
                      "CodecProfile": "LC",
                      "CodingMode": "CODING_MODE_2_0",
                      "RawFormat": "NONE",
                      "SampleRate": 48000,
                      "Specification": "MPEG4"
                    }
                  },
                  "LanguageCodeControl": "FOLLOW_INPUT"
                }
              ]
            }
          ],
          "OutputGroupSettings": {
            "Type": "FILE_GROUP_SETTINGS",
            "FileGroupSettings": {}
          }
        }
      ],
      "AdAvailOffset": 0
    },
    "StatusUpdateInterval": "SECONDS_60",
    "Priority": 0
}
```

即使請求導致錯誤，系統仍會傳回您隨請求一起傳送的 JSON 承載。因此，傳回的 JSON 不一定是任務範本的新定義。

由於 JSON 承載可能很長，您可能需要向上捲動以查看任何錯誤訊息。

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的使用 Elemental MediaConvert 任務範本。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateJobTemplate。

update-preset

以下程式碼範例顯示如何使用 update-preset。

AWS CLI

變更預設集

下列update-preset範例會取代指定預設集的描述。

```
aws mediaconvert update-preset \
--name Customer1 \
--description "New description text."
--endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

此命令不會產生輸出。輸出：

```
{
    "Preset": {
        "Arn": "arn:aws:mediaconvert:us-east-1:003235472598:presets/SimpleMP4",
        "Settings": {
        ...<output settings removed for brevity>...
        },
        "Type": "CUSTOM",
        "LastUpdated": 1568938411,
        "Description": "New description text.",
        "Name": "SimpleMP4",
        "CreatedAt": 1568938240
    }
}
```

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的使用 Elemental MediaConvert 輸出預設集。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePreset。

update-queue

以下程式碼範例顯示如何使用 update-queue。

AWS CLI

變更佇列

下列update-queue範例會將其狀態變更為 ，以暫停指定的佇列PAUSED。

```
aws mediaconvert update-queue \
--name Customer1 \
--status PAUSED
--endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

輸出：

```
{
    "Queue": {
        "LastUpdated": 1568839845,
        "Status": "PAUSED",
        "ProgressingJobsCount": 0,
        "CreatedAt": 1526428516,
        "Arn": "arn:aws:mediaconvert:us-west-1:123456789012:queues/Customer1",
        "Name": "Customer1",
        "SubmittedJobsCount": 0,
        "PricingPlan": "ON_DEMAND",
        "Type": "CUSTOM"
    }
}
```

如需詳細資訊，請參閱《AWS Elemental MediaConvert 使用者指南》中的使用 Elemental MediaConvert 佇列。 AWS MediaConvert

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateQueue。

使用 的 MediaLive 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 MediaLive 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-channel`

以下程式碼範例顯示如何使用 `create-channel`。

AWS CLI

建立頻道

下列`create-channel`範例會透過傳入包含您要指定之參數的 JSON 檔案來建立頻道。

此範例中的頻道會擷取 HLS PULL 輸入，該輸入會連接到包含視訊、音訊和內嵌字幕的來源。頻道會建立一個 HLS 輸出群組，並將 Akamai 伺服器做為目的地。輸出群組包含兩個輸出：一個用於 H.265 視訊和 AAC 音訊，另一個用於 Web-VTT 字幕，僅以英文顯示。

此範例頻道的 JSON 包含使用 HLS PULL 輸入，並以 Akamai 做為目的地產生 HLS 輸出群組的頻道所需的最低參數。JSON 包含下列主要區段：

`InputAttachments`，指定一個音訊來源，以及一個字幕來源。它不會指定影片選擇器，這表示 MediaLive 會擷取它在來源中找到的第一個影片。 `Destinations`包含此頻道中單一輸出群組的兩個 IP 地址 (URLs)。這些地址需要密碼。 `EncoderSettings`包含子區段。`AudioDescriptions`指定頻道包含一個音訊輸出資產，其使用來自 InputAttachments 的來源，並以 AAC 格式產生音訊。 `CaptionDescriptions`指定頻道包含一個字幕輸出資產，其使用來自 InputAttachments 的來源，並以 Web-VTT 格式產生字幕。 `VideoDescriptions`指定頻道包含一個視訊輸出資產，具有指定的解析度。 `OutputGroups`指定輸出群組。在此範例中，有一個名為 的群組`Akamai`。使用 HLS PUT 進行連線。輸出群組包含兩個輸出。一個輸出用於視訊資產 （名為 `Video_high`) 和音訊資產 （名為 `Audio_EN`)。一個輸出適用於字幕資產 （名為 `WebVTT_EN`)。

在此範例中，某些參數不包含任何值或包含巢狀空白參數。例如，`Video_and_audio`輸出的 OutputSettings 包含數個以空參數 M3u8Settings 結尾的巢狀參數。必須包含此參數，但您可以省略一個、數個或所有子項，這表示子項將採用其預設值或為 null。

套用至此範例頻道但未在此檔案中指定的所有參數都會採用預設值、設定為 null，或採用 MediaLive 產生的唯一值。

```
aws medialive create-channel \
    --cli-input-json file://channel-in-hls-out-hls-akamai.json
```

channel-in-hls-out-hls-akamai.json 的內容：

```
{
    "Name": "News_West",
    "RoleArn": "arn:aws:iam::111122223333:role/MediaLiveAccessRole",
    "InputAttachments": [
        {
            "InputAttachmentName": "local_news",
            "InputId": "1234567",
            "InputSettings": {
                "AudioSelectors": [
                    {
                        "Name": "English-Audio",
                        "SelectorSettings": {
                            "AudioLanguageSelection": {
                                "LanguageCode": "EN"
                            }
                        }
                    }
                ],
                "CaptionSelectors": [
                    {
                        "LanguageCode": "ENE",
                        "Name": "English_embedded"
                    }
                ]
            }
        }
    ],
    "Destinations": [
        {
            "Id": "akamai-server-west",
            "Settings": [
                {
                    "PasswordParam": "/medialive/examplecorp1",
                    "Url": "http://203.0.113.55/news/news_west",
                    "Username": "examplecorp"
                },
                {
                    "PasswordParam": "/medialive/examplecorp2",
                    "Url": "http://203.0.113.82/news/news_west",
                    "Username": "examplecorp"
                }
            ]
```

```
        }
    ],
    "EncoderSettings": {
        "AudioDescriptions": [
            {
                "AudioSelectorName": "English-Audio",
                "CodecSettings": {
                    "AacSettings": {}
                },
                "Name": "Audio_EN"
            }
        ],
        "CaptionDescriptions": [
            {
                "CaptionSelectorName": "English_embedded",
                "DestinationSettings": {
                    "WebvttDestinationSettings": {}
                },
                "Name": "WebVTT_EN"
            }
        ],
        "VideoDescriptions": [
            {
                "Height": 720,
                "Name": "Video_high",
                "Width": 1280
            }
        ],
        "OutputGroups": [
            {
                "Name": "Akamai",
                "OutputGroupSettings": {
                    "HlsGroupSettings": {
                        "Destination": {
                            "DestinationRefId": "akamai-server-west"
                        },
                        "HlsCdnSettings": {
                            "HlsBasicPutSettings": {}
                        }
                    }
                },
                "Outputs": [
                    {
                        "AudioDescriptionNames": [
```

```
                            "Audio_EN"
                        ],
                        "OutputName": "Video_and_audio",
                        "OutputSettings": {
                            "HlsOutputSettings": {
                                "HlsSettings": {
                                    "StandardHlsSettings": {
                                        "M3u8Settings": {}
                                    }
                                },
                                "NameModifier": "_1"
                            }
                        },
                        "VideoDescriptionName": "Video_high"
                    },
                    {
                        "CaptionDescriptionNames": [
                            "WebVTT_EN"
                        ],
                        "OutputName": "Captions-WebVTT",
                        "OutputSettings": {
                            "HlsOutputSettings": {
                                "HlsSettings": {
                                    "StandardHlsSettings": {
                                        "M3u8Settings": {}
                                    }
                                },
                                "NameModifier": "_2"
                            }
                        }
                    }
                ]
            }
        ],
        "TimecodeConfig": {
            "Source": "EMBEDDED"
        }
    }
}
```

輸出：

輸出會重複傳回 JSON 檔案的內容，加上下列值。所有參數都會依字母順序排序。

ARN 頻道的 。ARN 的最後一部分是唯一的頻道 ID。EgressEndpoints在此範例頻道中為空白，因為它僅用於 PUSH 輸入。套用時，它會在 MediaLive 上顯示推送內容的地址。OutputGroups，Outputs。這些會顯示輸出群組和輸出的所有參數，包括您未包含但與此頻道相關的參數。參數可能是空的（可能表示此頻道組態中的參數或功能已停用），或可能顯示將套用的預設值。LogLevel 設定為預設值 (DISABLED)。Tags 設定為預設值 (null)PipelinesRunningCount，並State顯示頻道的目前狀態。

如需詳細資訊，請參閱 AWS Elemental MediaLive 使用者指南中的從 Scratch 建立頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateChannel。

create-input

以下程式碼範例顯示如何使用 create-input。

AWS CLI

建立輸入

下列create-input範例會透過傳入 JSON 檔案來建立HLS PULL輸入，其中包含適用於此類型輸入的參數。此範例輸入的 JSON 會指定輸入的兩個來源（地址），以支援擷取中的備援。這些地址需要密碼。

```
aws medialive create-input \
    --cli-input-json file://input-hls-pull-news.json
```

input-hls-pull-news.json 的內容：

```
{
    "Name": "local_news",
    "RequestId": "cli000059",
    "Sources": [
            {
                    "Url": "https://203.0.113.13/newschannel/anytownusa.m3u8",
                    "Username": "examplecorp",
                    "PasswordParam": "/medialive/examplecorp1"
            },
       {
                    "Url": "https://198.51.100.54/fillervideos/oceanwaves.mp4",
                    "Username": "examplecorp",
                    "PasswordParam": "examplecorp2"
```

```
            }
    ],
    "Type": "URL_PULL"
}
```

輸出：

輸出會重複傳回 JSON 檔案的內容，加上下列值。所有參數都會依字母順序排序。

Arn 輸入。ARN 的最後一個部分是唯一的輸入 ID。Attached Channels對於新建立的輸入一律為空白。在此範例中為空白Destinations，因為它僅用於 PUSH 輸入。Id對於輸入，與 ARN 中的 ID 相同。在此範例中為空白MediaConnectFlows，因為它僅用於 MediaConnect 類型的輸入。在此範例中為空白SecurityGroups，因為它僅用於此輸入的 PUSH 輸入State。 Tags為空白（此參數的預設值）。

如需詳細資訊，請參閱 AWS Elemental MediaLive 使用者指南中的建立輸入。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateInput。

使用 的 MediaPackage 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 MediaPackage 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-channel`

以下程式碼範例顯示如何使用 create-channel。

AWS CLI

建立頻道

下列create-channel命令會在目前帳戶中建立名為 sportschannel的頻道。

```
aws mediapackage create-channel --id sportschannel
```

輸出：

```
{
   "Arn": "arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0",
   "HlsIngest": {
        "IngestEndpoints": [
            {
                "Id": "6d345804ec3f46c9b454a91d4a80d0e0",
                "Password": "generatedwebdavpassword1",
                "Url": "https://f31c86aed53b815a.mediapackage.us-
west-2.amazonaws.com/in/
v2/6d345804ec3f46c9b454a91d4a80d0e0/6d345804ec3f46c9b454a91d4a80d0e0/channel",
                "Username": "generatedwebdavusername1"
            },
            {
                "Id": "2daa32878af24803b24183727211b8ff",
                "Password": "generatedwebdavpassword2",
                "Url": "https://6ebbe7e04c4b0afa.mediapackage.us-
west-2.amazonaws.com/in/
v2/6d345804ec3f46c9b454a91d4a80d0e0/2daa32878af24803b24183727211b8ff/channel",
                "Username": "generatedwebdavusername2"
            }
        ]
    },
    "Id": "sportschannel",
    "Tags": {
        "region": "west"
    }
}
```

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的建立頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateChannel。

create-origin-endpoint

以下程式碼範例顯示如何使用 create-origin-endpoint。

AWS CLI

建立原始伺服器端點

下列create-origin-endpoint命令cmafsports會使用 JSON 檔案中提供的套件設定和指定的端點設定來建立名為 的原始伺服器端點。

```
aws mediapackage create-origin-endpoint \
    --channel-id sportschannel \
    --id cmafsports \
    --cmaf-package file://file/path/cmafpkg.json --description "cmaf output of
 sports" \
    --id cmaf_sports \
    --manifest-name sports_channel \
    --startover-window-seconds 300 \
    --tags region=west,media=sports \
    --time-delay-seconds 10
```

輸出：

```
{
    "Arn": "arn:aws:mediapackage:us-
west-2:111222333:origin_endpoints/1dc6718be36f4f34bb9cd86bc50925e6",
    "ChannelId": "sportschannel",
    "CmafPackage": {
        "HlsManifests": [
            {
                "AdMarkers": "PASSTHROUGH",
                "Id": "cmaf_sports_endpoint",
                "IncludeIframeOnlyStream": true,
                "ManifestName": "index",
                "PlaylistType": "EVENT",
                "PlaylistWindowSeconds": 300,
                "ProgramDateTimeIntervalSeconds": 300,
                "Url": "https://c4af3793bf76b33c.mediapackage.us-
west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/cmaf_sports_endpoint/
index.m3u8"
            }
        ],
        "SegmentDurationSeconds": 2,
        "SegmentPrefix": "sportschannel"
    },
    "Description": "cmaf output of sports",
```

```
    "Id": "cmaf_sports",
    "ManifestName": "sports_channel",
    "StartoverWindowSeconds": 300,
    "Tags": {
        "region": "west",
        "media": "sports"
    },
    "TimeDelaySeconds": 10,
    "Url": "",
    "Whitelist": []
}
```

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的建立端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateOriginEndpoint。

delete-channel

以下程式碼範例顯示如何使用 delete-channel。

AWS CLI

刪除頻道

下列delete-channel命令會刪除名為 的頻道test。

```
aws mediapackage delete-channel \
    --id test
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的刪除頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteChannel。

delete-origin-endpoint

以下程式碼範例顯示如何使用 delete-origin-endpoint。

AWS CLI

刪除原始伺服器端點

下列delete-origin-endpoint命令會刪除名為 的原始伺服器端點tester2。

```
aws mediapackage delete-origin-endpoint \
    --id tester2
```

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的刪除端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteOriginEndpoint。

describe-channel

以下程式碼範例顯示如何使用 describe-channel。

AWS CLI

描述頻道

下列describe-channel命令會顯示名為 之頻道的所有詳細資訊test。

```
aws mediapackage describe-channel \
    --id test
```

輸出：

```
{
    "Arn": "arn:aws:mediapackage:us-
west-2:111222333:channels/584797f1740548c389a273585dd22a63",
    "HlsIngest": {
        "IngestEndpoints": [
            {
                "Id": "584797f1740548c389a273585dd22a63",
                "Password": "webdavgeneratedpassword1",
                "Url": "https://9be9c4405c474882.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/584797f1740548c389a273585dd22a63/channel",
                "Username": "webdavgeneratedusername1"
            },
            {
                "Id": "7d187c8616fd455f88aaa5a9fcf74442",
                "Password": "webdavgeneratedpassword2",
                "Url": "https://7bf454c57220328d.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/7d187c8616fd455f88aaa5a9fcf74442/channel",
```

```
                "Username": "webdavgeneratedusername2"
            }
        ]
    },
    "Id": "test",
    "Tags": {}
}
```

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的檢視頻道詳細資訊<https://docs.aws.amazon.com/mediapackage/latest/ug/channels-view.html>

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeChannel。

describe-origin-endpoint

以下程式碼範例顯示如何使用 describe-origin-endpoint。

AWS CLI

描述原始伺服器端點

下列describe-origin-endpoint命令會顯示名為 的原始伺服器端點的所有詳細資訊cmaf_sports。

```
aws mediapackage describe-origin-endpoint \
    --id cmaf_sports
```

輸出：

```
{
    "Arn": "arn:aws:mediapackage:us-
west-2:111222333:origin_endpoints/1dc6718be36f4f34bb9cd86bc50925e6",
    "ChannelId": "sportschannel",
    "CmafPackage": {
        "HlsManifests": [
            {
                "AdMarkers": "NONE",
                "Id": "cmaf_sports_endpoint",
                "IncludeIframeOnlyStream": false,
                "PlaylistType": "EVENT",
                "PlaylistWindowSeconds": 60,
                "ProgramDateTimeIntervalSeconds": 0,
```

```
                "Url": "https://c4af3793bf76b33c.mediapackage.us-
west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/cmaf_sports_endpoint/
index.m3u8"
            }
        ],
        "SegmentDurationSeconds": 2,
        "SegmentPrefix": "sportschannel"
    },
    "Id": "cmaf_sports",
    "ManifestName": "index",
    "StartoverWindowSeconds": 0,
    "Tags": {
        "region": "west",
        "media": "sports"
    },
    "TimeDelaySeconds": 0,
    "Url": "",
    "Whitelist": []
}
```

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的檢視單一端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeOriginEndpoint。

`list-channels`

以下程式碼範例顯示如何使用 `list-channels`。

AWS CLI

列出所有頻道

下列`list-channels`命令會列出目前 AWS 帳戶上設定的所有頻道。

```
aws mediapackage list-channels
```

輸出：

```
{
    "Channels": [
        {
            "Arn": "arn:aws:mediapackage:us-
west-2:111222333:channels/584797f1740548c389a273585dd22a63",
```

```
            "HlsIngest": {
                "IngestEndpoints": [
                    {
                        "Id": "584797f1740548c389a273585dd22a63",
                        "Password": "webdavgeneratedpassword1",
                        "Url": "https://9be9c4405c474882.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/584797f1740548c389a273585dd22a63/channel",
                        "Username": "webdavgeneratedusername1"
                    },
                    {
                        "Id": "7d187c8616fd455f88aaa5a9fcf74442",
                        "Password": "webdavgeneratedpassword2",
                        "Url": "https://7bf454c57220328d.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/7d187c8616fd455f88aaa5a9fcf74442/channel",
                        "Username": "webdavgeneratedusername2"
                    }
                ]
            },
            "Id": "test",
            "Tags": {}
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Elemental MediaPackage 使用者指南》中的檢視頻道詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListChannels。

`list-origin-endpoints`

以下程式碼範例顯示如何使用 `list-origin-endpoints`。

AWS CLI

列出頻道上的所有原始端點

下列`list-origin-endpoints`命令列出在名為 的頻道上設定的所有原始伺服器端點`test`。

```
aws mediapackage list-origin-endpoints \
    --channel-id test
```

輸出：


```
{
    "OriginEndpoints": [
        {
            "Arn": "arn:aws:mediapackage:us-
west-2:111222333:origin_endpoints/247cff871f2845d3805129be22f2c0a2",
            "ChannelId": "test",
            "DashPackage": {
                "ManifestLayout": "FULL",
                "ManifestWindowSeconds": 60,
                "MinBufferTimeSeconds": 30,
                "MinUpdatePeriodSeconds": 15,
                "PeriodTriggers": [],
                "Profile": "NONE",
                "SegmentDurationSeconds": 2,
                "SegmentTemplateFormat": "NUMBER_WITH_TIMELINE",
                "StreamSelection": {
                    "MaxVideoBitsPerSecond": 2147483647,
                    "MinVideoBitsPerSecond": 0,
                    "StreamOrder": "ORIGINAL"
                },
                "SuggestedPresentationDelaySeconds": 25
            },
            "Id": "tester2",
            "ManifestName": "index",
            "StartoverWindowSeconds": 0,
            "Tags": {},
            "TimeDelaySeconds": 0,
            "Url": "https://8343f7014c0ea438.mediapackage.us-west-2.amazonaws.com/
out/v1/247cff871f2845d3805129be22f2c0a2/index.mpd",
            "Whitelist": []
        },
        {
            "Arn": "arn:aws:mediapackage:us-
west-2:111222333:origin_endpoints/869e237f851549e9bcf10e3bc2830839",
            "ChannelId": "test",
            "HlsPackage": {
                "AdMarkers": "NONE",
                "IncludeIframeOnlyStream": false,
                "PlaylistType": "EVENT",
                "PlaylistWindowSeconds": 60,
                "ProgramDateTimeIntervalSeconds": 0,
                "SegmentDurationSeconds": 6,
```

```
                "StreamSelection": {
                    "MaxVideoBitsPerSecond": 2147483647,
                    "MinVideoBitsPerSecond": 0,
                    "StreamOrder": "ORIGINAL"
                },
                "UseAudioRenditionGroup": false
            },
            "Id": "tester",
            "ManifestName": "index",
            "StartoverWindowSeconds": 0,
            "Tags": {},
            "TimeDelaySeconds": 0,
            "Url": "https://8343f7014c0ea438.mediapackage.us-west-2.amazonaws.com/
out/v1/869e237f851549e9bcf10e3bc2830839/index.m3u8",
            "Whitelist": []
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的檢視與頻道相關聯的所有端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListOriginEndpoints。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出指派給資源的標籤

下列list-tags-for-resource命令會列出指派給指定資源的標籤。

```
aws mediapackage list-tags-for-resource \
    --resource-arn arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0
```

輸出：

```
{
    "Tags": {
```

```
        "region": "west"
    }
}
```

如需詳細資訊，請參閱《AWS Elemental MediaPackage 使用者指南》中的 Elemental MediaPackage 中的標記資源。 AWS MediaPackage

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

rotate-ingest-endpoint-credentials

以下程式碼範例顯示如何使用 rotate-ingest-endpoint-credentials。

AWS CLI

輪換擷取憑證

下列rotate-ingest-endpoint-credentials命令會輪換指定擷取端點的 WebDAV 使用者名稱和密碼。

```
aws mediapackage rotate-ingest-endpoint-credentials \
    --id test \
    --ingest-endpoint-id 584797f1740548c389a273585dd22a63
```

輸出：

```
{
    "Arn": "arn:aws:mediapackage:us-
west-2:111222333:channels/584797f1740548c389a273585dd22a63",
    "HlsIngest": {
        "IngestEndpoints": [
            {
                "Id": "584797f1740548c389a273585dd22a63",
                "Password": "webdavregeneratedpassword1",
                "Url": "https://9be9c4405c474882.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/584797f1740548c389a273585dd22a63/channel",
                "Username": "webdavregeneratedusername1"
            },
            {
                "Id": "7d187c8616fd455f88aaa5a9fcf74442",
                "Password": "webdavgeneratedpassword2",
```

```
                "Url": "https://7bf454c57220328d.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/7d187c8616fd455f88aaa5a9fcf74442/channel",
                "Username": "webdavgeneratedusername2"
            }
        ]
    },
    "Id": "test",
    "Tags": {}
}
```

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的在輸入 URL 上輪換登入資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RotateIngestEndpointCredentials。

`tag-resource`

以下程式碼範例顯示如何使用 `tag-resource`。

AWS CLI

將標籤加入資源

下列`tag-resource`命令會將`region=west`金鑰和值對新增至指定的資源。

```
aws mediapackage tag-resource \
    --resource-arn arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0 \
    --tags region=west
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Elemental MediaPackage 使用者指南》中的 Elemental MediaPackage 中的標記資源。 AWS MediaPackage

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

`untag-resource`

以下程式碼範例顯示如何使用 `untag-resource`。

AWS CLI

從資源移除標籤

下列untag-resource命令region會從指定的頻道移除具有 金鑰的標籤。

```
aws mediapackage untag-resource \
    --resource-arn arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0 \
    --tag-keys region
```

如需詳細資訊，請參閱《AWS Elemental MediaPackage 使用者指南》中的 Elemental MediaPackage 中的標記資源。 AWS MediaPackage

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-channel

以下程式碼範例顯示如何使用 update-channel。

AWS CLI

更新頻道

下列update-channel命令會更新名為 的頻道sportschannel，以包含描述 24x7 sports。

```
aws mediapackage update-channel \
    --id sportschannel \
    --description "24x7 sports"
```

輸出：

```
{
    "Arn": "arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0",
    "Description": "24x7 sports",
    "HlsIngest": {
        "IngestEndpoints": [
            {
                "Id": "6d345804ec3f46c9b454a91d4a80d0e0",
                "Password": "generatedwebdavpassword1",
```

```
                "Url": "https://f31c86aed53b815a.mediapackage.us-
west-2.amazonaws.com/in/
v2/6d345804ec3f46c9b454a91d4a80d0e0/6d345804ec3f46c9b454a91d4a80d0e0/channel",
                "Username": "generatedwebdavusername1"
            },
            {
                "Id": "2daa32878af24803b24183727211b8ff",
                "Password": "generatedwebdavpassword2",
                "Url": "https://6ebbe7e04c4b0afa.mediapackage.us-
west-2.amazonaws.com/in/
v2/6d345804ec3f46c9b454a91d4a80d0e0/2daa32878af24803b24183727211b8ff/channel",
                "Username": "generatedwebdavusername2"
            }
        ]
    },
    "Id": "sportschannel",
    "Tags": {}
}
```

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的編輯頻道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateChannel。

update-origin-endpoint

以下程式碼範例顯示如何使用 update-origin-endpoint。

AWS CLI

更新原始伺服器端點

下列update-origin-endpoint命令會更新名為 的原始伺服器端點cmaf_sports。它會將時間延遲變更為0秒。

```
aws mediapackage update-origin-endpoint \
    --id cmaf_sports \
    --time-delay-seconds 0
```

輸出：

```
{
    "Arn": "arn:aws:mediapackage:us-
west-2:111222333:origin_endpoints/1dc6718be36f4f34bb9cd86bc50925e6",
```

```
    "ChannelId": "sportschannel",
    "CmafPackage": {
        "HlsManifests": [
            {
                "AdMarkers": "NONE",
                "Id": "cmaf_sports_endpoint",
                "IncludeIframeOnlyStream": false,
                "PlaylistType": "EVENT",
                "PlaylistWindowSeconds": 60,
                "ProgramDateTimeIntervalSeconds": 0,
                "Url": "https://c4af3793bf76b33c.mediapackage.us-
west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/cmaf_sports_endpoint/
index.m3u8"
            }
        ],
        "SegmentDurationSeconds": 2,
        "SegmentPrefix": "sportschannel"
    },
    "Id": "cmaf_sports",
    "ManifestName": "index",
    "StartoverWindowSeconds": 0,
    "Tags": {
        "region": "west",
        "media": "sports"
    },
    "TimeDelaySeconds": 0,
    "Url": "",
    "Whitelist": []
}
```

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的編輯端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateOriginEndpoint。

使用 的 MediaPackage VOD 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 MediaPackage VOD 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

create-asset

以下程式碼範例顯示如何使用 create-asset。

AWS CLI

建立資產

下列create-asset範例會在目前 AWS 帳戶中建立名為 Chicken_Asset的資產。資產會將檔案擷取30sec_chicken.smil至 MediaPackage。

```
aws mediapackage-vod create-asset \
    --id chicken_asset \
    --packaging-group-id hls_chicken_gp \
    --source-role-arn arn:aws:iam::111122223333:role/EMP_Vod \
    --source-arn arn:aws:s3::111122223333:video-bucket/A/30sec_chicken.smil
```

輸出：

```
{
    "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:assets/chicken_asset",
    "Id":"chicken_asset",
    "PackagingGroupId":"hls_chicken_gp",
    "SourceArn":"arn:aws:s3::111122223333:video-bucket/A/30sec_chicken.smil",
    "SourceRoleArn":"arn:aws:iam::111122223333:role/EMP_Vod",
    "EgressEndpoints":[
        {
            "PackagingConfigurationId":"New_config_1",
            "Url":"https://c75ea2668ab49d02bca7ae10ef31c59e.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/
v1/6644b55df1744261ab3732a8e5cdaf07/904b06a58c7645e08d57d40d064216ac/
f5b2e633ff4942228095d164c10074f3/index.m3u8"
        },
        {
            "PackagingConfigurationId":"new_hls",
```

```
            "Url":" https://c75ea2668ab49d02bca7ae10ef31c59e.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/v1/6644b55df1744261ab3732a8e5cdaf07/
fe8f1f00a80e424cb4f8da4095835e9e/7370ec57432343af816332356d2bd5c6/string.m3u8"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的擷取資產。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAsset。

create-packaging-configuration

以下程式碼範例顯示如何使用 create-packaging-configuration。

AWS CLI

建立封裝組態

下列create-packaging-configuration範例會在名為 的封裝群組new_hls中建立名為 的封裝組態hls_chicken。此範例使用名為 之磁碟上的檔案hls_pc.json來提供詳細資訊。

```
aws mediapackage-vod create-packaging-configuration \
    --id new_hls \
    --packaging-group-id hls_chicken \
    --hls-package file://hls_pc.json
```

hls_pc.json 的內容：

```
{
    "HlsManifests":[
        {
            "AdMarkers":"NONE",
            "IncludeIframeOnlyStream":false,
            "ManifestName":"string",
            "ProgramDateTimeIntervalSeconds":60,
            "RepeatExtXKey":true,
            "StreamSelection":{
                "MaxVideoBitsPerSecond":1000,
                "MinVideoBitsPerSecond":0,
                "StreamOrder":"ORIGINAL"
            }
        }
```

```
    ],
    "SegmentDurationSeconds":6,
    "UseAudioRenditionGroup":false
}
```

輸出：

```
{
    "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-configurations/
new_hls",
    "Id":"new_hls",
    "PackagingGroupId":"hls_chicken",
    "HlsManifests":{
        "SegmentDurationSeconds":6,
        "UseAudioRenditionGroup":false,
        "HlsMarkers":[
            {
                "AdMarkers":"NONE",
                "IncludeIframeOnlyStream":false,
                "ManifestName":"string",
                "ProgramDateTimeIntervalSeconds":60,
                "RepeatExtXKey":true,
                "StreamSelection":{
                    "MaxVideoBitsPerSecond":1000,
                    "MinVideoBitsPerSecond":0,
                    "StreamOrder":"ORIGINAL"
                }
            }
        ]
    }
}
```

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的建立封裝組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePackagingConfiguration。

create-packaging-group

以下程式碼範例顯示如何使用 create-packaging-group。

AWS CLI

建立封裝群組

下列create-packaging-group範例列出目前 AWS 帳戶中設定的所有封裝群組。

```
aws mediapackage-vod create-packaging-group \
    --id hls_chicken
```

輸出：

```
{
    "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-groups/
hls_chicken",
    "Id": "hls_chicken"
}
```

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的建立封裝群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePackagingGroup。

delete-asset

以下程式碼範例顯示如何使用 delete-asset。

AWS CLI

刪除資產

下列delete-asset範例會刪除名為 的資產30sec_chicken。

```
aws mediapackage-vod delete-asset \
    --id 30sec_chicken
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的刪除資產。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAsset。

delete-packaging-configuration

以下程式碼範例顯示如何使用 delete-packaging-configuration。

AWS CLI

刪除封裝組態

下列delete-packaging-configuration範例會刪除名為 的封裝組態CMAF。

```
aws mediapackage-vod delete-packaging-configuration \
    --id CMAF
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的刪除封裝組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePackagingConfiguration。

delete-packaging-group

以下程式碼範例顯示如何使用 delete-packaging-group。

AWS CLI

刪除封裝群組

下列delete-packaging-group範例會刪除名為 的封裝群組Dash_widevine。

```
aws mediapackage-vod delete-packaging-group \
    --id Dash_widevine
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的刪除封裝群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePackagingGroup。

describe-asset

以下程式碼範例顯示如何使用 describe-asset。

AWS CLI

描述資產

下列describe-asset範例顯示名為 之資產的所有詳細資訊30sec_chicken。

```
aws mediapackage-vod describe-asset \
```

```
    --id 30sec_chicken
```

輸出：

```
{
    "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:assets/30sec_chicken",
    "Id":"30sec_chicken",
    "PackagingGroupId":"Packaging_group_1",
    "SourceArn":"arn:aws:s3::111122223333:video-bucket/A/30sec_chicken.smil",
    "SourceRoleArn":"arn:aws:iam::111122223333:role/EMP_Vod",
    "EgressEndpoints":[
        {
            "PackagingConfigurationId":"DASH",
            "Url":"https://a5f46a44118ba3e3724ef39ef532e701.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/v1/
aad7962c569946119c2d5a691be5663c/66c25aff456d463aae0855172b3beb27/4ddfda6da17c4c279a1b8401cb
index.mpd"
        },
        {
            "PackagingConfigurationId":"HLS",
            "Url":"https://a5f46a44118ba3e3724ef39ef532e701.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/v1/
aad7962c569946119c2d5a691be5663c/6e5bf286a3414254a2bf0d22ae148d7e/06b5875b4d004c3cbdc4da2dc4
index.m3u8"
        },
        {
            "PackagingConfigurationId":"CMAF",
            "Url":"https://a5f46a44118ba3e3724ef39ef532e701.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/v1/
aad7962c569946119c2d5a691be5663c/628fb5d8d89e4702958b020af27fde0e/05eb062214064238ad6330a443
index.m3u8"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Elemental MediaPackage 使用者指南》中的檢視資產詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAsset。

describe-packaging-configuration

以下程式碼範例顯示如何使用 describe-packaging-configuration。

AWS CLI

描述封裝組態

下列describe-packaging-configuration範例顯示名為 之封裝組態的所有詳細資訊DASH。

```
aws mediapackage-vod describe-packaging-configuration \
    --id DASH
```

輸出：

```
{
    "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-configurations/
DASH",
    "Id":"DASH",
    "PackagingGroupId":"Packaging_group_1",
    "DashPackage":[
        {
            "SegmentDurationSeconds":"2"
        },
        {
            "DashManifests":{
                "ManifestName":"index",
                "MinBufferTimeSeconds":"30",
                "Profile":"NONE"
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的檢視封裝組態詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePackagingConfiguration。

describe-packaging-group

以下程式碼範例顯示如何使用 describe-packaging-group。

AWS CLI

描述封裝群組

下列describe-packaging-group範例顯示名為 之封裝群組的所有詳細資訊Packaging_group_1。

```
aws mediapackage-vod describe-packaging-group \
    --id Packaging_group_1
```

輸出：

```
{
    "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-groups/
Packaging_group_1",
    "Id": "Packaging_group_1"
}
```

如需詳細資訊，請參閱《AWS Elemental MediaPackage 使用者指南》中的檢視封裝群組詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePackagingGroup。

list-assets

以下程式碼範例顯示如何使用 list-assets。

AWS CLI

列出所有資產

下列list-assets範例列出目前 AWS 帳戶中設定的所有資產。

```
aws mediapackage-vod list-assets
```

輸出：

```
{
    "Assets": [
        "Arn": "arn:aws:mediapackage-vod:us-
west-2:111122223333:assets/30sec_chicken",
        "Id": "30sec_chicken",
        "PackagingGroupId": "Packaging_group_1",
        "SourceArn": "arn:aws:s3::111122223333:video-bucket/A/30sec_chicken.smil",
        "SourceRoleArn": "arn:aws:iam::111122223333:role/EMP_Vod"
    ]
```

```
}
```

如需詳細資訊，請參閱《AWS Elemental MediaPackage 使用者指南》中的檢視資產詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAssets。

list-packaging-configurations

以下程式碼範例顯示如何使用 list-packaging-configurations。

AWS CLI

列出所有封裝組態

下列list-packaging-configurations範例列出在名為 的封裝群組上設定的所有封裝組態Packaging_group_1。

```
aws mediapackage-vod list-packaging-configurations \
    --packaging-group-id Packaging_group_1
```

輸出：

```
{
    "PackagingConfigurations":[
        {
            "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
configurations/CMAF",
            "Id":"CMAF",
            "PackagingGroupId":"Packaging_group_1",
            "CmafPackage":[
                {
                    "SegmentDurationSeconds":"2"
                },
                {
                    "HlsManifests":{
                        "AdMarkers":"NONE",
                        "RepeatExtXKey":"False",
                        "ManifestName":"index",
                        "ProgramDateTimeIntervalSeconds":"0",
                        "IncludeIframeOnlyStream":"False"
                    }
                }
            ]
```

```
        },
        {
            "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
configurations/DASH",
            "Id":"DASH",
            "PackagingGroupId":"Packaging_group_1",
            "DashPackage":[
                {
                    "SegmentDurationSeconds":"2"
                },
                {
                    "DashManifests":{
                        "ManifestName":"index",
                        "MinBufferTimeSeconds":"30",
                        "Profile":"NONE"
                    }
                }
            ]
        },
        {
            "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
configurations/HLS",
            "Id":"HLS",
            "PackagingGroupId":"Packaging_group_1",
            "HlsPackage":[
                {
                    "SegmentDurationSeconds":"6",
                    "UseAudioRenditionGroup":"False"
                },
                {
                    "HlsManifests":{
                        "AdMarkers":"NONE",
                        "RepeatExtXKey":"False",
                        "ManifestName":"index",
                        "ProgramDateTimeIntervalSeconds":"0",
                        "IncludeIframeOnlyStream":"False"
                    }
                }
            ]
        },
        {
            "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
configurations/New_config_0_copy",
            "Id":"New_config_0_copy",
```

```
            "PackagingGroupId":"Packaging_group_1",
            "HlsPackage":[
                {
                    "SegmentDurationSeconds":"6",
                    "UseAudioRenditionGroup":"False"
                },
                {
                    "Encryption":{
                        "EncryptionMethod":"AWS_128",
                        "SpekeKeyProvider":{
                           "RoleArn":"arn:aws:iam:111122223333::role/SPEKERole",
                            "Url":"https://lfgubdvs97.execute-api.us-
west-2.amazonaws.com/EkeStage/copyProtection/",
                            "SystemIds":[
                                "81376844-f976-481e-a84e-cc25d39b0b33"
                            ]
                        }
                    },
                    {
                        "HlsManifests":{
                            "AdMarkers":"NONE",
                            "RepeatExtXKey":"False",
                            "ManifestName":"index",
                            "ProgramDateTimeIntervalSeconds":"0",
                            "IncludeIframeOnlyStream":"False"
                        }
                    }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的檢視封裝組態詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPackagingConfigurations。

`list-packaging-groups`

以下程式碼範例顯示如何使用 `list-packaging-groups`。

AWS CLI

列出所有封裝群組

下列list-packaging-groups範例列出目前 AWS 帳戶中設定的所有封裝群組。

```
aws mediapackage-vod list-packaging-groups
```

輸出：

```
{
    "PackagingGroups": [
        {
            "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
groups/Dash_widevine",
            "Id": "Dash_widevine"
        },
        {
            "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
groups/Encrypted_HLS",
            "Id": "Encrypted_HLS"
        },
        {
            "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
groups/Packaging_group_1",
            "Id": "Packaging_group_1"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Elemental MediaPackage 使用者指南中的檢視封裝群組詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPackagingGroups。

使用 的 MediaStore 資料平面範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 MediaStore Data Plane 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

delete-object

以下程式碼範例顯示如何使用 delete-object。

AWS CLI

刪除物件

下列delete-object範例會刪除指定的物件。

```
aws mediastore-data delete-object \
    --endpoint=https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path=/folder_name/README.md
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的刪除物件。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteObject。

describe-object

以下程式碼範例顯示如何使用 describe-object。

AWS CLI

檢視物件的標頭

下列describe-object範例顯示指定路徑中物件的標頭。

```
aws mediastore-data describe-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path events/baseball/setup.jpg
```

輸出：

```
{
    "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT",
    "ContentType": "image/jpeg",
    "ContentLength": "3860266",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
```

```
}
```

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的檢視物件的詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeObject。

get-object

以下程式碼範例顯示如何使用 get-object。

AWS CLI

範例 1：下載整個物件

下列get-object範例會下載指定的物件。

```
aws mediastore-data get-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path events/baseball/setup.jpg setup.jpg
```

輸出：

```
{
    "ContentType": "image/jpeg",
    "StatusCode": 200,
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
    "ContentLength": "3860266",
    "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT"
}
```

範例 2：下載物件的一部分

下列get-object範例會下載 物件的指定部分。

```
aws mediastore-data get-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path events/baseball/setup.jpg setup.jpg \
    --range "bytes=0-100"
```

輸出：

```
{
```

```
    "StatusCode": 206,
    "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT",
    "ContentType": "image/jpeg",
    "ContentRange": "bytes 0-100/3860266",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
    "ContentLength": "101"
}
```

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的下載物件。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetObject。

`list-items`

以下程式碼範例顯示如何使用 `list-items`。

AWS CLI

範例 1：檢視存放在容器中的項目（物件和資料夾）清單

下列`list-items`範例顯示存放在指定容器中的項目（物件和資料夾）清單。

```
aws mediastore-data list-items \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com
```

輸出：

```
{
    "Items": [
        {
            "Type": "OBJECT",
            "ContentLength": 3784,
            "Name": "setup.jpg",
            "ETag":
 "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
            "ContentType": "image/jpeg",
            "LastModified": 1563571859.379
        },
        {
            "Type": "FOLDER",
            "Name": "events"
        }
    ]
```

```
}
```

範例 2：檢視存放在資料夾中的項目清單（物件和資料夾）

下列list-items範例顯示存放在指定資料夾中的項目（物件和資料夾）清單。

```
aws mediastore-data list-items \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path events/baseball
```

輸出：

```
{
    "Items": [
        {
            "ETag":
 "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
            "ContentType": "image/jpeg",
            "Type": "OBJECT",
            "ContentLength": 3860266,
            "LastModified": 1563573031.872,
            "Name": "setup.jpg"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的檢視物件清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListItems。

put-object

以下程式碼範例顯示如何使用 put-object。

AWS CLI

範例 1：將物件上傳至容器

下列put-object範例會將物件上傳至指定的容器。

```
aws mediastore-data put-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --body ReadMe.md \
```

```
    --path ReadMe.md \
    --cache-control "max-age=6, public" \
    --content-type binary/octet-stream
```

輸出：

```
{
    "ContentSHA256":
 "f29bc64a9d3732b4b9035125fdb3285f5b6455778edca72414671e0ca3b2e0de",
    "StorageClass": "TEMPORAL",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
}
```

範例 2：將物件上傳至容器內的資料夾

下列put-object範例會將物件上傳至容器內指定的資料夾。

```
aws mediastore-data put-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --body ReadMe.md \
    --path /september-events/ReadMe.md \
    --cache-control "max-age=6, public" \
    --content-type binary/octet-stream
```

輸出：

```
{
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
    "ContentSHA256":
 "f29bc64a9d3732b4b9035125fdb3285f5b6455778edca72414671e0ca3b2e0de",
    "StorageClass": "TEMPORAL"
}
```

如需詳細資訊，請參閱 AWS Elemental MediaStore 使用者指南中的上傳物件。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutObject。

使用 的 MediaTailor 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 MediaTailor 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`delete-playback-configuration`

以下程式碼範例顯示如何使用 `delete-playback-configuration`。

AWS CLI

刪除組態

以下內容會`delete-playback-configuration`刪除名為 的組態`campaign_short`。

```
aws mediatailor delete-playback-configuration \
    --name campaign_short
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Elemental MediaTailor 使用者指南中的刪除組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePlaybackConfiguration。

`get-playback-configuration`

以下程式碼範例顯示如何使用 `get-playback-configuration`。

AWS CLI

描述組態

以下`get-playback-configuration`顯示名為 之組態的所有詳細資訊`west_campaign`。

```
aws mediatailor get-playback-configuration \
    --name west_campaign
```

輸出：

```
{
     "AdDecisionServerUrl": "http://your.ads.url",
     "CdnConfiguration": {},
     "DashConfiguration": {
         "ManifestEndpointPrefix":
 "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
dash/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/",
         "MpdLocation": "EMT_DEFAULT",
         "OriginManifestType": "MULTI_PERIOD"
     },
     "HlsConfiguration": {
         "ManifestEndpointPrefix":
 "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
master/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/"
     },
     "Name": "west_campaign",
     "PlaybackConfigurationArn": "arn:aws:mediatailor:us-
west-2:123456789012:playbackConfiguration/west_campaign",
     "PlaybackEndpointPrefix":
 "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com",
     "SessionInitializationEndpointPrefix":
 "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
session/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/",
     "Tags": {},
     "VideoContentSourceUrl": "https://8343f7014c0ea438.mediapackage.us-
west-2.amazonaws.com/out/v1/683f0f2ff7cd43a48902e6dcd5e16dcf/index.m3u8"
}
```

如需詳細資訊，請參閱 AWS Elemental MediaTailor 使用者指南中的檢視組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPlaybackConfiguration。

list-playback-configurations

以下程式碼範例顯示如何使用 list-playback-configurations。

AWS CLI

列出所有組態

以下list-playback-configurations顯示目前 AWS 帳戶上組態的所有詳細資訊。

```
aws mediatailor list-playback-configurations
```

輸出：

```
{
    "Items": [
        {
            "AdDecisionServerUrl": "http://your.ads.url",
            "CdnConfiguration": {},
            "DashConfiguration": {
                "ManifestEndpointPrefix":
 "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
dash/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/",
                "MpdLocation": "EMT_DEFAULT",
                "OriginManifestType": "MULTI_PERIOD"
            },
            "HlsConfiguration": {
                "ManifestEndpointPrefix":
 "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
master/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/"
            },
            "Name": "west_campaign",
            "PlaybackConfigurationArn": "arn:aws:mediatailor:us-
west-2:123456789012:playbackConfiguration/west_campaign",
            "PlaybackEndpointPrefix":
 "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com",
            "SessionInitializationEndpointPrefix":
 "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
session/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/",
            "Tags": {},
            "VideoContentSourceUrl": "https://8343f7014c0ea438.mediapackage.us-
west-2.amazonaws.com/out/v1/683f0f2ff7cd43a48902e6dcd5e16dcf/index.m3u8"
        },
        {
            "AdDecisionServerUrl": "http://your.ads.url",
            "CdnConfiguration": {},
            "DashConfiguration": {
                "ManifestEndpointPrefix":
 "https://73511f91d6a24ca2b93f3cf1d7cedd67.mediatailor.us-west-2.amazonaws.com/v1/
dash/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/sports_campaign/",
                "MpdLocation": "DISABLED",
                "OriginManifestType": "MULTI_PERIOD"
            },
```

```
            "HlsConfiguration": {
                "ManifestEndpointPrefix":
 "https://73511f91d6a24ca2b93f3cf1d7cedd67.mediatailor.us-west-2.amazonaws.com/v1/
master/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/sports_campaign/"
            },
            "Name": "sports_campaign",
            "PlaybackConfigurationArn": "arn:aws:mediatailor:us-
west-2:123456789012:playbackConfiguration/sports_campaign",
            "PlaybackEndpointPrefix":
 "https://73511f91d6a24ca2b93f3cf1d7cedd67.mediatailor.us-west-2.amazonaws.com",
            "SessionInitializationEndpointPrefix":
 "https://73511f91d6a24ca2b93f3cf1d7cedd67.mediatailor.us-west-2.amazonaws.com/v1/
session/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/sports_campaign/",
            "SlateAdUrl": "http://s3.bucket/slate_ad.mp4",
            "Tags": {},
            "VideoContentSourceUrl": "https://c4af3793bf76b33c.mediapackage.us-
west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/sports_endpoint/
index.m3u8"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Elemental MediaTailor 使用者指南中的檢視組態<https://docs.aws.amazon.com/mediatailor/latest/ug/configurations-view.html>。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPlaybackConfigurations。

put-playback-configuration

以下程式碼範例顯示如何使用 put-playback-configuration。

AWS CLI

建立組態

以下內容put-playback-configuration會建立名為 的組態campaign_short。

```
aws mediatailor put-playback-configuration \
    --name campaign_short \
    --ad-decision-server-url http://your.ads.url \
    --video-content-source-url http://video.bucket/index.m3u8
```

輸出：

```
{
    "AdDecisionServerUrl": "http://your.ads.url",
    "CdnConfiguration": {},
    "DashConfiguration": {
         "ManifestEndpointPrefix":
 "https://13484114d38f4383bc0d6a7cb879bd00.mediatailor.us-west-2.amazonaws.com/v1/
dash/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/campaign_short/",
         "MpdLocation": "EMT_DEFAULT",
         "OriginManifestType": "MULTI_PERIOD"
    },
    "HlsConfiguration": {
        "ManifestEndpointPrefix":
 "https://13484114d38f4383bc0d6a7cb879bd00.mediatailor.us-west-2.amazonaws.com/v1/
master/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/campaign_short/"
    },
    "Name": "campaign_short",
    "PlaybackConfigurationArn": "arn:aws:mediatailor:us-
west-2:123456789012:playbackConfiguration/campaign_short",
    "PlaybackEndpointPrefix":
 "https://13484114d38f4383bc0d6a7cb879bd00.mediatailor.us-west-2.amazonaws.com",
    "SessionInitializationEndpointPrefix":
 "https://13484114d38f4383bc0d6a7cb879bd00.mediatailor.us-west-2.amazonaws.com/v1/
session/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/campaign_short/",
    "Tags": {},
    "VideoContentSourceUrl": "http://video.bucket/index.m3u8"
}
```

如需詳細資訊，請參閱 AWS Elemental MediaTailor 使用者指南中的建立組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutPlaybackConfiguration。

使用 的 MemoryDB 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 MemoryDB 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

copy-snapshot

以下程式碼範例顯示如何使用 copy-snapshot。

AWS CLI

複製快照

下列copy-snapshot範例會建立快照的副本。

```
aws memorydb copy-snapshot \
    --source-snapshot-name my-cluster-snapshot \
    --target-snapshot-name my-cluster-snapshot-copy
```

輸出

```
{
    "Snapshot": {
        "Name": "my-cluster-snapshot-copy",
        "Status": "creating",
        "Source": "manual",
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:snapshot/my-cluster-
snapshot-copy",
        "ClusterConfiguration": {
            "Name": "my-cluster",
            "Description": " ",
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "Port": 6379,
            "ParameterGroupName": "default.memorydb-redis6",
            "SubnetGroupName": "my-sg",
            "VpcId": "vpc-xx2574fc",
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "04:30-05:30",
            "NumShards": 2
        }
    }
```

```
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的複製快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CopySnapshot。

create-acl

以下程式碼範例顯示如何使用 create-acl。

AWS CLI

建立 ACL

下列create-acl範例會建立新的存取控制清單。

```
aws memorydb create-acl \
    --acl-name "new-acl-1" \
    --user-names "my-user"
```

輸出：

```
{
    "ACL": {
        "Name": "new-acl-1",
        "Status": "creating",
        "UserNames": [
            "my-user"
        ],
        "MinimumEngineVersion": "6.2",
        "Clusters": [],
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:acl/new-acl-1"
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的使用存取控制清單驗證使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAcl。

create-cluster

以下程式碼範例顯示如何使用 create-cluster。

AWS CLI

建立叢集

下列create-cluster範例會建立新的叢集。

```
aws memorydb create-cluster \
    --cluster-name my-new-cluster \
    --node-type db.r6g.large \
    --acl-name my-acl \
    --subnet-group my-sg
```

輸出：

```
{
    "Cluster": {
        "Name": "my-new-cluster",
        "Status": "creating",
        "NumberOfShards": 1,
        "AvailabilityMode": "MultiAZ",
        "ClusterEndpoint": {
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:cluster/my-new-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "sat:10:00-sat:11:00",
        "SnapshotWindow": "07:30-08:30",
        "ACLName": "my-acl",
        "AutoMinorVersionUpgrade": true
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的管理叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCluster。

create-parameter-group

以下程式碼範例顯示如何使用 create-parameter-group。

AWS CLI

建立參數群組

下列create-parameter-group範例會建立參數群組。

```
aws memorydb create-parameter-group \
    --parameter-group-name myRedis6x \
    --family memorydb_redis6 \
    --description "my-parameter-group"
```

輸出：

```
{
    "ParameterGroup": {
        "Name": "myredis6x",
        "Family": "memorydb_redis6",
        "Description": "my-parameter-group",
        "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:parametergroup/myredis6x"
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的建立參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateParameterGroup。

create-snapshot

以下程式碼範例顯示如何使用 create-snapshot。

AWS CLI

建立快照

下列create-snapshot範例會建立快照。

```
aws memorydb create-snapshot \
    --cluster-name my-cluster \
```

```
    --snapshot-name my-cluster-snapshot
```

輸出：

```
{
    "Snapshot": {
        "Name": "my-cluster-snapshot1",
        "Status": "creating",
        "Source": "manual",
        "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:snapshot/my-cluster-
snapshot",
        "ClusterConfiguration": {
            "Name": "my-cluster",
            "Description": "",
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "Port": 6379,
            "ParameterGroupName": "default.memorydb-redis6",
            "SubnetGroupName": "my-sg",
            "VpcId": "vpc-862xxxxc",
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "04:30-05:30",
            "NumShards": 2
        }
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的製作手動快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSnapshot。

create-subnet-group

以下程式碼範例顯示如何使用 create-subnet-group。

AWS CLI

建立子網路群組

下列create-subnet-group範例會建立子網路群組。

```
aws memorydb create-subnet-group \
```

```
    --subnet-group-name mysubnetgroup \
    --description "my subnet group" \
    --subnet-ids subnet-5623xxxx
```

輸出：

```
{
    "SubnetGroup": {
        "Name": "mysubnetgroup",
        "Description": "my subnet group",
        "VpcId": "vpc-86257xxx",
        "Subnets": [
            {
                "Identifier": "subnet-5623xxxx",
                "AvailabilityZone": {
                    "Name": "us-east-1a"
                }
            }
        ],
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:subnetgroup/mysubnetgroup"
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的建立子網路群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSubnetGroup。

create-user

以下程式碼範例顯示如何使用 create-user。

AWS CLI

建立使用者

下列create-user範例會建立新的使用者。

```
aws memorydb create-user \
    --user-name user-name-1 \
    --access-string "~objects:* ~items:* ~public:*" \
     --authentication-mode \
         Passwords="enterapasswordhere",Type=password
```

輸出：

```
{
    "User": {
        "Name": "user-name-1",
        "Status": "active",
        "AccessString": "off ~objects:* ~items:* ~public:* resetchannels -@all",
        "ACLNames": [],
        "MinimumEngineVersion": "6.2",
        "Authentication": {
            "Type": "password",
            "PasswordCount": 1
        },
        "ARN": "arn:aws:memorydb:us-west-2:491658xxxxxx:user/user-name-1"
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的使用存取控制清單驗證使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateUser。

delete-acl

以下程式碼範例顯示如何使用 delete-acl。

AWS CLI

刪除 ACL

下列delete-acl範例會刪除存取控制清單。

```
aws memorydb delete-acl \
    --acl-name "new-acl-1"
```

輸出：

```
{
    "ACL": {
        "Name": "new-acl-1",
        "Status": "deleting",
        "UserNames": [
            "pat"
        ],
```

```
        "MinimumEngineVersion": "6.2",
        "Clusters": [],
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:acl/new-acl-1"
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的使用存取控制清單驗證使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAcl。

delete-cluster

以下程式碼範例顯示如何使用 delete-cluster。

AWS CLI

刪除叢集

下列delete-cluster範例會刪除叢集。

```
aws memorydb delete-cluster \
    --cluster-name my-new-cluster
```

輸出：

```
{
    "Cluster": {
        "Name": "my-new-cluster",
        "Status": "deleting",
        "NumberOfShards": 1,
        "ClusterEndpoint": {
            "Address": "clustercfg.my-new-cluster.xxxxx.memorydb.us-
east-1.amazonaws.com",
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-new-cluster",
```

```
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "sat:10:00-sat:11:00",
        "SnapshotWindow": "07:30-08:30",
        "AutoMinorVersionUpgrade": true
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的刪除叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCluster。

delete-parameter-group

以下程式碼範例顯示如何使用 delete-parameter-group。

AWS CLI

刪除參數群組

下列delete-parameter-group範例會刪除參數群組。

```
aws memorydb delete-parameter-group \
    --parameter-group-name myRedis6x
```

輸出：

```
{
    "ParameterGroup": {
        "Name": "myredis6x",
        "Family": "memorydb_redis6",
        "Description": "my-parameter-group",
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/myredis6x"
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的刪除參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteParameterGroup。

delete-snapshot

以下程式碼範例顯示如何使用 delete-snapshot。

AWS CLI

刪除快照

下列delete-snapshot範例會刪除快照。

```
aws memorydb delete-snapshot \
    --snapshot-name my-cluster-snapshot
```

輸出：

```
{
    "Snapshot": {
        "Name": "my-cluster-snapshot",
        "Status": "deleting",
        "Source": "manual",
        "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:snapshot/my-cluster-
snapshot",
        "ClusterConfiguration": {
            "Name": "my-cluster",
            "Description": "",
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "Port": 6379,
            "ParameterGroupName": "default.memorydb-redis6",
            "SubnetGroupName": "my-sg",
            "VpcId": "vpc-862xxxxc",
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "04:30-05:30",
            "NumShards": 2
        }
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的刪除快照。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteSnapshot。

delete-subnet-group

以下程式碼範例顯示如何使用 delete-subnet-group。

AWS CLI

刪除子網路群組

下列delete-subnet-group範例會刪除子網路。

```
aws memorydb delete-subnet-group \
    --subnet-group-name mysubnetgroup
```

輸出：

```
{
    "SubnetGroup": {
        "Name": "mysubnetgroup",
        "Description": "my subnet group",
        "VpcId": "vpc-86xxxx4fc",
        "Subnets": [
            {
                "Identifier": "subnet-56xxx61b",
                "AvailabilityZone": {
                    "Name": "us-east-1a"
                }
            }
        ],
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:subnetgroup/mysubnetgroup"
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的刪除子網路群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSubnetGroup。

delete-user

以下程式碼範例顯示如何使用 delete-user。

AWS CLI

若要刪除使用者

下列delete-user範例會刪除使用者。

```
aws memorydb delete-user \
```

```
    --user-name my-user
```

輸出：

```
{
    "User": {
        "Name": "my-user",
        "Status": "deleting",
        "AccessString": "on ~app::* resetchannels -@all +@read",
        "ACLNames": [
            "my-acl"
        ],
        "MinimumEngineVersion": "6.2",
        "Authentication": {
            "Type": "password",
            "PasswordCount": 1
        },
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:user/my-user"
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的使用存取控制清單驗證使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteUser。

describe-acls

以下程式碼範例顯示如何使用 describe-acls。

AWS CLI

傳回 ACLs清單

下列 describe-acls` 會傳回 ACLs清單。

```
aws memorydb describe-acls
```

輸出：

```
{
    "ACLs": [
        {
```

```
            "Name": "open-access",
            "Status": "active",
            "UserNames": [
                "default"
            ],
            "MinimumEngineVersion": "6.2",
            "Clusters": [],
            "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:acl/open-access"
        },
        {
            "Name": my-acl",
            "Status": "active",
            "UserNames": [],
            "MinimumEngineVersion": "6.2",
            "Clusters": [
                "my-cluster"
            ],
            "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxxx:acl/my-acl"
        }
    ]
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的使用存取控制清單驗證使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAcls。

describe-clusters

以下程式碼範例顯示如何使用 describe-clusters。

AWS CLI

傳回叢集清單

下列 describe-clusters` 會傳回叢集清單。

```
aws memorydb describe-clusters
```

輸出：

```
{
    "Clusters": [
        {
```

```
                "Name": "my-cluster",
                "Status": "available",
                "NumberOfShards": 2,
                "ClusterEndpoint": {
                    "Address": "clustercfg.my-cluster.llru6f.memorydb.us-
east-1.amazonaws.com",
                    "Port": 6379
                },
                "NodeType": "db.r6g.large",
                "EngineVersion": "6.2",
                "EnginePatchVersion": "6.2.6",
                "ParameterGroupName": "default.memorydb-redis6",
                "ParameterGroupStatus": "in-sync",
                "SecurityGroups": [
                    {
                        "SecurityGroupId": "sg-0a1434xxxxxc9fae",
                        "Status": "active"
                    }
                ],
                "SubnetGroupName": "pat-sg",
                "TLSEnabled": true,
                "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:cluster/my-cluster",
                "SnapshotRetentionLimit": 0,
                "MaintenanceWindow": "wed:03:00-wed:04:00",
                "SnapshotWindow": "04:30-05:30",
                "ACLName": "my-acl",
                "AutoMinorVersionUpgrade": true
        }
    ]
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的管理叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClusters。

`describe-engine-versions`

以下程式碼範例顯示如何使用 `describe-engine-versions`。

AWS CLI

傳回引擎版本的清單

下列 describe-engine-versions` 會傳回引擎版本的清單。

```
aws memorydb describe-engine-versions
```

輸出：

```
{
    "EngineVersions": [
        {
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
            "ParameterGroupFamily": "memorydb_redis6"
        }
    ]
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的引擎版本和升級。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEngineVersions。

describe-events

以下程式碼範例顯示如何使用 describe-events。

AWS CLI

傳回事件清單

下列 describe-events` 會傳回事件清單。

```
aws memorydb describe-events
```

輸出：

```
{
    "Events": [
        {
            "SourceName": "my-cluster",
            "SourceType": "cluster",
            "Message": "Increase replica count started for replication group my-
cluster on 2022-07-22T14:09:01.440Z",
            "Date": "2022-07-22T07:09:01.443000-07:00"
        },
```

```
        {
            "SourceName": "my-user",
            "SourceType": "user",
            "Message": "Create user my-user operation completed.",
            "Date": "2022-07-22T07:00:02.975000-07:00"
        }
    ]
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的監控事件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEvents。

describe-parameter-groups

以下程式碼範例顯示如何使用 describe-parameter-groups。

AWS CLI

傳回參數群組的清單

下列 describe-parameter-groups` 會傳回參數群組的清單。

```
aws memorydb describe-parameter-groups
```

輸出：

```
{
    "ParameterGroups": [
        {
            "Name": "default.memorydb-redis6",
            "Family": "memorydb_redis6",
            "Description": "Default parameter group for memorydb_redis6",
            "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/
default.memorydb-redis6"
        }
    ]
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的使用參數群組設定引擎參數。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeParameterGroups。

describe-parameters

以下程式碼範例顯示如何使用 describe-parameters。

AWS CLI

傳回參數清單

下列 describe-parameters` 會傳回參數清單。

```
aws memorydb describe-parameters
```

輸出：

```
{
    "Parameters": [
        {
            "Name": "acllog-max-len",
            "Value": "128",
            "Description": "The maximum length of the ACL Log",
            "DataType": "integer",
            "AllowedValues": "1-10000",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "activedefrag",
            "Value": "no",
            "Description": "Enabled active memory defragmentation",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "active-defrag-cycle-max",
            "Value": "75",
            "Description": "Maximal effort for defrag in CPU percentage",
            "DataType": "integer",
            "AllowedValues": "1-75",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "active-defrag-cycle-min",
            "Value": "5",
            "Description": "Minimal effort for defrag in CPU percentage",
```

```
            "DataType": "integer",
            "AllowedValues": "1-75",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "active-defrag-ignore-bytes",
            "Value": "104857600",
            "Description": "Minimum amount of fragmentation waste to start active
defrag",
            "DataType": "integer",
            "AllowedValues": "1048576-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "active-defrag-max-scan-fields",
            "Value": "1000",
            "Description": "Maximum number of set/hash/zset/list fields that will be
processed from the main dictionary scan",
            "DataType": "integer",
            "AllowedValues": "1-1000000",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "active-defrag-threshold-lower",
            "Value": "10",
            "Description": "Minimum percentage of fragmentation to start active
defrag",
            "DataType": "integer",
            "AllowedValues": "1-100",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "active-defrag-threshold-upper",
            "Value": "100",
            "Description": "Maximum percentage of fragmentation at which we use
maximum effort",
            "DataType": "integer",
            "AllowedValues": "1-100",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "active-expire-effort",
            "Value": "1",
```

```
            "Description": "The amount of effort that redis uses to expire items in
the active expiration job",
            "DataType": "integer",
            "AllowedValues": "1-10",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "activerehashing",
            "Value": "yes",
            "Description": "Apply rehashing or not",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "client-output-buffer-limit-normal-hard-limit",
            "Value": "0",
            "Description": "Normal client output buffer hard limit in bytes",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "client-output-buffer-limit-normal-soft-limit",
            "Value": "0",
            "Description": "Normal client output buffer soft limit in bytes",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "client-output-buffer-limit-normal-soft-seconds",
            "Value": "0",
            "Description": "Normal client output buffer soft limit in seconds",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "client-output-buffer-limit-pubsub-hard-limit",
            "Value": "33554432",
            "Description": "Pubsub client output buffer hard limit in bytes",
            "DataType": "integer",
            "AllowedValues": "0-",
```

```
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "client-output-buffer-limit-pubsub-soft-limit",
            "Value": "8388608",
            "Description": "Pubsub client output buffer soft limit in bytes",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "client-output-buffer-limit-pubsub-soft-seconds",
            "Value": "60",
            "Description": "Pubsub client output buffer soft limit in seconds",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "hash-max-ziplist-entries",
            "Value": "512",
            "Description": "The maximum number of hash entries in order for the
dataset to be compressed",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "hash-max-ziplist-value",
            "Value": "64",
            "Description": "The threshold of biggest hash entries in order for the
dataset to be compressed",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "hll-sparse-max-bytes",
            "Value": "3000",
            "Description": "HyperLogLog sparse representation bytes limit",
            "DataType": "integer",
            "AllowedValues": "1-16000",
            "MinimumEngineVersion": "6.2.4"
        },
```

```
        {
            "Name": "lazyfree-lazy-eviction",
            "Value": "no",
            "Description": "Perform an asynchronous delete on evictions",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "lazyfree-lazy-expire",
            "Value": "no",
            "Description": "Perform an asynchronous delete on expired keys",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "lazyfree-lazy-server-del",
            "Value": "no",
            "Description": "Perform an asynchronous delete on key updates",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "lazyfree-lazy-user-del",
            "Value": "no",
            "Description": "Specifies whether the default behavior of DEL command
acts the same as UNLINK",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "lfu-decay-time",
            "Value": "1",
            "Description": "The amount of time in minutes to decrement the key
counter for LFU eviction policyd",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "lfu-log-factor",
```

```
            "Value": "10",
            "Description": "The log factor for incrementing key counter for LFU
 eviction policy",
            "DataType": "integer",
            "AllowedValues": "1-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "list-compress-depth",
            "Value": "0",
            "Description": "Number of quicklist ziplist nodes from each side of
 the list to exclude from compression. The head and tail of the list are always
 uncompressed for fast push/pop operations",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "maxmemory-policy",
            "Value": "noeviction",
            "Description": "Max memory policy",
            "DataType": "string",
            "AllowedValues": "volatile-lru,allkeys-lru,volatile-lfu,allkeys-
lfu,volatile-random,allkeys-random,volatile-ttl,noeviction",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "maxmemory-samples",
            "Value": "3",
            "Description": "Max memory samples",
            "DataType": "integer",
            "AllowedValues": "1-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "notify-keyspace-events",
            "Description": "The keyspace events for Redis to notify Pub/Sub clients
 about. By default all notifications are disabled",
            "DataType": "string",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "set-max-intset-entries",
            "Value": "512",
```

```
            "Description": "The limit in the size of the set in order for the
dataset to be compressed",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "slowlog-log-slower-than",
            "Value": "10000",
            "Description": "The execution time, in microseconds, to exceed in order
for the command to get logged. Note that a negative number disables the slow log,
while a value of zero forces the logging of every command",
            "DataType": "integer",
            "AllowedValues": "-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "slowlog-max-len",
            "Value": "128",
            "Description": "The length of the slow log. There is no limit to this
length. Just be aware that it will consume memory. You can reclaim memory used by
the slow log with SLOWLOG RESET.",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "stream-node-max-bytes",
            "Value": "4096",
            "Description": "The maximum size of a single node in a stream in bytes",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "stream-node-max-entries",
            "Value": "100",
            "Description": "The maximum number of items a single node in a stream
can contain",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
```

```
            "Name": "tcp-keepalive",
            "Value": "300",
            "Description": "If non-zero, send ACKs every given number of seconds",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "timeout",
            "Value": "0",
            "Description": "Close connection if client is idle for a given number of
seconds, or never if 0",
            "DataType": "integer",
            "AllowedValues": "0,20-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "tracking-table-max-keys",
            "Value": "1000000",
            "Description": "The maximum number of keys allowed for the tracking
table for client side caching",
            "DataType": "integer",
            "AllowedValues": "1-100000000",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "zset-max-ziplist-entries",
            "Value": "128",
            "Description": "The maximum number of sorted set entries in order for
the dataset to be compressed",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "zset-max-ziplist-value",
            "Value": "64",
            "Description": "The threshold of biggest sorted set entries in order for
the dataset to be compressed",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的使用參數群組設定引擎參數。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeParameter。

`describe-snapshots`

以下程式碼範例顯示如何使用 `describe-snapshots`。

AWS CLI

傳回快照清單

下列 describe-snapshots` 會傳回快照清單。

```
aws memorydb describe-snapshots
```

輸出：

```
{
"Snapshots": [
    {
        "Name": "my-cluster-snapshot",
        "Status": "available",
        "Source": "manual",
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx2:snapshot/my-cluster-
snapshot",
        "ClusterConfiguration": {
            "Name": "my-cluster",
            "Description": " ",
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "Port": 6379,
            "ParameterGroupName": "default.memorydb-redis6",
            "SubnetGroupName": "my-sg",
            "VpcId": "vpc-862574fc",
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "04:30-05:30",
            "NumShards": 2
        }
```

```
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的快照和還原。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeSnapshots。

`describe-subnet-groups`

以下程式碼範例顯示如何使用 `describe-subnet-groups`。

AWS CLI

傳回子網路群組清單

下列 describe-subnet-groups` 會傳回子網路群組的清單。

```
aws memorydb describe-subnet-groups
```

輸出

```
{
    "SubnetGroups": [
        {
            "Name": "my-sg",
            "Description": "pat-sg",
            "VpcId": "vpc-86xxx4fc",
            "Subnets": [
                {
                    "Identifier": "subnet-faxx84a6",
                    "AvailabilityZone": {
                        "Name": "us-east-1b"
                    }
                },
                {
                    "Identifier": "subnet-56xxf61b",
                    "AvailabilityZone": {
                        "Name": "us-east-1a"
                    }
                }
            ],
            "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:subnetgroup/my-sg"
```

```
        }
    ]
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的子網路和子網路群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSubnetGroups。

describe-users

以下程式碼範例顯示如何使用 describe-users。

AWS CLI

傳回使用者清單

下列 describe-users` 會傳回使用者清單。

```
aws memorydb describe-users
```

輸出

```
{
    "Users": [
        {
            "Name": "default",
            "Status": "active",
            "AccessString": "on ~* &* +@all",
            "ACLNames": [
                "open-access"
            ],
            "MinimumEngineVersion": "6.0",
            "Authentication": {
                "Type": "no-password"
            },
            "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:user/default"
        },
        {
            "Name": "my-user",
            "Status": "active",
            "AccessString": "off ~objects:* ~items:* ~public:* resetchannels -@all",
            "ACLNames": [],
```

```
            "MinimumEngineVersion": "6.2",
            "Authentication": {
                "Type": "password",
                "PasswordCount": 2
            },
            "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:user/my-user"
        }
    ]
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的使用存取控制清單驗證使用者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeUsers。

failover-shard

以下程式碼範例顯示如何使用 failover-shard。

AWS CLI

容錯移轉碎片

下列 failover-shard` 會在碎片上失敗。

```
aws memorydb failover-shard \
    --cluster-name my-cluster --shard-name 0001
```

輸出：

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "available",
        "NumberOfShards": 2,
        "ClusterEndpoint": {
            "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
```

```
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SecurityGroups": [
            {
                "SecurityGroupId": "sg-0a143xxxx45c9fae",
                "Status": "active"
            }
        ],
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
        "AutoMinorVersionUpgrade": true
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的使用MultiAZ同步備份將停機時間降至最低。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 FailoverShard。

`list-allowed-node-type-updates`

以下程式碼範例顯示如何使用 `list-allowed-node-type-updates`。

AWS CLI

傳回允許的節點類型更新清單

下列 list-allowed-node-type-updates 會傳回可用節點類型更新的清單。

```
aws memorydb list-allowed-node-type-updates
```

輸出：

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "available",
        "NumberOfShards": 2,
        "ClusterEndpoint": {
```

```
            "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SecurityGroups": [
            {
                "SecurityGroupId": "sg-0a143xxxx45c9fae",
                "Status": "active"
            }
        ],
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
        "AutoMinorVersionUpgrade": true
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的擴展。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAllowedNodeTypeUpdates。

list-tags

以下程式碼範例顯示如何使用 `list-tags`。

AWS CLI

傳回標籤清單

下列 list-tags 會傳回標籤清單。

```
aws memorydb list-tags \
    --resource-arn arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster
```

輸出：

```
{
    "TagList": [
        {
            "Key": "mytag",
            "Value": "myvalue"
        }
    ]
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的標記資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTags。

reset-parameter-group

以下程式碼範例顯示如何使用 reset-parameter-group。

AWS CLI

重設參數群組

下列 reset-parameter-group` 會重設參數群組。

```
aws memorydb reset-parameter-group \
    --parameter-group-name my-parameter-group \
    --all-parameters
```

輸出：

```
{
    "ParameterGroup": {
        "Name": "my-parameter-group",
        "Family": "memorydb_redis6",
        "Description": "my parameter group",
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/my-parameter-
group"
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的使用參數群組設定引擎參數。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetParameterGroup。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

標記資源

下列 tag-resource` 會將標籤新增至資源。

```
aws memorydb tag-resource \
    --resource-arn arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster \
    --tags Key="mykey",Value="myvalue"
```

輸出：

```
{
    "TagList": [
        {
            "Key": "mytag",
            "Value": "myvalue"
        },
        {
            "Key": "mykey",
            "Value": "myvalue"
        }
    ]
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的標記資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

更新 ACL

下列 update-acl` 會透過新增使用者來更新 ACL。

```
aws memorydb untag-resource \
    --resource-arn arn:aws:memorydb:us-east-1:491658xxxxx:cluster/my-cluster \
    --tag-keys mykey
```

輸出：

```
{
    "TagList": [
        {
            "Key": "mytag",
            "Value": "myvalue"
        }
    ]
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的標記資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-cluster

以下程式碼範例顯示如何使用 update-cluster。

AWS CLI

更新叢集

下列 update-cluster`` 會將叢集的參數群組更新為 my-parameter-group。

```
aws memorydb update-cluster \
    --cluster-name my-cluster \
    --parameter-group-name my-parameter-group
```

輸出：

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "available",
        "NumberOfShards": 2,
        "AvailabilityMode": "MultiAZ",
```

```
        "ClusterEndpoint": {
            "Address": "clustercfg.my-cluster.llru6f.memorydb.us-
east-1.amazonaws.com",
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "my-parameter-group",
        "ParameterGroupStatus": "in-sync",
        "SecurityGroups": [
            {
                "SecurityGroupId": "sg-0a143xxxxxc9fae",
                "Status": "active"
            }
        ],
        "SubnetGroupName": "pat-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
        "ACLName": "my-acl",
        "AutoMinorVersionUpgrade": true
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的修改叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateCluster。

update-parameter-group

以下程式碼範例顯示如何使用 update-parameter-group。

AWS CLI

更新參數群組

下列 update-parameter-group`` 會更新參數群組。

```
aws memorydb update-parameter-group \
    --parameter-group-name my-parameter-group \
```

```
    --parameter-name-values "ParameterName=activedefrag, ParameterValue=no"
```

輸出：

```
{
    "ParameterGroup": {
        "Name": "my-parameter-group",
        "Family": "memorydb_redis6",
        "Description": "my parameter group",
        "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:parametergroup/my-parameter-
group"
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的修改參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateParameterGroup。

update-subnet-group

以下程式碼範例顯示如何使用 update-subnet-group。

AWS CLI

更新子網路群組

下列 update-subnet-group` 會更新子網路群組的子網路 ID。

```
aws memorydb update-subnet-group \
    --subnet-group-name my-sg \
    --subnet-ids subnet-01f29d458f3xxxxx
```

輸出：

```
{
    "SubnetGroup": {
        "Name": "my-sg-1",
        "Description": "my-sg",
        "VpcId": "vpc-09d2cfc01xxxxxxx",
        "Subnets": [
            {
                "Identifier": "subnet-01f29d458fxxxxxx",
```

```
                "AvailabilityZone": {
                    "Name": "us-east-1a"
                }
            }
        ],
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:subnetgroup/my-sg"
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的子網路和子網路群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSubnetGroup。

update-user

以下程式碼範例顯示如何使用 update-user。

AWS CLI

更新使用者

以下內容會update-user修改使用者的存取字串。

```
aws memorydb update-user \
    --user-name my-user \
    --access-string "off ~objects:* ~items:* ~public:* resetchannels -@all"
```

輸出：

```
{
    "User": {
        "Name": "my-user",
        "Status": "modifying",
        "AccessString": "off ~objects:* ~items:* ~public:* resetchannels -@all",
        "ACLNames": [
            "myt-acl"
        ],
        "MinimumEngineVersion": "6.2",
        "Authentication": {
            "Type": "password",
            "PasswordCount": 2
        },
```

```
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:user/my-user"
    }
}
```

如需詳細資訊，請參閱 MemoryDB 使用者指南中的使用存取控制清單驗證使用者。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateUser。

使用 的 Amazon MSK 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon MSK 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-cluster`

以下程式碼範例顯示如何使用 `create-cluster`。

AWS CLI

建立 Amazon MSK 叢集

下列`create-cluster`範例會建立名為 的 MSK 叢集，`MessagingCluster`其中包含三個代理程式節點。名為 的 JSON 檔案會`brokernodegroupinfo.json`指定您希望 Amazon MSK 分配代理程式節點的三個子網路。此範例不會指定監控層級，因此叢集會取得`DEFAULT`層級。

```
aws kafka create-cluster \
    --cluster-name "MessagingCluster" \
    --broker-node-group-info file://brokernodegroupinfo.json \
    --kafka-version "2.2.1" \
    --number-of-broker-nodes 3
```

brokernodegroupinfo.json 的內容：

```
{
    "InstanceType": "kafka.m5.xlarge",
    "BrokerAZDistribution": "DEFAULT",
    "ClientSubnets": [
        "subnet-0123456789111abcd",
        "subnet-0123456789222abcd",
        "subnet-0123456789333abcd"
    ]
}
```

輸出：

```
{
    "ClusterArn": "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
    "ClusterName": "MessagingCluster",
    "State": "CREATING"
}
```

如需詳細資訊，請參閱《Amazon Managed Streaming for Apache Kafka》中的建立 Amazon MSK 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCluster。

create-configuration

以下程式碼範例顯示如何使用 create-configuration。

AWS CLI

建立自訂 Amazon MSK 組態

下列create-configuration範例會使用輸入檔案中指定的伺服器屬性來建立自訂 MSK 組態。

```
aws kafka create-configuration \
    --name "CustomConfiguration" \
    --description "Topic autocreation enabled; Apache ZooKeeper timeout 2000 ms; Log
 rolling 604800000 ms." \
    --kafka-versions "2.2.1" \
```

```
    --server-properties file://configuration.txt
```

configuration.txt 的內容：

```
auto.create.topics.enable = true
zookeeper.connection.timeout.ms = 2000
log.roll.ms = 604800000
```

此命令不會產生輸出。輸出：

```
{
    "Arn": "arn:aws:kafka:us-west-2:123456789012:configuration/CustomConfiguration/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
    "CreationTime": "2019-10-09T15:26:05.548Z",
    "LatestRevision":
        {
            "CreationTime": "2019-10-09T15:26:05.548Z",
            "Description": "Topic autocreation enabled; Apache ZooKeeper timeout
 2000 ms; Log rolling 604800000 ms.",
            "Revision": 1
        },
    "Name": "CustomConfiguration"
}
```

如需詳細資訊，請參閱《Amazon Managed Streaming for Apache Kafka 開發人員指南》中的 Amazon MSK 組態操作。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateConfiguration。

describe-cluster

以下程式碼範例顯示如何使用 describe-cluster。

AWS CLI

描述叢集

下列describe-cluster範例說明 Amazon MSK 叢集。

```
aws kafka describe-cluster \
    --cluster-arn arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5
```

輸出：

```
{
    "ClusterInfo": {
        "BrokerNodeGroupInfo": {
            "BrokerAZDistribution": "DEFAULT",
            "ClientSubnets": [
                "subnet-cbfff283",
                "subnet-6746046b"
            ],
            "InstanceType": "kafka.m5.large",
            "SecurityGroups": [
                "sg-f839b688"
            ],
            "StorageInfo": {
                "EbsStorageInfo": {
                    "VolumeSize": 100
                }
            }
        },
        "ClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5",
        "ClusterName": "demo-cluster-1",
        "CreationTime": "2020-07-09T02:31:36.223000+00:00",
        "CurrentBrokerSoftwareInfo": {
            "KafkaVersion": "2.2.1"
        },
        "CurrentVersion": "K3AEGXETSR30VB",
        "EncryptionInfo": {
            "EncryptionAtRest": {
                "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/
a7ca56d5-0768-4b64-a670-339a9fbef81c"
            },
            "EncryptionInTransit": {
                "ClientBroker": "TLS_PLAINTEXT",
                "InCluster": true
            }
        },
        "EnhancedMonitoring": "DEFAULT",
        "OpenMonitoring": {
            "Prometheus": {
                "JmxExporter": {
                    "EnabledInBroker": false
                },
```

```
                "NodeExporter": {
                    "EnabledInBroker": false
                }
            }
        },
        "NumberOfBrokerNodes": 2,
        "State": "ACTIVE",
        "Tags": {},
        "ZookeeperConnectString": "z-2.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-1.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-3.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181"
    }
}
```

如需詳細資訊，請參閱《Amazon Managed Streaming for Apache Kafka 開發人員指南》中的列出 Amazon MSK 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCluster。

get-bootstrap-brokers

以下程式碼範例顯示如何使用 get-bootstrap-brokers。

AWS CLI

取得引導代理程式

下列get-bootstrap-brokers範例會擷取 Amazon MSK 叢集的引導代理程式資訊。

```
aws kafka get-bootstrap-brokers \
    --cluster-arn arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5
```

輸出：

```
{
    "BootstrapBrokerString": "b-1.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:9092,b-2.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:9092",
    "BootstrapBrokerStringTls": "b-1.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:9094,b-2.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:9094"
```

```
}
```

如需詳細資訊，請參閱《Amazon Managed Streaming for Apache Kafka 開發人員指南》中的取得引導中介裝置。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBootstrapBrokers。

list-clusters

以下程式碼範例顯示如何使用 `list-clusters`。

AWS CLI

列出可用的叢集

下列`list-clusters`範例列出您 AWS 帳戶中的 Amazon MSK 叢集。

```
aws kafka list-clusters
```

輸出：

```
{
    "ClusterInfoList": [
        {
            "BrokerNodeGroupInfo": {
                "BrokerAZDistribution": "DEFAULT",
                "ClientSubnets": [
                    "subnet-cbfff283",
                    "subnet-6746046b"
                ],
                "InstanceType": "kafka.m5.large",
                "SecurityGroups": [
                    "sg-f839b688"
                ],
                "StorageInfo": {
                    "EbsStorageInfo": {
                        "VolumeSize": 100
                    }
                }
            },
            "ClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5",
```

```
            "ClusterName": "demo-cluster-1",
            "CreationTime": "2020-07-09T02:31:36.223000+00:00",
            "CurrentBrokerSoftwareInfo": {
                "KafkaVersion": "2.2.1"
            },
            "CurrentVersion": "K3AEGXETSR30VB",
            "EncryptionInfo": {
                "EncryptionAtRest": {
                    "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/
a7ca56d5-0768-4b64-a670-339a9fbef81c"
                },
                "EncryptionInTransit": {
                    "ClientBroker": "TLS_PLAINTEXT",
                    "InCluster": true
                }
            },
            "EnhancedMonitoring": "DEFAULT",
            "OpenMonitoring": {
                "Prometheus": {
                    "JmxExporter": {
                        "EnabledInBroker": false
                    },
                    "NodeExporter": {
                        "EnabledInBroker": false
                    }
                }
            },
            "NumberOfBrokerNodes": 2,
            "State": "ACTIVE",
            "Tags": {},
            "ZookeeperConnectString": "z-2.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-1.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-3.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Managed Streaming for Apache Kafka 開發人員指南》中的列出 Amazon MSK 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListClusters。

update-broker-storage

以下程式碼範例顯示如何使用 update-broker-storage。

AWS CLI

更新代理程式的 EBS 儲存體

下列update-broker-storage範例會更新叢集中所有代理程式的 EBS 儲存量。Amazon MSK 會將每個代理程式的目標儲存量設定為範例中指定的數量。您可以透過描述叢集或列出所有叢集來取得叢集的目前版本。

```
aws kafka update-broker-storage \
    --cluster-arn "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2" \
    --current-version "K21V3IB1VIZYYH" \
    --target-broker-ebs-volume-info "KafkaBrokerNodeId=ALL,VolumeSizeGB=1100"
```

輸出會傳回update-broker-storage此操作的 ARN。若要判斷此操作是否已完成，請使用 describe-cluster-operation命令搭配此 ARN 做為輸入。

```
{
    "ClusterArn": "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
    "ClusterOperationArn": "arn:aws:kafka:us-west-2:123456789012:cluster-
operation/V123450123/a1b2c3d4-1234-abcd-cdef-22222EXAMPLE-2/a1b2c3d4-abcd-1234-
bcde-33333EXAMPLE"
}
```

如需詳細資訊，請參閱《Amazon Managed Streaming for Apache Kafka 開發人員指南》中的更新中介裝置的 EBS 儲存體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateBrokerStorage。

update-cluster-configuration

以下程式碼範例顯示如何使用 update-cluster-configuration。

AWS CLI

更新 Amazon MSK 叢集的組態

下列update-cluster-configuration範例會更新指定現有 MSK 叢集的組態。它使用自訂 MSK 組態。

```
aws kafka update-cluster-configuration \
    --cluster-arn "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2" \
    --configuration-info file://configuration-info.json \
    --current-version "K21V3IB1VIZYYH"
```

configuration-info.json 的內容：

```
{
    "Arn": "arn:aws:kafka:us-west-2:123456789012:configuration/CustomConfiguration/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
    "Revision": 1
}
```

輸出會傳回update-cluster-configuration此操作的 ARN。若要判斷此操作是否已完成，請使用 describe-cluster-operation命令搭配此 ARN 做為輸入。

```
{
    "ClusterArn": "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
    "ClusterOperationArn": "arn:aws:kafka:us-west-2:123456789012:cluster-operation/V123450123/a1b2c3d4-1234-abcd-cdef-22222EXAMPLE-2/a1b2c3d4-abcd-1234-bcde-33333EXAMPLE"
}
```

如需詳細資訊，請參閱《Amazon Managed Streaming for Apache Kafka 開發人員指南》中的更新 Amazon MSK 叢集的組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateClusterConfiguration。

使用 的網路流量監控範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Network Flow Monitor 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-monitor`

以下程式碼範例顯示如何使用 `create-monitor`。

AWS CLI

建立監視器

下列`create-monitor`範例`demo`會在指定的帳戶中建立名為 的監視器。

```
aws networkflowmonitor create-monitor \
    --monitor-name demo \
    --local-resources type="AWS::EC2::VPC",identifier="arn:aws:ec2:us-
east-1:123456789012:vpc/vpc-03ea55eeda25adbb0"  \
    --scope-arn arn:aws:networkflowmonitor:us-east-1:123456789012:scope/
e21cda79-30a0-4c12-9299-d8629d76d8cf
```

輸出：

```
{
    "monitorArn": "arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/demo",
    "monitorName": "demo",
    "monitorStatus": "ACTIVE",
    "tags": {}
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的在網路流量監視器中建立監視器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateMonitor。

`create-scope`

以下程式碼範例顯示如何使用 `create-scope`。

AWS CLI

建立範圍

下列create-scope範例會建立範圍，其中包含一組資源，網路流量監控會針對這些資源產生網路流量指標。

```
aws networkflowmonitor create-scope \
    --targets '[{"targetIdentifier":{"targetId":
{"accountId":"123456789012"},"targetType":"ACCOUNT"},"region":"us-east-1"}]'
```

輸出：

```
{
    "scopeId": "97626f8d-8a21-4b5d-813a-1a0962dd4615",
    "status": "IN_PROGRESS",
    "tags": {}
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的網路流量監視器的元件和功能。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateScope。

delete-monitor

以下程式碼範例顯示如何使用 delete-monitor。

AWS CLI

刪除監視器

下列delete-monitor範例會刪除指定帳戶中名為 demo的監視器。

```
aws networkflowmonitor delete-monitor \
    --monitor-name demo
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的在網路流量監視器中刪除監視器。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteMonitor。

delete-scope

以下程式碼範例顯示如何使用 delete-scope。

AWS CLI

刪除範圍

下列delete-scope範例會刪除指定的範圍。

```
aws networkflowmonitor delete-scope \
    --scope-id fdc20616-6bb4-4242-a24e-a748e65ca7ac
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的網路流量監視器的元件和功能。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteScope。

get-monitor

以下程式碼範例顯示如何使用 get-monitor。

AWS CLI

擷取監視器的相關資訊

下列get-monitor範例顯示demo指定帳戶中名為 之監視器的相關資訊。

```
aws networkflowmonitor get-monitor \
    --monitor-name Demo
```

輸出：

```
{
    "monitorArn": "arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/Demo",
    "monitorName": "Demo",
    "monitorStatus": "ACTIVE",
    "localResources": [
```

```
        {
            "type": "AWS::EC2::VPC",
            "identifier": "arn:aws:ec2:us-east-1:123456789012:vpc/
vpc-03ea55eeda25adbb0"
        }
    ],
    "remoteResources": [],
    "createdAt": "2024-12-09T12:21:51.616000-06:00",
    "modifiedAt": "2024-12-09T12:21:55.412000-06:00",
    "tags": {}
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的網路流量監視器的元件和功能。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMonitor。

get-query-results-workload-insights-top-contributors-data

以下程式碼範例顯示如何使用 get-query-results-workload-insights-top-contributors-data。

AWS CLI

擷取工作負載洞見的首要貢獻者資料

下列get-query-results-workload-insights-top-contributors-data範例會傳回指定查詢的資料。

```
aws networkflowmonitor get-query-results-workload-insights-top-contributors-data \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id cc4f4ab3-3103-33b8-80ff-d6597a0c6cea
```

輸出：

```
{
    "datapoints": [
        {
            "timestamps": [
                "2024-12-09T19:00:00+00:00",
                "2024-12-09T19:05:00+00:00",
                "2024-12-09T19:10:00+00:00"
            ],
```

```
            "values": [
                259943.0,
                194856.0,
                216432.0
            ],
            "label": "use1-az6"
        }
    ],
    "unit": "Bytes"
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用工作負載洞察評估網路流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetQueryResultsWorkloadInsightsTopContributorsData。

get-query-results-workload-insights-top-contributors

以下程式碼範例顯示如何使用 get-query-results-workload-insights-top-contributors。

AWS CLI

擷取工作負載洞見的主要貢獻者

下列get-query-results-workload-insights-top-contributors範例會傳回指定查詢的資料。

```
aws networkflowmonitor get-query-results-workload-insights-top-contributors \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id 1fc423d3-b144-37a6-80e6-e2c7d26eea0c
```

輸出：

```
{
    "topContributors": [
        {
            "accountId": "123456789012",
            "localSubnetId": "subnet-0a5b30fb95dca2c14",
            "localAz": "use1-az6",
```

```
            "localVpcId": "vpc-03ea55eeda25adbb0",
            "localRegion": "us-east-1",
            "remoteIdentifier": "",
            "value": 908443,
            "localSubnetArn": "arn:aws:ec2:us-east-1:123456789012:subnet/
subnet-0a5b30fb95dca2c14",
            "localVpcArn": "arn:aws:ec2:us-east-1:123456789012:vpc/
vpc-03ea55eeda25adbb0"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用工作負載洞察評估網路流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetQueryResultsWorkloadInsightsTopContributors。

get-query-status-monitor-top-contributors

以下程式碼範例顯示如何使用 get-query-status-monitor-top-contributors。

AWS CLI

擷取查詢的狀態

下列get-query-status-monitor-top-contributors範例顯示指定帳戶中查詢的目前狀態。

```
aws networkflowmonitor get-query-status-monitor-top-contributors \
    --monitor-name Demo \
    --query-id 5398eabd-bc40-3f5f-aba3-bcb639d3c7ca
```

輸出：

```
{
    "status": "SUCCEEDED"
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用工作負載洞察評估網路流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetQueryStatusMonitorTopContributors。

get-query-status-workload-insights-top-contributors-data

以下程式碼範例顯示如何使用 get-query-status-workload-insights-top-contributors-data。

AWS CLI

擷取查詢的狀態

下列get-query-status-workload-insights-top-contributors-data範例顯示指定帳戶中查詢的目前狀態。

```
aws networkflowmonitor get-query-status-workload-insights-top-contributors-data \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id 4333754d-8ae1-3f29-b6b7-c36db2e7f8ac
```

輸出：

```
{
    "status": "SUCCEEDED"
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用工作負載洞察評估網路流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetQueryStatusWorkloadInsightsTopContributorsData。

get-query-status-workload-insights-top-contributors

以下程式碼範例顯示如何使用 get-query-status-workload-insights-top-contributors。

AWS CLI

擷取查詢的狀態

下列get-query-status-workload-insights-top-contributors範例顯示指定帳戶中查詢的目前狀態。

```
aws networkflowmonitor get-query-status-workload-insights-top-contributors \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id f2a87c70-3e5a-362e-8beb-4747d13d8419
```

輸出：

```
{
    "status": "SUCCEEDED"
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用工作負載洞察評估網路流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetQueryStatusWorkloadInsightsTopContributors。

get-scope

以下程式碼範例顯示如何使用 get-scope。

AWS CLI

擷取範圍的相關資訊

下列get-scope範例顯示範圍的相關資訊，例如狀態、標籤、名稱和目標詳細資訊。

```
aws networkflowmonitor get-scope \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf
```

輸出：

```
{
    "scopeId": "e21cda79-30a0-4c12-9299-d8629d76d8cf",
    "status": "SUCCEEDED",
    "scopeArn": "arn:aws:networkflowmonitor:us-east-1:123456789012:scope/
e21cda79-30a0-4c12-9299-d8629d76d8cf",
    "targets": [
        {
            "targetIdentifier": {
                "targetId": {
```

```
                    "accountId": "123456789012"
                },
                "targetType": "ACCOUNT"
            },
            "region": "us-east-1"
        }
    ],
    "tags": {}
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的網路流量監視器的元件和功能。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetScope。

`list-monitors`

以下程式碼範例顯示如何使用 `list-monitors`。

AWS CLI

擷取監視器清單

下列`list-monitors`範例會傳回指定帳戶中的所有監視器。

```
aws networkflowmonitor list-monitors
```

輸出：

```
{
    "monitors": [
        {
            "monitorArn": "arn:aws:networkflowmonitor:us-
east-1:123456789012:monitor/Demo",
            "monitorName": "Demo",
            "monitorStatus": "ACTIVE"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的網路流量監視器的元件和功能。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMonitors。

list-scopes

以下程式碼範例顯示如何使用 list-scopes。

AWS CLI

擷取範圍清單

下列list-scopes範例列出指定帳戶中的所有範圍。

```
aws networkflowmonitor list-scopes
```

輸出：

```
{
    "scopes": [
        {
            "scopeId": "fdc20616-6bb4-4242-a24e-a748e65ca7ac",
            "status": "SUCCEEDED",
            "scopeArn": "arn:aws:networkflowmonitor:us-east-1:123456789012:scope/
fdc20616-6bb4-4242-a24e-a748e65ca7ac"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的網路流量監視器的元件和功能。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListScopes。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出標籤

下列list-tags-for-resource範例會傳回與指定資源相關聯的所有標籤。

```
aws networkflowmonitor list-tags-for-resource \
    --resource-arn arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/Demo
```

輸出：

```
{
    "tags": {
        "Value": "Production",
        "Key": "stack"
    }
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的標記 Amazon CloudWatch 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

start-query-monitor-top-contributors

以下程式碼範例顯示如何使用 start-query-monitor-top-contributors。

AWS CLI

啟動查詢

下列start-query-monitor-top-contributors範例會啟動查詢，其會傳回查詢 ID 以擷取主要參與者。

```
aws networkflowmonitor start-query-monitor-top-contributors \
    --monitor-name Demo \
    --start-time 2024-12-09T19:00:00Z \
    --end-time 2024-12-09T19:15:00Z \
    --metric-name DATA_TRANSFERRED \
    --destination-category UNCLASSIFIED
```

輸出：

```
{
    "queryId": "aecd3a88-0283-35b0-a17d-6e944dc8531d"
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用工作負載洞察評估網路流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartQueryMonitorTopContributors。

start-query-workload-insights-top-contributors-data

以下程式碼範例顯示如何使用 start-query-workload-insights-top-contributors-data。

AWS CLI

啟動查詢

下列start-query-workload-insights-top-contributors-data範例會啟動查詢，其會傳回查詢 ID 以擷取主要參與者。

```
aws networkflowmonitor start-query-workload-insights-top-contributors-data \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --start-time 2024-12-09T19:00:00Z \
    --end-time 2024-12-09T19:15:00Z \
    --metric-name DATA_TRANSFERRED \
    --destination-category UNCLASSIFIED
```

輸出：

```
{
    "queryId": "cc4f4ab3-3103-33b8-80ff-d6597a0c6cea"
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用工作負載洞察評估網路流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartQueryWorkloadInsightsTopContributorsData。

start-query-workload-insights-top-contributors

以下程式碼範例顯示如何使用 start-query-workload-insights-top-contributors。

AWS CLI

啟動查詢

下列start-query-workload-insights-top-contributors範例會啟動查詢，其會傳回查詢 ID 以擷取主要參與者。

```
aws networkflowmonitor start-query-workload-insights-top-contributors \
```

```
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --start-time 2024-12-09T19:00:00Z \
    --end-time 2024-12-09T19:15:00Z \
    --metric-name DATA_TRANSFERRED \
    --destination-category UNCLASSIFIED
```

輸出：

```
{
    "queryId": "1fc423d3-b144-37a6-80e6-e2c7d26eea0c"
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用工作負載洞察評估網路流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartQueryWorkloadInsightsTopContributors。

stop-query-monitor-top-contributors

以下程式碼範例顯示如何使用 stop-query-monitor-top-contributors。

AWS CLI

停止查詢

下列stop-query-monitor-top-contributors範例會停止指定帳戶中的查詢。

```
aws networkflowmonitor stop-query-monitor-top-contributors \
    --monitor-name Demo \
    --query-id aecd3a88-0283-35b0-a17d-6e944dc8531d
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用工作負載洞察評估網路流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopQueryMonitorTopContributors。

stop-query-workload-insights-top-contributors-data

以下程式碼範例顯示如何使用 stop-query-workload-insights-top-contributors-data。

AWS CLI

停止查詢

下列stop-query-workload-insights-top-contributors-data範例會停止指定帳戶中的查詢。

```
aws networkflowmonitor stop-query-workload-insights-top-contributors-data \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id cc4f4ab3-3103-33b8-80ff-d6597a0c6cea
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用工作負載洞察評估網路流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopQueryWorkloadInsightsTopContributorsData。

stop-query-workload-insights-top-contributors

以下程式碼範例顯示如何使用 stop-query-workload-insights-top-contributors。

AWS CLI

停止查詢

下列stop-query-workload-insights-top-contributors範例會停止指定帳戶中的查詢。

```
aws networkflowmonitor stop-query-workload-insights-top-contributors \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id 1fc423d3-b144-37a6-80e6-e2c7d26eea0c
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用工作負載洞察評估網路流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopQueryWorkloadInsightsTopContributors。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤新增至指定的資源

下列tag-resource範例會將標籤新增至指定帳戶中的監視器。

```
aws networkflowmonitor tag-resource \
    --resource-arn arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/Demo \
    --tags Key=stack,Value=Production
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的標記 Amazon CloudWatch 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從指定的資源移除標籤

下列untag-resource範例會從指定帳戶中的監視器移除標籤。

```
aws networkflowmonitor untag-resource \
    --resource-arn arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/Demo \
    --tag-keys stack
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的標記 Amazon CloudWatch 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-monitor

以下程式碼範例顯示如何使用 update-monitor。

AWS CLI

更新現有的監視器

下列update-monitor範例會更新指定帳戶中名為 Demo的監視器。

```
aws networkflowmonitor update-monitor \
    --monitor-name Demo \
    --local-resources-to-add type="AWS::EC2::VPC",identifier="arn:aws:ec2:us-
east-1:123456789012:vpc/vpc-048d08dfbec623f94"
```

輸出：

```
{
    "monitorArn": "arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/Demo",
    "monitorName": "Demo",
    "monitorStatus": "ACTIVE",
    "tags": {
        "Value": "Production",
        "Key": "stack"
    }
}
```

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的網路流量監視器的元件和功能。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateMonitor。

使用 的 Network Manager 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Network Manager 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

associate-customer-gateway

以下程式碼範例顯示如何使用 associate-customer-gateway。

AWS CLI

建立客戶閘道的關聯

下列associate-customer-gateway範例會將指定全球網路cgw-11223344556677889中的客戶閘道與裝置 建立關聯device-07f6fd08867abc123。

```
aws networkmanager associate-customer-gateway \
    --customer-gateway-arn arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889  \
    --global-network-id global-network-01231231231231231 \
    --device-id device-07f6fd08867abc123 \
    --region us-west-2
```

輸出：

```
{
    "CustomerGatewayAssociation": {
        "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889",
        "GlobalNetworkId": "global-network-01231231231231231",
        "DeviceId": "device-07f6fd08867abc123",
        "State": "PENDING"
    }
}
```

如需詳細資訊，請參閱 Transit Gateway Network Manager 指南中的客戶閘道關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateCustomerGateway。

associate-link

以下程式碼範例顯示如何使用 associate-link。

AWS CLI

建立連結的關聯

下列associate-link範例會將連結link-11112222aaaabbbb1與裝置 建立關聯device-07f6fd08867abc123。連結和裝置位於指定的全球網路中。

```
aws networkmanager associate-link \
    --global-network-id global-network-01231231231231231 \
    --device-id device-07f6fd08867abc123 \
    --link-id link-11112222aaaabbbb1 \
    --region us-west-2
```

輸出：

```
{
    "LinkAssociation": {
        "GlobalNetworkId": "global-network-01231231231231231",
        "DeviceId": "device-07f6fd08867abc123",
        "LinkId": "link-11112222aaaabbbb1",
        "LinkAssociationState": "PENDING"
    }
}
```

如需詳細資訊，請參閱 Transit Gateway Network Manager 指南中的裝置和連結關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateLink。

create-core-network

以下程式碼範例顯示如何使用 create-core-network。

AWS CLI

建立核心網路

下列create-core-network範例會使用 AWS Cloud WAN 全域網路中的選用描述和標籤來建立核心網路。

```
aws networkmanager create-core-network \
    --global-network-id global-network-cdef-EXAMPLE22222 \
    --description "Main headquarters location" \
```

```
    --tags Key=Name,Value="New York City office"
```

輸出：

```
{
    "CoreNetwork": {
        "GlobalNetworkId": "global-network-cdef-EXAMPLE22222",
        "CoreNetworkId": "core-network-cdef-EXAMPLE33333",
        "CoreNetworkArn": "arn:aws:networkmanager::987654321012:core-network/core-
network-cdef-EXAMPLE33333",
        "Description": "Main headquarters location",
        "CreatedAt": "2022-01-10T19:53:59+00:00",
        "State": "AVAILABLE",
        "Tags": [
            {
                "Key": "Name",
                "Value": "New York City office"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《AWS Cloud WAN 使用者指南》中的全域和核心網路。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCoreNetwork。

create-device

以下程式碼範例顯示如何使用 create-device。

AWS CLI

建立裝置

下列create-device範例會在指定的全球網路中建立裝置。裝置詳細資訊包括描述、類型、廠商、模型和序號。

```
aws networkmanager create-device
    --global-network-id global-network-01231231231231231 \
    --description "New York office device" \
    --type "office device" \
    --vendor "anycompany" \
```

```
    --model "abcabc" \
    --serial-number "1234" \
    --region us-west-2
```

輸出：

```
{
    "Device": {
        "DeviceId": "device-07f6fd08867abc123",
        "DeviceArn": "arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123",
        "GlobalNetworkId": "global-network-01231231231231231",
        "Description": "New York office device",
        "Type": "office device",
        "Vendor": "anycompany",
        "Model": "abcabc",
        "SerialNumber": "1234",
        "CreatedAt": 1575554005.0,
        "State": "PENDING"
    }
}
```

如需詳細資訊，請參閱 Transit Gateway Network Manager 指南中的使用裝置。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDevice。

create-global-network

以下程式碼範例顯示如何使用 create-global-network。

AWS CLI

建立全球網路

下列create-global-network範例會建立新的全球網路。建立時的初始狀態為 PENDING。

```
aws networkmanager create-global-network
```

輸出：

```
{
    "GlobalNetwork": {
```

```
        "GlobalNetworkId": "global-network-00a77fc0f722dae74",
        "GlobalNetworkArn": "arn:aws:networkmanager::987654321012:global-network/
global-network-00a77fc0f722dae74",
        "CreatedAt": "2022-03-14T20:31:56+00:00",
        "State": "PENDING"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateGlobalNetwork。

create-link

以下程式碼範例顯示如何使用 create-link。

AWS CLI

建立連結

下列create-link範例會在指定的全球網路中建立連結。連結包含有關連結類型、頻寬和提供者的描述和詳細資訊。網站 ID 表示連結相關聯的網站。

```
aws networkmanager create-link \
    --global-network-id global-network-01231231231231231 \
    --description "VPN Link" \
    --type "broadband" \
    --bandwidth UploadSpeed=10,DownloadSpeed=20 \
    --provider "AnyCompany" \
    --site-id site-444555aaabbb11223 \
    --region us-west-2
```

輸出：

```
{
    "Link": {
        "LinkId": "link-11112222aaaabbbb1",
        "LinkArn": "arn:aws:networkmanager::123456789012:link/global-
network-01231231231231231/link-11112222aaaabbbb1",
        "GlobalNetworkId": "global-network-01231231231231231",
        "SiteId": "site-444555aaabbb11223",
        "Description": "VPN Link",
        "Type": "broadband",
        "Bandwidth": {
```

```
            "UploadSpeed": 10,
            "DownloadSpeed": 20
        },
        "Provider": "AnyCompany",
        "CreatedAt": 1575555811.0,
        "State": "PENDING"
    }
}
```

如需詳細資訊，請參閱 Transit Gateway Network Manager 指南中的使用連結。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLink。

create-site

以下程式碼範例顯示如何使用 create-site。

AWS CLI

建立網站

下列create-site範例會在指定的全球網路中建立網站。網站詳細資訊包含描述和位置資訊。

```
aws networkmanager create-site \
    --global-network-id global-network-01231231231231231 \
    --description  "New York head office" \
    --location Latitude=40.7128,Longitude=-74.0060 \
    --region us-west-2
```

輸出：

```
{
    "Site": {
        "SiteId": "site-444555aaabbb11223",
        "SiteArn": "arn:aws:networkmanager::123456789012:site/global-
network-01231231231231231/site-444555aaabbb11223",
        "GlobalNetworkId": "global-network-01231231231231231",
        "Description": "New York head office",
        "Location": {
            "Latitude": "40.7128",
            "Longitude": "-74.0060"
        },
```

```
        "CreatedAt": 1575554300.0,
        "State": "PENDING"
    }
}
```

如需詳細資訊，請參閱 Transit Gateway Network Manager 指南中的使用網站。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSite。

`create-vpc-attachment`

以下程式碼範例顯示如何使用 `create-vpc-attachment`。

AWS CLI

建立 VPC 連接

下列`create-vpc-attachment`範例會在核心網路中建立具有 IPv6 支援的 VPC 連接。

```
aws networkmanager create-vpc-attachment \
    --core-network-id core-network-0fab62fe438d94db6 \
    --vpc-arn arn:aws:ec2:us-east-1:987654321012:vpc/vpc-09f37f69e2786eeb8  \
    --subnet-arns arn:aws:ec2:us-east-1:987654321012:subnet/subnet-04ca4e010857e7bb7
 \
    --Ipv6Support=true
```

輸出：

```
{
    "VpcAttachment": {
        "Attachment": {
            "CoreNetworkId": "core-network-0fab62fe438d94db6",
            "AttachmentId": "attachment-05e1da6eba87a06e6",
            "OwnerAccountId": "987654321012",
            "AttachmentType": "VPC",
            "State": "CREATING",
            "EdgeLocation": "us-east-1",
            "ResourceArn": "arn:aws:ec2:us-east-1:987654321012:vpc/
vpc-09f37f69e2786eeb8",
            "Tags": [],
            "CreatedAt": "2022-03-10T20:59:14+00:00",
            "UpdatedAt": "2022-03-10T20:59:14+00:00"
```

```
        },
        "SubnetArns": [
            "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-04ca4e010857e7bb7"
        ],
        "Options": {
            "Ipv6Support": true
        }
    }
}
```

如需詳細資訊，請參閱《Cloud WAN 使用者指南》中的建立附件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVpcAttachment。

`delete-attachment`

以下程式碼範例顯示如何使用 `delete-attachment`。

AWS CLI

刪除附件

下列`delete-attachment`範例會刪除 Connect 連接。

```
aws networkmanager delete-attachment \
    --attachment-id attachment-01feddaeae26ab68c
```

輸出：

```
{
    "Attachment": {
        "CoreNetworkId": "core-network-0f4b0a9d5ee7761d1",
        "AttachmentId": "attachment-01feddaeae26ab68c",
        "OwnerAccountId": "987654321012",
        "AttachmentType": "CONNECT",
        "State": "DELETING",
        "EdgeLocation": "us-east-1",
        "ResourceArn": "arn:aws:networkmanager::987654321012:attachment/
attachment-02c3964448fedf5aa",
        "CreatedAt": "2022-03-15T19:18:41+00:00",
        "UpdatedAt": "2022-03-15T19:28:59+00:00"
    }
```

```
}
```

如需詳細資訊，請參閱《Cloud WAN 使用者指南》中的刪除附件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAttachment。

delete-bucket-analytics-configuration

以下程式碼範例顯示如何使用 delete-bucket-analytics-configuration。

AWS CLI

刪除儲存貯體的分析組態

下列delete-bucket-analytics-configuration範例會移除指定儲存貯體和 ID 的分析組態。

```
aws s3api delete-bucket-analytics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBucketAnalyticsConfiguration。

delete-bucket-metrics-configuration

以下程式碼範例顯示如何使用 delete-bucket-metrics-configuration。

AWS CLI

刪除儲存貯體的指標組態

下列delete-bucket-metrics-configuration範例會移除指定儲存貯體和 ID 的指標組態。

```
aws s3api delete-bucket-metrics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 123
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBucketMetricsConfiguration。

delete-core-network

以下程式碼範例顯示如何使用 delete-core-network。

AWS CLI

刪除核心網路

下列delete-core-network範例會從 Cloud WAN 全域網路刪除核心網路。

```
aws networkmanager delete-core-network \
    --core-network-id core-network-0fab62fe438d94db6
```

輸出：

```
{
    "CoreNetwork": {
        "GlobalNetworkId": "global-network-0d59060f16a73bc41",
        "CoreNetworkId": "core-network-0fab62fe438d94db6",
        "Description": "Main headquarters location",
        "CreatedAt": "2021-12-09T18:31:11+00:00",
        "State": "DELETING",
        "Segments": [
            {
                "Name": "dev",
                "EdgeLocations": [
                    "us-east-1"
                ],
                "SharedSegments": []
            }
        ],
        "Edges": [
            {
                "EdgeLocation": "us-east-1",
                "Asn": 64512,
                "InsideCidrBlocks": []
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Cloud WAN 使用者指南》中的核心網路。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCoreNetwork。

delete-device

以下程式碼範例顯示如何使用 delete-device。

AWS CLI

刪除裝置

下列delete-device範例會從指定的全域網路刪除指定的裝置。

```
aws networkmanager delete-device \
    --global-network-id global-network-01231231231231231 \
    --device-id device-07f6fd08867abc123 \
    --region us-west-2
```

輸出：

```
{
    "Device": {
        "DeviceId": "device-07f6fd08867abc123",
        "DeviceArn": "arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123",
        "GlobalNetworkId": "global-network-01231231231231231",
        "Description": "New York office device",
        "Type": "office device",
        "Vendor": "anycompany",
        "Model": "abcabc",
        "SerialNumber": "1234",
        "SiteId": "site-444555aaabbb11223",
        "CreatedAt": 1575554005.0,
        "State": "DELETING"
    }
}
```

如需詳細資訊，請參閱 Transit Gateway Network Manager 指南中的使用裝置。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDevice。

delete-global-network

以下程式碼範例顯示如何使用 delete-global-network。

AWS CLI

刪除全域網路

下列delete-global-network範例會刪除全域網路。

```
aws networkmanager delete-global-network \
    --global-network-id global-network-052bedddccb193b6b
```

輸出：

```
{
    "GlobalNetwork": {
        "GlobalNetworkId": "global-network-052bedddccb193b6b",
        "GlobalNetworkArn": "arn:aws:networkmanager::987654321012:global-network/
global-network-052bedddccb193b6b",
        "CreatedAt": "2021-12-09T18:19:12+00:00",
        "State": "DELETING"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteGlobalNetwork。

delete-link

以下程式碼範例顯示如何使用 delete-link。

AWS CLI

刪除連結

下列delete-link範例會從指定的全域網路刪除指定的連結。

```
aws networkmanager delete-link \
    --global-network-id global-network-01231231231231231  \
    --link-id link-11112222aaaabbbb1 \
    --region us-west-2
```

輸出：

```
{
```

```
    "Link": {
        "LinkId": "link-11112222aaaabbbb1",
        "LinkArn": "arn:aws:networkmanager::123456789012:link/global-
network-01231231231231231/link-11112222aaaabbbb1",
        "GlobalNetworkId": "global-network-01231231231231231",
        "SiteId": "site-444555aaabbb11223",
        "Description": "VPN Link",
        "Type": "broadband",
        "Bandwidth": {
            "UploadSpeed": 20,
            "DownloadSpeed": 20
        },
        "Provider": "AnyCompany",
        "CreatedAt": 1575555811.0,
        "State": "DELETING"
    }
}
```

如需詳細資訊，請參閱 Transit Gateway Network Manager 指南中的使用連結。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLink。

delete-public-access-block

以下程式碼範例顯示如何使用 delete-public-access-block。

AWS CLI

刪除儲存貯體的封鎖公有存取組態

下列delete-public-access-block範例會移除指定儲存貯體上的封鎖公開存取組態。

```
aws s3api delete-public-access-block \
    --bucket amzn-s3-demo-bucket
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePublicAccessBlock。

delete-site

以下程式碼範例顯示如何使用 delete-site。

AWS CLI

刪除網站

下列delete-site範例會刪除指定全球網路中指定的網站 (site-444555aaabbb11223)。

```
aws networkmanager delete-site \
    --global-network-id global-network-01231231231231231  \
    --site-id site-444555aaabbb11223 \
    --region us-west-2
```

輸出：

```
{
    "Site": {
        "SiteId": "site-444555aaabbb11223",
        "SiteArn": "arn:aws:networkmanager::123456789012:site/global-
network-01231231231231231/site-444555aaabbb11223",
        "GlobalNetworkId": "global-network-01231231231231231",
        "Description": "New York head office",
        "Location": {
            "Latitude": "40.7128",
            "Longitude": "-74.0060"
        },
        "CreatedAt": 1575554300.0,
        "State": "DELETING"
    }
}
```

如需詳細資訊，請參閱 Transit Gateway Network Manager 指南中的使用網站。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSite。

deregister-transit-gateway

以下程式碼範例顯示如何使用 deregister-transit-gateway。

AWS CLI

從全球網路取消註冊傳輸閘道

下列deregister-transit-gateway範例會從指定的全域網路取消註冊指定的傳輸閘道。

```
aws networkmanager deregister-transit-gateway \
    --global-network-id global-network-01231231231231231 \
    --transit-gateway-arn arn:aws:ec2:us-west-2:123456789012:transit-gateway/
tgw-123abc05e04123abc \
    --region us-west-2
```

輸出：

```
{
    "TransitGatewayRegistration": {
        "GlobalNetworkId": "global-network-01231231231231231",
        "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/
tgw-123abc05e04123abc",
        "State": {
            "Code": "DELETING"
        }
    }
}
```

如需詳細資訊，請參閱 Transit Gateway Network Manager 指南中的 Transit Gateway 註冊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterTransitGateway。

describe-global-networks

以下程式碼範例顯示如何使用 describe-global-networks。

AWS CLI

描述您的 全球網路

下列describe-global-networks範例說明您帳戶中的所有全域網路。

```
aws networkmanager describe-global-networks \
    --region us-west-2
```

輸出：

```
{
    "GlobalNetworks": [
        {
```

```
            "GlobalNetworkId": "global-network-01231231231231231",
            "GlobalNetworkArn": "arn:aws:networkmanager::123456789012:global-
network/global-network-01231231231231231",
            "Description": "Company 1 global network",
            "CreatedAt": 1575553525.0,
            "State": "AVAILABLE"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeGlobalNetworks。

disassociate-customer-gateway

以下程式碼範例顯示如何使用 disassociate-customer-gateway。

AWS CLI

取消與客戶閘道的關聯

下列disassociate-customer-gateway範例會取消指定客戶閘道 (cgw-11223344556677889) 與指定全域網路的關聯。

```
aws networkmanager disassociate-customer-gateway \
    --global-network-id global-network-01231231231231231 \
    --customer-gateway-arn arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889 \
    --region us-west-2
```

輸出：

```
{
    "CustomerGatewayAssociation": {
        "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889",
        "GlobalNetworkId": "global-network-01231231231231231",
        "DeviceId": "device-07f6fd08867abc123",
        "State": "DELETING"
    }
}
```

如需詳細資訊，請參閱 Transit Gateway Network Manager 指南中的客戶閘道關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateCustomerGateway。

disassociate-link

以下程式碼範例顯示如何使用 disassociate-link。

AWS CLI

取消連結的關聯

下列disassociate-link範例會取消指定連結與指定全球網路device-07f6fd08867abc123中裝置的關聯。

```
aws networkmanager disassociate-link \
    --global-network-id global-network-01231231231231231 \
    --device-id device-07f6fd08867abc123 \
    --link-id link-11112222aaaabbbb1 \
    --region us-west-2
```

輸出：

```
{
    "LinkAssociation": {
        "GlobalNetworkId": "global-network-01231231231231231",
        "DeviceId": "device-07f6fd08867abc123",
        "LinkId": "link-11112222aaaabbbb1",
        "LinkAssociationState": "DELETING"
    }
}
```

如需詳細資訊，請參閱 Transit Gateway Network Manager 指南中的裝置和連結關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateLink。

get-bucket-analytics-configuration

以下程式碼範例顯示如何使用 get-bucket-analytics-configuration。

AWS CLI

擷取具有特定 ID 之儲存貯體的分析組態

下列get-bucket-analytics-configuration範例顯示指定儲存貯體和 ID 的分析組態。

```
aws s3api get-bucket-analytics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1
```

輸出：

```
{
    "AnalyticsConfiguration": {
        "StorageClassAnalysis": {},
        "Id": "1"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketAnalyticsConfiguration。

get-bucket-metrics-configuration

以下程式碼範例顯示如何使用 get-bucket-metrics-configuration。

AWS CLI

擷取具有特定 ID 之儲存貯體的指標組態

下列get-bucket-metrics-configuration範例顯示指定儲存貯體和 ID 的指標組態。

```
aws s3api get-bucket-metrics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 123
```

輸出：

```
{
    "MetricsConfiguration": {
        "Filter": {
            "Prefix": "logs"
        },
        "Id": "123"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketMetricsConfiguration。

get-customer-gateway-associations

以下程式碼範例顯示如何使用 get-customer-gateway-associations。

AWS CLI

取得您的客戶閘道關聯

下列get-customer-gateway-associations範例會取得指定全球網路的客戶閘道關聯。

```
aws networkmanager get-customer-gateway-associations \
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

輸出：

```
{
    "CustomerGatewayAssociations": [
        {
            "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-
gateway/cgw-11223344556677889",
            "GlobalNetworkId": "global-network-01231231231231231",
            "DeviceId": "device-07f6fd08867abc123",
            "State": "AVAILABLE"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCustomerGatewayAssociations。

get-devices

以下程式碼範例顯示如何使用 get-devices。

AWS CLI

取得您的裝置

下列get-devices範例會取得指定全球網路中的裝置。

```
aws networkmanager get-devices \
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

輸出：

```
{
    "Devices": [
        {
            "DeviceId": "device-07f6fd08867abc123",
            "DeviceArn": "arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123",
            "GlobalNetworkId": "global-network-01231231231231231",
            "Description": "NY office device",
            "Type": "office device",
            "Vendor": "anycompany",
            "Model": "abcabc",
            "SerialNumber": "1234",
            "CreatedAt": 1575554005.0,
            "State": "AVAILABLE"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDevices。

get-link-associations

以下程式碼範例顯示如何使用 get-link-associations。

AWS CLI

取得您的連結關聯

下列get-link-associations範例會取得指定全球網路中的連結關聯。

```
aws networkmanager get-link-associations \
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

輸出：

```
{
    "LinkAssociations": [
        {
            "GlobalNetworkId": "global-network-01231231231231231",
            "DeviceId": "device-07f6fd08867abc123",
            "LinkId": "link-11112222aaaabbbb1",
            "LinkAssociationState": "AVAILABLE"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLinkAssociations。

get-links

以下程式碼範例顯示如何使用 get-links。

AWS CLI

取得您的連結

下列get-links範例會取得指定全球網路中的連結。

```
aws networkmanager get-links \
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

輸出：

```
{
    "Links": [
        {
            "LinkId": "link-11112222aaaabbbb1",
            "LinkArn": "arn:aws:networkmanager::123456789012:link/global-
network-01231231231231231/link-11112222aaaabbbb1",
            "GlobalNetworkId": "global-network-01231231231231231",
            "SiteId": "site-444555aaabbb11223",
            "Description": "VPN Link",
            "Type": "broadband",
            "Bandwidth": {
                "UploadSpeed": 10,
                "DownloadSpeed": 20
```

```
            },
            "Provider": "AnyCompany",
            "CreatedAt": 1575555811.0,
            "State": "AVAILABLE"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLinks。

get-object-retention

以下程式碼範例顯示如何使用 get-object-retention。

AWS CLI

擷取物件的物件保留組態

下列get-object-retention範例會擷取指定物件的物件保留組態。

```
aws s3api get-object-retention \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --key doc1.rtf
```

輸出：

```
{
    "Retention": {
        "Mode": "GOVERNANCE",
        "RetainUntilDate": "2025-01-01T00:00:00.000Z"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetObjectRetention。

get-public-access-block

以下程式碼範例顯示如何使用 get-public-access-block。

AWS CLI

設定或修改儲存貯體的封鎖公開存取組態

下列get-public-access-block範例顯示指定儲存貯體的封鎖公開存取組態。

```
aws s3api get-public-access-block --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "PublicAccessBlockConfiguration": {
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "BlockPublicAcls": true,
        "RestrictPublicBuckets": true
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPublicAccessBlock。

get-sites

以下程式碼範例顯示如何使用 get-sites。

AWS CLI

取得您的網站

下列get-sites範例會取得指定全球網路中的網站。

```
aws networkmanager get-sites \
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

輸出：

```
{
    "Sites": [
        {
            "SiteId": "site-444555aaabbb11223",
            "SiteArn": "arn:aws:networkmanager::123456789012:site/global-
network-01231231231231231/site-444555aaabbb11223",
            "GlobalNetworkId": "global-network-01231231231231231",
            "Description": "NY head office",
```

```
            "Location": {
                "Latitude": "40.7128",
                "Longitude": "-74.0060"
            },
            "CreatedAt": 1575554528.0,
            "State": "AVAILABLE"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSites。

get-transit-gateway-registrations

以下程式碼範例顯示如何使用 get-transit-gateway-registrations。

AWS CLI

取得您的傳輸閘道註冊

下列get-transit-gateway-registrations範例會取得已註冊至指定全球網路的傳輸閘道。

```
aws networkmanager get-transit-gateway-registrations \
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

輸出：

```
{
    "TransitGatewayRegistrations": [
        {
            "GlobalNetworkId": "global-network-01231231231231231",
            "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-
gateway/tgw-123abc05e04123abc",
            "State": {
                "Code": "AVAILABLE"
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTransitGatewayRegistrations。

get-vpc-attachment

以下程式碼範例顯示如何使用 get-vpc-attachment。

AWS CLI

取得 VPC 連接

下列get-vpc-attachment範例會傳回 VPC 連接的相關資訊。

```
aws networkmanager get-vpc-attachment \
    --attachment-id  attachment-03b7ea450134787da
```

輸出：

```
{
    "VpcAttachment": {
        "Attachment": {
            "CoreNetworkId": "core-network-0522de1b226a5d7b3",
            "AttachmentId": "attachment-03b7ea450134787da",
            "OwnerAccountId": "987654321012",
            "AttachmentType": "VPC",
            "State": "CREATING",
            "EdgeLocation": "us-east-1",
            "ResourceArn": "arn:aws:ec2:us-east-1:987654321012:vpc/vpc-a7c4bbda",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "DevVPC"
                }
            ],
            "CreatedAt": "2022-03-11T17:48:58+00:00",
            "UpdatedAt": "2022-03-11T17:48:58+00:00"
        },
        "SubnetArns": [
            "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-202cde6c",
            "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-e5022dba",
            "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-2387ae02",
            "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-cda9dffc"
        ],
        "Options": {
            "Ipv6Support": false
        }
```

```
    }
}
```

如需詳細資訊，請參閱《Cloud WAN 使用者指南》中的附件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetVpcAttachment。

list-bucket-analytics-configurations

以下程式碼範例顯示如何使用 list-bucket-analytics-configurations。

AWS CLI

擷取儲存貯體的分析組態清單

以下內容會list-bucket-analytics-configurations擷取指定儲存貯體的分析組態清單。

```
aws s3api list-bucket-analytics-configurations \
    --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "AnalyticsConfigurationList": [
        {
            "StorageClassAnalysis": {},
            "Id": "1"
        }
    ],
    "IsTruncated": false
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBucketAnalyticsConfigurations。

list-bucket-metrics-configurations

以下程式碼範例顯示如何使用 list-bucket-metrics-configurations。

AWS CLI

擷取儲存貯體的指標組態清單

下列list-bucket-metrics-configurations範例會擷取指定儲存貯體的指標組態清單。

```
aws s3api list-bucket-metrics-configurations \
    --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "IsTruncated": false,
    "MetricsConfigurationList": [
        {
            "Filter": {
                "Prefix": "logs"
            },
            "Id": "123"
        },
        {
            "Filter": {
                "Prefix": "tmp"
            },
            "Id": "234"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBucketMetricsConfigurations。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出資源的標籤

下列list-tags-for-resource範例列出指定裝置資源的標籤 (device-07f6fd08867abc123)。

```
aws networkmanager list-tags-for-resource \
    --resource-arn arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123 \
```

```
    --region us-west-2
```

輸出：

```
{
    "TagList": [
        {
            "Key": "Network",
            "Value": "Northeast"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

put-bucket-metrics-configuration

以下程式碼範例顯示如何使用 put-bucket-metrics-configuration。

AWS CLI

設定儲存貯體的指標組態

下列put-bucket-metrics-configuration範例會為指定的儲存貯體設定 ID 為 123 的指標組態。

```
aws s3api put-bucket-metrics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 123 \
    --metrics-configuration '{"Id": "123", "Filter": {"Prefix": "logs"}}'
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketMetricsConfiguration。

put-object-retention

以下程式碼範例顯示如何使用 put-object-retention。

AWS CLI

設定物件的物件保留組態

下列put-object-retention範例會設定指定物件的物件保留組態，直到 2025-01-01。

```
aws s3api put-object-retention \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --key doc1.rtf \
    --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutObjectRetention。

put-public-access-block

以下程式碼範例顯示如何使用 put-public-access-block。

AWS CLI

設定儲存貯體的封鎖公開存取組態

下列put-public-access-block範例會設定指定儲存貯體的限制性封鎖公開存取組態。

```
aws s3api put-public-access-block \
    --bucket amzn-s3-demo-bucket \
    --public-access-block-
configuration "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPub
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutPublicAccessBlock。

register-transit-gateway

以下程式碼範例顯示如何使用 register-transit-gateway。

AWS CLI

在全球網路中註冊傳輸閘道

下列register-transit-gateway範例會在指定的全球網路tgw-123abc05e04123abc中註冊傳輸閘道。

```
aws networkmanager register-transit-gateway \
```

```
    --global-network-id global-network-01231231231231231 \
    --transit-gateway-arn arn:aws:ec2:us-west-2:123456789012:transit-gateway/
tgw-123abc05e04123abc \
    --region us-west-2
```

輸出：

```
{
    "TransitGatewayRegistration": {
        "GlobalNetworkId": "global-network-01231231231231231",
        "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/
tgw-123abc05e04123abc",
        "State": {
            "Code": "PENDING"
        }
    }
}
```

如需詳細資訊，請參閱 Transit Gateway Network Manager 指南中的 Transit Gateway 註冊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterTransitGateway。

reject-attachment

以下程式碼範例顯示如何使用 reject-attachment。

AWS CLI

拒絕附件

下列reject-attachment範例會拒絕 VPC 連接請求。

```
aws networkmanager reject-attachment \
    --attachment-id  attachment-03b7ea450134787da
```

輸出：

```
{
    "Attachment": {
        "CoreNetworkId": "core-network-0522de1b226a5d7b3",
        "AttachmentId": "attachment-03b7ea450134787da",
        "OwnerAccountId": "987654321012",
```

```
        "AttachmentType": "VPC",
        "State": "AVAILABLE",
        "EdgeLocation": "us-east-1",
        "ResourceArn": "arn:aws:ec2:us-east-1:987654321012:vpc/vpc-a7c4bbda",
        "CreatedAt": "2022-03-11T17:48:58+00:00",
        "UpdatedAt": "2022-03-11T17:51:25+00:00"
    }
}
```

如需詳細資訊，請參閱《Cloud WAN 使用者指南》中的附件接受。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RejectAttachment。

start-route-analysis

以下程式碼範例顯示如何使用 start-route-analysis。

AWS CLI

開始路由分析

下列start-route-analysis範例會開始來源和目的地之間的分析，包括選用的 include-return-path。

```
aws networkmanager start-route-analysis \
    --global-network-id global-network-00aa0aaa0b0aaa000 \
    --source TransitGatewayAttachmentArn=arn:aws:ec2:us-east-1:503089527312:transit-gateway-attachment/tgw-attach-0d4a2d491bf68c093,IpAddress=10.0.0.0 \
    --destination TransitGatewayAttachmentArn=arn:aws:ec2:us-west-1:503089527312:transit-gateway-attachment/tgw-attach-002577f30bb181742,IpAddress=11.0.0.0 \
    --include-return-path
```

輸出：

```
{
    "RouteAnalysis": {
        "GlobalNetworkId": "global-network-00aa0aaa0b0aaa000
        "OwnerAccountId": "1111222233333",
        "RouteAnalysisId": "a1873de1-273c-470c-1a2bc2345678",
        "StartTimestamp": 1695760154.0,
        "Status": "RUNNING",
```

```
        "Source": {
            "TransitGatewayAttachmentArn": "arn:aws:ec2:us-
east-1:111122223333:transit-gateway-attachment/tgw-attach-1234567890abcdef0,
            "TransitGatewayArn": "arn:aws:ec2:us-east-1:111122223333:transit-
gateway/tgw-abcdef01234567890",
            "IpAddress": "10.0.0.0"
        },
        "Destination": {
            "TransitGatewayAttachmentArn": "arn:aws:ec2:us-
west-1:555555555555:transit-gateway-attachment/tgw-attach-021345abcdef6789",
            "TransitGatewayArn": "arn:aws:ec2:us-west-1:111122223333:transit-
gateway/tgw-09876543210fedcba0",
            "IpAddress": "11.0.0.0"
        },
        "IncludeReturnPath": true,
        "UseMiddleboxes": false
    }
}
```

如需詳細資訊，請參閱《 AWS 全球傳輸閘道網路使用者指南》中的 Route Analyzer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartRouteAnalysis。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤套用至資源

下列tag-resource範例會將標籤套用至Network=Northeast裝置 device-07f6fd08867abc123。

```
aws networkmanager tag-resource \
    --resource-arn arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123 \
    --tags Key=Network,Value=Northeast \
    --region us-west-2
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源移除標籤

下列untag-resource範例Network會從裝置 移除具有 金鑰的標籤device-07f6fd08867abc123。

```
aws networkmanager untag-resource \
    --resource-arn arn:aws:networkmanager::123456789012:device/global-network-01231231231231231/device-07f6fd08867abc123 ]
    --tag-keys Network \
    --region us-west-2
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-device

以下程式碼範例顯示如何使用 update-device。

AWS CLI

更新裝置

下列update-device範例device-07f6fd08867abc123會透過指定裝置的網站 ID 來更新裝置。

```
aws networkmanager update-device \
    --global-network-id global-network-01231231231231231 \
    --device-id device-07f6fd08867abc123 \
    --site-id site-444555aaabbb11223 \
    --region us-west-2
```

輸出：

```
{
    "Device": {
```

```
        "DeviceId": "device-07f6fd08867abc123",
        "DeviceArn": "arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123",
        "GlobalNetworkId": "global-network-01231231231231231",
        "Description": "NY office device",
        "Type": "Office device",
        "Vendor": "anycompany",
        "Model": "abcabc",
        "SerialNumber": "1234",
        "SiteId": "site-444555aaabbb11223",
        "CreatedAt": 1575554005.0,
        "State": "UPDATING"
    }
}
```

如需詳細資訊，請參閱 Transit Gateway Network Manager 指南中的使用裝置。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDevice。

update-global-network

以下程式碼範例顯示如何使用 update-global-network。

AWS CLI

更新全球網路

下列update-global-network範例會更新全域網路 的描述global-network-01231231231231231。

```
aws networkmanager update-global-network \
    --global-network-id global-network-01231231231231231 \
    --description "Head offices" \
    --region us-west-2
```

輸出：

```
{
    "GlobalNetwork": {
        "GlobalNetworkId": "global-network-01231231231231231",
        "GlobalNetworkArn": "arn:aws:networkmanager::123456789012:global-network/
global-network-01231231231231231",
```

```
        "Description": "Head offices",
        "CreatedAt": 1575553525.0,
        "State": "UPDATING"
    }
}
```

如需詳細資訊，請參閱 Transit Gateway Network Manager 指南中的全球網路。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateGlobalNetwork。

update-link

以下程式碼範例顯示如何使用 update-link。

AWS CLI

更新連結

下列update-link範例會更新連結 的頻寬資訊link-11112222aaaabbbb1。

```
aws networkmanager update-link \
    --global-network-id global-network-01231231231231231 \
    --link-id link-11112222aaaabbbb1 \
    --bandwidth  UploadSpeed=20,DownloadSpeed=20 \
    --region us-west-2
```

輸出：

```
{
    "Link": {
        "LinkId": "link-11112222aaaabbbb1",
        "LinkArn": "arn:aws:networkmanager::123456789012:link/global-
network-01231231231231231/link-11112222aaaabbbb1",
        "GlobalNetworkId": "global-network-01231231231231231",
        "SiteId": "site-444555aaabbb11223",
        "Description": "VPN Link",
        "Type": "broadband",
        "Bandwidth": {
            "UploadSpeed": 20,
            "DownloadSpeed": 20
        },
        "Provider": "AnyCompany",
```

```
        "CreatedAt": 1575555811.0,
        "State": "UPDATING"
    }
}
```

如需詳細資訊，請參閱 Transit Gateway Network Manager 指南中的使用連結。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateLink。

update-site

以下程式碼範例顯示如何使用 update-site。

AWS CLI

更新網站

下列update-site範例會更新指定全球網路site-444555aaabbb11223中網站的說明。

```
aws networkmanager update-site \
    --global-network-id global-network-01231231231231231 \
    --site-id site-444555aaabbb11223 \
    --description "New York Office site" \
    --region us-west-2
```

輸出：

```
{
    "Site": {
        "SiteId": "site-444555aaabbb11223",
        "SiteArn": "arn:aws:networkmanager::123456789012:site/global-
network-01231231231231231/site-444555aaabbb11223",
        "GlobalNetworkId": "global-network-01231231231231231",
        "Description": "New York Office site",
        "Location": {
            "Latitude": "40.7128",
            "Longitude": "-74.0060"
        },
        "CreatedAt": 1575554528.0,
        "State": "UPDATING"
    }
}
```

如需詳細資訊，請參閱 Transit Gateway Network Manager 指南中的使用網站。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSite。

使用 的 OpenSearch Service 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 OpenSearch Service 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-elasticsearch-domain`

以下程式碼範例顯示如何使用 `create-elasticsearch-domain`。

AWS CLI

建立 Amazon Elasticsearch Service 網域

下列`create-elasticsearch-domain`命令會在 VPC 中建立新的 Amazon Elasticsearch Service 網域，並將存取權限制為單一使用者。Amazon ES 會從指定的子網路和安全群組 ID 推斷 VPC IDs。

```
aws es create-elasticsearch-domain \
    --domain-name vpc-cli-example \
    --elasticsearch-version 6.2 \
    --elasticsearch-cluster-
config InstanceType=m4.large.elasticsearch,InstanceCount=1 \
    --ebs-options EBSEnabled=true,VolumeType=standard,VolumeSize=10 \
    --access-policies '{"Version": "2012-10-17", "Statement": [ { "Effect":
 "Allow", "Principal": {"AWS": "arn:aws:iam::123456789012:root" }, "Action":"es:*",
 "Resource": "arn:aws:es:us-west-1:123456789012:domain/vpc-cli-example/*" } ] }' \
    --vpc-options SubnetIds=subnet-1a2a3a4a,SecurityGroupIds=sg-2a3a4a5a
```

輸出：

```
{
    "DomainStatus": {
        "ElasticsearchClusterConfig": {
            "DedicatedMasterEnabled": false,
            "InstanceCount": 1,
            "ZoneAwarenessEnabled": false,
            "InstanceType": "m4.large.elasticsearch"
        },
        "DomainId": "123456789012/vpc-cli-example",
        "CognitoOptions": {
            "Enabled": false
        },
        "VPCOptions": {
            "SubnetIds": [
                "subnet-1a2a3a4a"
            ],
            "VPCId": "vpc-3a4a5a6a",
            "SecurityGroupIds": [
                "sg-2a3a4a5a"
            ],
            "AvailabilityZones": [
                "us-west-1c"
            ]
        },
        "Created": true,
        "Deleted": false,
        "EBSOptions": {
            "VolumeSize": 10,
            "VolumeType": "standard",
            "EBSEnabled": true
        },
        "Processing": true,
        "DomainName": "vpc-cli-example",
        "SnapshotOptions": {
            "AutomatedSnapshotStartHour": 0
        },
        "ElasticsearchVersion": "6.2",
        "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:root\"},\"Action\":
\"es:*\",\"Resource\":\"arn:aws:es:us-west-1:123456789012:domain/vpc-cli-example/*
\"}]}",
        "AdvancedOptions": {
```

```
            "rest.action.multi.allow_explicit_index": "true"
        },
        "EncryptionAtRestOptions": {
            "Enabled": false
        },
        "ARN": "arn:aws:es:us-west-1:123456789012:domain/vpc-cli-example"
    }
}
```

如需詳細資訊，請參閱《Amazon Elasticsearch Service 開發人員指南》中的建立和管理 Amazon Elasticsearch Service 網域。 Amazon Elasticsearch Service

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateElasticsearchDomain。

describe-elasticsearch-domain-config

以下程式碼範例顯示如何使用 describe-elasticsearch-domain-config。

AWS CLI

若要取得網域組態詳細資訊

下列describe-elasticsearch-domain-config範例提供指定網域的組態詳細資訊，以及每個個別網域元件的狀態資訊。

```
aws es describe-elasticsearch-domain-config \
    --domain-name cli-example
```

輸出：

```
{
    "DomainConfig": {
        "ElasticsearchVersion": {
            "Options": "7.4",
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589395827.325,
                "UpdateVersion": 8,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "ElasticsearchClusterConfig": {
```

```
            "Options": {
                "InstanceType": "c5.large.elasticsearch",
                "InstanceCount": 1,
                "DedicatedMasterEnabled": true,
                "ZoneAwarenessEnabled": false,
                "DedicatedMasterType": "c5.large.elasticsearch",
                "DedicatedMasterCount": 3,
                "WarmEnabled": true,
                "WarmType": "ultrawarm1.medium.elasticsearch",
                "WarmCount": 2
            },
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589395827.325,
                "UpdateVersion": 8,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "EBSOptions": {
            "Options": {
                "EBSEnabled": true,
                "VolumeType": "gp2",
                "VolumeSize": 10
            },
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589395827.325,
                "UpdateVersion": 8,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "AccessPolicies": {
            "Options": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Principal\":{\"AWS\":\"*\"},\"Action\":\"es:*\",\"Resource\":
\"arn:aws:es:us-east-1:123456789012:domain/cli-example/*\"}]}",
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589395827.325,
                "UpdateVersion": 8,
                "State": "Active",
                "PendingDeletion": false
            }
```

```
        },
        "SnapshotOptions": {
            "Options": {
                "AutomatedSnapshotStartHour": 0
            },
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589395827.325,
                "UpdateVersion": 8,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "VPCOptions": {
            "Options": {},
            "Status": {
                "CreationDate": 1591210426.162,
                "UpdateDate": 1591210426.162,
                "UpdateVersion": 18,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "CognitoOptions": {
            "Options": {
                "Enabled": false
            },
            "Status": {
                "CreationDate": 1591210426.163,
                "UpdateDate": 1591210426.163,
                "UpdateVersion": 18,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "EncryptionAtRestOptions": {
            "Options": {
                "Enabled": true,
                "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/1a2a3a4a-1a2a-1a2a-1a2a-1a2a3a4a5a6a"
            },
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589395827.325,
```

```
            "UpdateVersion": 8,
            "State": "Active",
            "PendingDeletion": false
        }
    },
    "NodeToNodeEncryptionOptions": {
        "Options": {
            "Enabled": true
        },
        "Status": {
            "CreationDate": 1589395034.946,
            "UpdateDate": 1589395827.325,
            "UpdateVersion": 8,
            "State": "Active",
            "PendingDeletion": false
        }
    },
    "AdvancedOptions": {
        "Options": {
            "rest.action.multi.allow_explicit_index": "true"
        },
        "Status": {
            "CreationDate": 1589395034.946,
            "UpdateDate": 1589395827.325,
            "UpdateVersion": 8,
            "State": "Active",
            "PendingDeletion": false
        }
    },
    "LogPublishingOptions": {
        "Options": {},
        "Status": {
            "CreationDate": 1591210426.164,
            "UpdateDate": 1591210426.164,
            "UpdateVersion": 18,
            "State": "Active",
            "PendingDeletion": false
        }
    },
    "DomainEndpointOptions": {
        "Options": {
            "EnforceHTTPS": true,
            "TLSSecurityPolicy": "Policy-Min-TLS-1-0-2019-07"
        },
```

```
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589395827.325,
                "UpdateVersion": 8,
                "State": "Active",
                "PendingDeletion": false
            }
        },
        "AdvancedSecurityOptions": {
            "Options": {
                "Enabled": true,
                "InternalUserDatabaseEnabled": true
            },
            "Status": {
                "CreationDate": 1589395034.946,
                "UpdateDate": 1589827485.577,
                "UpdateVersion": 14,
                "State": "Active",
                "PendingDeletion": false
            }
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Elasticsearch Service 開發人員指南》中的建立和管理 Amazon Elasticsearch Service 網域。 Amazon Elasticsearch Service

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeElasticsearchDomainConfig。

describe-elasticsearch-domain

以下程式碼範例顯示如何使用 describe-elasticsearch-domain。

AWS CLI

取得單一網域的詳細資訊

下列describe-elasticsearch-domain範例提供指定網域的組態詳細資訊。

```
aws es describe-elasticsearch-domain \
    --domain-name cli-example
```

輸出：

```
{
    "DomainStatus": {
        "DomainId": "123456789012/cli-example",
        "DomainName": "cli-example",
        "ARN": "arn:aws:es:us-east-1:123456789012:domain/cli-example",
        "Created": true,
        "Deleted": false,
        "Endpoint": "search-cli-example-1a2a3a4a5a6a7a8a9a0a.us-
east-1.es.amazonaws.com",
        "Processing": false,
        "UpgradeProcessing": false,
        "ElasticsearchVersion": "7.4",
        "ElasticsearchClusterConfig": {
            "InstanceType": "c5.large.elasticsearch",
            "InstanceCount": 1,
            "DedicatedMasterEnabled": true,
            "ZoneAwarenessEnabled": false,
            "DedicatedMasterType": "c5.large.elasticsearch",
            "DedicatedMasterCount": 3,
            "WarmEnabled": true,
            "WarmType": "ultrawarm1.medium.elasticsearch",
            "WarmCount": 2
        },
        "EBSOptions": {
            "EBSEnabled": true,
            "VolumeType": "gp2",
            "VolumeSize": 10
        },
        "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Principal\":{\"AWS\":\"*\"},\"Action\":\"es:*\",\"Resource\":
\"arn:aws:es:us-east-1:123456789012:domain/cli-example/*\"}]}",
        "SnapshotOptions": {
            "AutomatedSnapshotStartHour": 0
        },
        "CognitoOptions": {
            "Enabled": false
        },
        "EncryptionAtRestOptions": {
            "Enabled": true,
            "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/1a2a3a4a-1a2a-1a2a-1a2a-1a2a3a4a5a6a"
        },
        "NodeToNodeEncryptionOptions": {
```

```
            "Enabled": true
        },
        "AdvancedOptions": {
            "rest.action.multi.allow_explicit_index": "true"
        },
        "ServiceSoftwareOptions": {
            "CurrentVersion": "R20200522",
            "NewVersion": "",
            "UpdateAvailable": false,
            "Cancellable": false,
            "UpdateStatus": "COMPLETED",
            "Description": "There is no software update available for this domain.",
            "AutomatedUpdateDate": 0.0
        },
        "DomainEndpointOptions": {
            "EnforceHTTPS": true,
            "TLSSecurityPolicy": "Policy-Min-TLS-1-0-2019-07"
        },
        "AdvancedSecurityOptions": {
            "Enabled": true,
            "InternalUserDatabaseEnabled": true
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Elasticsearch Service 開發人員指南》中的建立和管理 Amazon Elasticsearch Service 網域。 Amazon Elasticsearch Service

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeElasticsearchDomain。

describe-elasticsearch-domains

以下程式碼範例顯示如何使用 describe-elasticsearch-domains。

AWS CLI

取得一或多個網域的詳細資訊

下列describe-elasticsearch-domains範例提供一或多個網域的組態詳細資訊。

```
aws es describe-elasticsearch-domains \
    --domain-names cli-example-1 cli-example-2
```

輸出：

```
{
    "DomainStatusList": [{
            "DomainId": "123456789012/cli-example-1",
            "DomainName": "cli-example-1",
            "ARN": "arn:aws:es:us-east-1:123456789012:domain/cli-example-1",
            "Created": true,
            "Deleted": false,
            "Endpoint": "search-cli-example-1-1a2a3a4a5a6a7a8a9a0a.us-
east-1.es.amazonaws.com",
            "Processing": false,
            "UpgradeProcessing": false,
            "ElasticsearchVersion": "7.4",
            "ElasticsearchClusterConfig": {
                "InstanceType": "c5.large.elasticsearch",
                "InstanceCount": 1,
                 "DedicatedMasterEnabled": true,
                "ZoneAwarenessEnabled": false,
                "DedicatedMasterType": "c5.large.elasticsearch",
                "DedicatedMasterCount": 3,
                "WarmEnabled": true,
                "WarmType": "ultrawarm1.medium.elasticsearch",
                "WarmCount": 2
            },
            "EBSOptions": {
                "EBSEnabled": true,
                "VolumeType": "gp2",
                "VolumeSize": 10
            },
            "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Principal\":{\"AWS\":\"*\"},\"Action\":\"es:*\",\"Resource\":
\"arn:aws:es:us-east-1:123456789012:domain/cli-example-1/*\"}]}",
            "SnapshotOptions": {
                "AutomatedSnapshotStartHour": 0
            },
            "CognitoOptions": {
                "Enabled": false
            },
            "EncryptionAtRestOptions": {
                "Enabled": true,
                "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/1a2a3a4a-1a2a-1a2a-1a2a-1a2a3a4a5a6a"
            },
```

```
            "NodeToNodeEncryptionOptions": {
                "Enabled": true
            },
            "AdvancedOptions": {
                "rest.action.multi.allow_explicit_index": "true"
            },
            "ServiceSoftwareOptions": {
                "CurrentVersion": "R20200522",
                "NewVersion": "",
                "UpdateAvailable": false,
                "Cancellable": false,
                "UpdateStatus": "COMPLETED",
                "Description": "There is no software update available for this
domain.",
                "AutomatedUpdateDate": 0.0
            },
            "DomainEndpointOptions": {
                "EnforceHTTPS": true,
                "TLSSecurityPolicy": "Policy-Min-TLS-1-0-2019-07"
            },
            "AdvancedSecurityOptions": {
                "Enabled": true,
                "InternalUserDatabaseEnabled": true
            }
        },
        {
            "DomainId": "123456789012/cli-example-2",
            "DomainName": "cli-example-2",
            "ARN": "arn:aws:es:us-east-1:123456789012:domain/cli-example-2",
            "Created": true,
            "Deleted": false,
            "Processing": true,
            "UpgradeProcessing": false,
            "ElasticsearchVersion": "7.4",
            "ElasticsearchClusterConfig": {
                "InstanceType": "r5.large.elasticsearch",
                "InstanceCount": 1,
                "DedicatedMasterEnabled": false,
                "ZoneAwarenessEnabled": false,
                "WarmEnabled": false
            },
            "EBSOptions": {
                "EBSEnabled": true,
                "VolumeType": "gp2",
```

```
                "VolumeSize": 10
            },
            "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Deny\",\"Principal\":{\"AWS\":\"*\"},\"Action\":\"es:*\",\"Resource\":
\"arn:aws:es:us-east-1:123456789012:domain/cli-example-2/*\"}]}",
            "SnapshotOptions": {
                "AutomatedSnapshotStartHour": 0
            },
            "CognitoOptions": {
                "Enabled": false
            },
            "EncryptionAtRestOptions": {
                "Enabled": false
            },
            "NodeToNodeEncryptionOptions": {
                "Enabled": false
            },
            "AdvancedOptions": {
                "rest.action.multi.allow_explicit_index": "true"
            },
            "ServiceSoftwareOptions": {
                "CurrentVersion": "",
                "NewVersion": "",
                "UpdateAvailable": false,
                "Cancellable": false,
                "UpdateStatus": "COMPLETED",
                "Description": "There is no software update available for this
 domain.",
                "AutomatedUpdateDate": 0.0
            },
            "DomainEndpointOptions": {
                "EnforceHTTPS": false,
                "TLSSecurityPolicy": "Policy-Min-TLS-1-0-2019-07"
            },
            "AdvancedSecurityOptions": {
                "Enabled": false,
                "InternalUserDatabaseEnabled": false
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Elasticsearch Service 開發人員指南》中的建立和管理 Amazon Elasticsearch Service 網域。 Amazon Elasticsearch Service

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeElasticsearchDomains。

describe-reserved-elasticsearch-instances

以下程式碼範例顯示如何使用 describe-reserved-elasticsearch-instances。

AWS CLI

檢視所有預留執行個體

下列describe-elasticsearch-domains範例提供您在區域中預留的所有執行個體的摘要。

```
aws es describe-reserved-elasticsearch-instances
```

輸出：

```
{
    "ReservedElasticsearchInstances": [{
        "FixedPrice": 100.0,
        "ReservedElasticsearchInstanceOfferingId":
 "1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a",
        "ReservationName": "my-reservation",
        "PaymentOption": "PARTIAL_UPFRONT",
        "UsagePrice": 0.0,
        "ReservedElasticsearchInstanceId": "9a8a7a6a-5a4a-3a2a-1a0a-9a8a7a6a5a4a",
        "RecurringCharges": [{
            "RecurringChargeAmount": 0.603,
            "RecurringChargeFrequency": "Hourly"
        }],
        "State": "payment-pending",
        "StartTime": 1522872571.229,
        "ElasticsearchInstanceCount": 3,
        "Duration": 31536000,
        "ElasticsearchInstanceType": "m4.2xlarge.elasticsearch",
        "CurrencyCode": "USD"
    }]
}
```

如需詳細資訊，請參閱《Amazon Elasticsearch Service 開發人員指南》中的預留執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReservedElasticsearchInstances。

`list-domain-names`

以下程式碼範例顯示如何使用 `list-domain-names`。

AWS CLI

列出所有網域

下列`list-domain-names`範例提供區域中所有網域的快速摘要。

```
aws es list-domain-names
```

輸出：

```
{
    "DomainNames": [{
            "DomainName": "cli-example-1"
        },
        {
            "DomainName": "cli-example-2"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Elasticsearch Service 開發人員指南》中的建立和管理 Amazon Elasticsearch Service 網域。 Amazon Elasticsearch Service

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDomainNames。

OpsWorks 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 OpsWorks。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`assign-instance`

以下程式碼範例顯示如何使用 `assign-instance`。

AWS CLI

將已註冊的執行個體指派給 layer

下列範例會將已註冊的執行個體指派給自訂 layer。

```
aws opsworks --region us-east-1 assign-instance --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2 --layer-ids 26cf1d32-6876-42fa-bbf1-9cadc0bff938
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的將註冊的執行個體指派給 Layer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssignInstance。

`assign-volume`

以下程式碼範例顯示如何使用 `assign-volume`。

AWS CLI

將已註冊的磁碟區指派給執行個體

下列範例會將已註冊的 Amazon Elastic Block Store (Amazon EBS) 磁碟區指派給執行個體。磁碟區是透過磁碟區 ID 來識別，這是當您向堆疊註冊磁碟區時， AWS OpsWorks 指派的 GUID，而不是 Amazon Elastic Compute Cloud (Amazon EC2) 磁碟區 ID。執行 之前`assign-volume`，您必須先執行 `update-volume`，將掛載點指派給磁碟區。

```
aws opsworks --region us-east-1 assign-volume --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2 --volume-id 26cf1d32-6876-42fa-bbf1-9cadc0bff938
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的將 Amazon EBS 磁碟區指派給執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssignVolume。

associate-elastic-ip

以下程式碼範例顯示如何使用 associate-elastic-ip。

AWS CLI

將彈性 IP 地址與執行個體建立關聯

下列範例會將彈性 IP 地址與指定的執行個體建立關聯。

```
aws opsworks --region us-east-1 associate-elastic-ip --instance-
id dfe18b02-5327-493d-91a4-c5c0c448927f --elastic-ip 54.148.130.96
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的資源管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateElasticIp。

attach-elastic-load-balancer

以下程式碼範例顯示如何使用 attach-elastic-load-balancer。

AWS CLI

將負載平衡器連接到 layer

下列範例會將以其名稱識別的負載平衡器連接到指定的 layer。

```
aws opsworks --region us-east-1 attach-elastic-load-balancer --elastic-load-
balancer-name Java-LB --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的 Elastic Load Balancing。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachElasticLoadBalancer。

create-app

以下程式碼範例顯示如何使用 create-app。

AWS CLI

範例 1：建立應用程式

下列範例會從存放在 GitHub 儲存庫中的程式碼建立名為 SimplePHPApp 的 PHP 應用程式。命令使用應用程式來源定義的速記格式。

```
aws opsworks create-app \
    --region us-east-1 \
    --stack-id f6673d70-32e6-4425-8999-265dd002fec7 \
    --name SimplePHPApp \
    --type php \
    --app-source Type=git,Url=git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git,Revision=version1
```

輸出：

```
{
    "AppId": "6cf5163c-a951-444f-a8f7-3716be75f2a2"
}
```

範例 2：使用連接的資料庫建立應用程式

下列範例會從存放在公有 S3 儲存貯體中 .zip 封存檔中的程式碼建立 JSP 應用程式。它會連接 RDS 資料庫執行個體，做為應用程式的資料存放區。應用程式和資料庫來源是在執行命令的目錄中的個別 JSON 檔案中定義。

```
aws opsworks create-app \
    --region us-east-1 \
    --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8 \
    --name SimpleJSP \
    --type java \
```

```
    --app-source file://appsource.json \
    --data-sources file://datasource.json
```

應用程式來源資訊位於 中appsource.json，並包含下列項目。

```
{
    "Type": "archive",
    "Url": "https://s3.amazonaws.com/opsworks-demo-assets/simplejsp.zip"
}
```

資料庫來源資訊位於 中datasource.json，並包含下列項目。

```
[
    {
        "Type": "RdsDbInstance",
        "Arn": "arn:aws:rds:us-west-2:123456789012:db:clitestdb",
        "DatabaseName": "mydb"
    }
]
```

注意：對於 RDS 資料庫執行個體，您必須先使用 register-rds-db-instance向堆疊註冊執行個體。對於 MySQL App Server 執行個體，將 Type設定為 OpsworksMysqlInstance。這些執行個體是由 AWS OpsWorks 建立，因此不需要註冊。

輸出：

```
{
    "AppId": "26a61ead-d201-47e3-b55c-2a7c666942f8"
}
```

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的新增應用程式。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateApp。

create-deployment

以下程式碼範例顯示如何使用 create-deployment。

AWS CLI

範例 1：部署應用程式並執行堆疊命令

下列範例示範如何使用 create-deployment命令來部署應用程式和執行堆疊命令。請注意，指定命令的 JSON 物件中的引號 (") 字元前面都是逸出字元 (\)。如果沒有逸出字元，命令可能會傳回無效的 JSON 錯誤。

下列create-deployment範例會將應用程式部署到指定的堆疊。

```
aws opsworks create-deployment \
    --stack-id cfb7e082-ad1d-4599-8e81-de1c39ab45bf \
    --app-id 307be5c8-d55d-47b5-bd6e-7bd417c6c7eb
    --command "{\"Name\":\"deploy\"}"
```

輸出：

```
{
    "DeploymentId": "5746c781-df7f-4c87-84a7-65a119880560"
}
```

範例 2：部署 Rails 應用程式並遷移資料庫

下列create-deployment命令會將 Ruby on Rails 應用程式部署到指定的堆疊，並遷移資料庫。

```
aws opsworks create-deployment \
    --stack-id cfb7e082-ad1d-4599-8e81-de1c39ab45bf \
    --app-id 307be5c8-d55d-47b5-bd6e-7bd417c6c7eb \
    --command "{\"Name\":\"deploy\", \"Args\":{\"migrate\":[\"true\"]}}"
```

輸出：

```
{
    "DeploymentId": "5746c781-df7f-4c87-84a7-65a119880560"
}
```

如需部署的詳細資訊，請參閱 AWS OpsWorks 使用者指南中的部署應用程式。

範例 3：執行配方

下列create-deployment命令會在指定堆疊的執行個體上執行自訂配方 phpapp::appsetup。

```
aws opsworks create-deployment \
    --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb \
```

```
    --command "{\"Name\":\"execute_recipes\", \"Args\":{\"recipes\":
[\"phpapp::appsetup\"]}}"
```

輸出：

```
{
    "DeploymentId": "5cbaa7b9-4e09-4e53-aa1b-314fbd106038"
}
```

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的執行堆疊命令。

範例 4：安裝相依性

下列create-deployment命令會在指定堆疊中的執行個體上安裝相依性，例如套件或 Ruby Gem。

```
aws opsworks create-deployment \
    --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb \
    --command "{\"Name\":\"install_dependencies\"}"
```

輸出：

```
{
    "DeploymentId": "aef5b255-8604-4928-81b3-9b0187f962ff"
}
```

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的執行堆疊命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDeployment。

create-instance

以下程式碼範例顯示如何使用 create-instance。

AWS CLI

建立執行個體

下列create-instance命令會在指定的堆疊中建立名為 myinstance1 的 m1.large Amazon Linux 執行個體。執行個體會指派給一個 layer。

```
aws opsworks --region us-east-1 create-instance --stack-id 935450cc-61e0-4b03-
a3e0-160ac817d2bb --layer-ids 5c8c272a-f2d5-42e3-8245-5bf3927cb65b --
hostname myinstance1 --instance-type m1.large --os "Amazon Linux"
```

若要使用自動產生的名稱，請呼叫 get-hostname-suggestion，這會根據您建立堆疊時指定的主題產生主機名稱。然後將該名稱傳遞至主機名稱引數。

輸出：

```
{
  "InstanceId": "5f9adeaa-c94c-42c6-aeef-28a5376002cd"
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的將執行個體新增至 Layer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateInstance。

create-layer

以下程式碼範例顯示如何使用 create-layer。

AWS CLI

建立圖層

下列create-layer命令會在指定的堆疊中建立名為 MyPHPLayer 的 PHP App Server 層。

```
aws opsworks create-layer --region us-east-1 --stack-
id f6673d70-32e6-4425-8999-265dd002fec7 --type php-app --name MyPHPLayer --
shortname myphplayer
```

輸出：

```
{
  "LayerId": "0b212672-6b4b-40e4-8a34-5a943cf2e07a"
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的如何建立 Layer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLayer。

create-server

以下程式碼範例顯示如何使用 create-server。

AWS CLI

建立伺服器

下列create-server範例automate-06會在您的預設區域中建立名為 的新 Chef Automate 伺服器。請注意，預設值用於大多數其他設定，例如要保留的備份數量，以及維護和備份開始時間。執行create-server命令之前，請先完成《AWS OpsWorks for Chef Automate 使用者指南》中的 OpsWorks AWS for Chef Automate 入門中的先決條件。

```
aws opsworks-cm create-server \
    --engine "ChefAutomate" \
    --instance-profile-arn "arn:aws:iam::012345678901:instance-profile/aws-opsworks-cm-ec2-role" \
    --instance-type "t2.medium" \
    --server-name "automate-06" \
    --service-role-arn "arn:aws:iam::012345678901:role/aws-opsworks-cm-service-role"
```

輸出：

```
{
    "Server": {
        "AssociatePublicIpAddress": true,
        "BackupRetentionCount": 10,
        "CreatedAt": 2019-12-29T13:38:47.520Z,
        "DisableAutomatedBackup": FALSE,
        "Endpoint": "https://opsworks-cm.us-east-1.amazonaws.com",
        "Engine": "ChefAutomate",
        "EngineAttributes": [
            {
                "Name": "CHEF_AUTOMATE_ADMIN_PASSWORD",
                "Value": "1Example1"
            }
        ],
        "EngineModel": "Single",
        "EngineVersion": "2019-08",
```

```
        "InstanceProfileArn": "arn:aws:iam::012345678901:instance-profile/aws-
opsworks-cm-ec2-role",
        "InstanceType": "t2.medium",
        "PreferredBackupWindow": "Sun:02:00",
        "PreferredMaintenanceWindow": "00:00",
        "SecurityGroupIds": [ "sg-12345678" ],
        "ServerArn": "arn:aws:iam::012345678901:instance/automate-06-1010V4UU2WRM2",
        "ServerName": "automate-06",
        "ServiceRoleArn": "arn:aws:iam::012345678901:role/aws-opsworks-cm-service-
role",
        "Status": "CREATING",
        "SubnetIds": [ "subnet-12345678" ]
    }
}
```

如需詳細資訊，請參閱 AWS OpsWorks for Chef Automate API 參考中的 CreateServer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateServer。

create-stack

以下程式碼範例顯示如何使用 create-stack。

AWS CLI

建立堆疊

下列create-stack命令會建立名為 CLI Stack 的堆疊。

```
aws opsworks create-stack --name "CLI Stack" --stack-region "us-east-1" --service-
role-arn arn:aws:iam::123456789012:role/aws-opsworks-service-role --default-
instance-profile-arn arn:aws:iam::123456789012:instance-profile/aws-opsworks-ec2-
role --region us-east-1
```

service-role-arn 和 default-instance-profile-arn 是必要參數。您通常會在建立第一個堆疊時，使用 AWS OpsWorks 為您建立的堆疊。若要取得帳戶的 Amazon Resource Name (ARNs)，請前往 IAM 主控台，選擇導覽面板Roles中的 ，選擇角色或設定檔，然後選擇Summary標籤。

輸出：

```
{
```

```
  "StackId": "f6673d70-32e6-4425-8999-265dd002fec7"
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的建立新堆疊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateStack。

`create-user-profile`

以下程式碼範例顯示如何使用 `create-user-profile`。

AWS CLI

建立使用者設定檔

您可以透過呼叫 create-user-profile 來建立使用者設定檔，將 AWS Identity and Access Manager (IAM) 使用者匯入至 AWS OpsWorks。下列範例會為由 Amazon Resource Name (ARN) 識別的 cli-user-test IAM 使用者建立使用者設定檔。此範例會為使用者指派 的 SSH 使用者名稱，myusername並啟用自我管理，以允許使用者指定 SSH 公有金鑰。

```
aws opsworks --region us-east-1 create-user-profile --iam-user-
arn arn:aws:iam::123456789102:user/cli-user-test --ssh-username myusername --allow-
self-management
```

輸出：

```
{
  "IamUserArn": "arn:aws:iam::123456789102:user/cli-user-test"
}
```

秘訣：此命令會將 IAM 使用者匯入至 AWS OpsWorks，但僅限於附加政策授予的許可。您可以使用 `set-permissions`命令授予 per-stack AWS OpsWorks 許可。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的將使用者匯入至 OpsWorks。 AWS OpsWorks

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateUserProfile。

delete-app

以下程式碼範例顯示如何使用 delete-app。

AWS CLI

刪除應用程式

下列範例會刪除指定的應用程式，以其應用程式 ID 識別。您可以前往 AWS OpsWorks 主控台上的應用程式詳細資訊頁面，或執行 describe-apps命令來取得應用程式 ID。

```
aws opsworks delete-app --region us-east-1 --app-id 577943b9-2ec1-4baf-
a7bf-1d347601edc5
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的應用程式。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteApp。

delete-instance

以下程式碼範例顯示如何使用 delete-instance。

AWS CLI

刪除執行個體

下列delete-instance範例會刪除指定的執行個體，以其執行個體 ID 識別。您可以在 AWS OpsWorks 主控台中開啟執行個體的詳細資訊頁面，或執行 describe-instances命令來尋找執行個體 ID。

如果執行個體在線上，您必須先呼叫 來停止執行個體stop-instance，然後必須等到執行個體停止。執行 describe-instances 以檢查執行個體狀態。

若要移除執行個體的 Amazon EBS 磁碟區或彈性 IP 地址，請分別新增 --delete-volumes或 --delete-elastic-ip引數。

```
aws opsworks delete-instance \
    --region us-east-1 \
    --instance-id 3a21cfac-4a1f-4ce2-a921-b2cfba6f7771
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的 DeleteOpsWorks 執行個體。 AWS OpsWorks

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteInstance。

delete-layer

以下程式碼範例顯示如何使用 delete-layer。

AWS CLI

刪除圖層

下列範例會刪除指定的 layer，以其 layer ID 識別。您可以前往 AWS OpsWorks 主控台上的 layer 詳細資訊頁面，或執行 describe-layers命令來取得 layer ID。

注意：刪除 layer 之前，您必須使用 delete-instance 來刪除 layer 的所有執行個體。

```
aws opsworks delete-layer --region us-east-1 --layer-id a919454e-b816-4598-b29a-5796afb498ed
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的 DeleteOpsWorks 執行個體。 AWS OpsWorks

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLayer。

delete-stack

以下程式碼範例顯示如何使用 delete-stack。

AWS CLI

刪除堆疊

下列範例會刪除指定的堆疊，以其堆疊 ID 識別。您可以按一下 AWS OpsWorks 主控台上的堆疊設定，或執行 describe-stacks命令來取得堆疊 ID。

注意：刪除 layer 之前，您必須使用 delete-app、 delete-instance和 delete-layer 來刪除堆疊的所有應用程式、執行個體和 layer。

```
aws opsworks delete-stack --region us-east-1 --stack-
id 154a9d89-7e9e-433b-8de8-617e53756c84
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的關閉堆疊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteStack。

delete-user-profile

以下程式碼範例顯示如何使用 delete-user-profile。

AWS CLI

刪除使用者設定檔，並從 AWS OpsWorks 移除 IAM 使用者

下列範例會刪除指定的 AWS Identity and Access Management (IAM) 使用者的使用者設定檔，該使用者由 Amazon Resource Name (ARN) 識別。操作會從 AWS OpsWorks 移除使用者，但不會刪除 IAM 使用者。您必須針對該任務使用 IAM 主控台、CLI 或 API。

```
aws opsworks --region us-east-1 delete-user-profile --iam-user-
arn arn:aws:iam::123456789102:user/cli-user-test
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的將使用者匯入至 OpsWorks。 AWS OpsWorks

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteUserProfile。

deregister-elastic-ip

以下程式碼範例顯示如何使用 deregister-elastic-ip。

AWS CLI

從堆疊取消註冊彈性 IP 地址

下列範例會從其堆疊取消註冊由其 IP 地址識別的彈性 IP 地址。

```
aws opsworks deregister-elastic-ip --region us-east-1 --elastic-ip 54.148.130.96
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的取消註冊彈性 IP 地址。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterElasticIp。

deregister-instance

以下程式碼範例顯示如何使用 deregister-instance。

AWS CLI

從堆疊取消註冊已註冊的執行個體

下列deregister-instance命令會從其堆疊取消註冊已註冊的執行個體。

```
aws opsworks --region us-east-1 deregister-instance --instance-id 4d6d1710-
ded9-42a1-b08e-b043ad7af1e2
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的取消註冊已註冊的執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterInstance。

deregister-rds-db-instance

以下程式碼範例顯示如何使用 deregister-rds-db-instance。

AWS CLI

從堆疊取消註冊 Amazon RDS 資料庫執行個體

下列範例會從其堆疊取消註冊由其 ARN 識別的 RDS 資料庫執行個體。

```
aws opsworks deregister-rds-db-instance --region us-east-1 --rds-db-instance-
arn arn:aws:rds:us-west-2:123456789012:db:clitestdb
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱《ASW OpsWorks 使用者指南》中的取消註冊 Amazon RDS 執行個體。

執行個體 ID：clitestdb Master Usernams：cliuser Master PWD： some23！pwd 資料庫名稱：mydb aws opsworks deregister-rds-db-instance --region us-east-1 --rds-db-instance-arn arn：aws：rds：us-west-2：645732743964：db：clitestdb

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterRdsDbInstance。

deregister-volume

以下程式碼範例顯示如何使用 `deregister-volume`。

AWS CLI

取消註冊 Amazon EBS 磁碟區

下列範例會從其堆疊取消註冊 EBS 磁碟區。磁碟區會以磁碟區 ID 識別，這是當您向堆疊註冊磁碟區時 AWS OpsWorks 指派的 GUID，而不是 EC2 磁碟區 ID。

```
aws opsworks deregister-volume --region us-east-1 --volume-id 5c48ef52-3144-4bf5-
beaa-fda4deb23d4d
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的取消註冊 Amazon EBS 磁碟區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的DeregisterVolume。

describe-apps

以下程式碼範例顯示如何使用 describe-apps。

AWS CLI

描述應用程式

下列describe-apps命令說明指定堆疊中的應用程式。

```
aws opsworks describe-apps \
    --stack-id 38ee91e2-abdc-4208-a107-0b7168b3cc7a \
    --region us-east-1
```

輸出：

```
{
    "Apps": [
        {
            "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
            "AppSource": {
            "Url": "https://s3-us-west-2.amazonaws.com/opsworks-demo-assets/
simplejsp.zip",
            "Type": "archive"
        },
            "Name": "SimpleJSP",
            "EnableSsl": false,
            "SslConfiguration": {},
            "AppId": "da1decc1-0dff-43ea-ad7c-bb667cd87c8b",
            "Attributes": {
            "RailsEnv": null,
            "AutoBundleOnDeploy": "true",
            "DocumentRoot": "ROOT"
        },
            "Shortname": "simplejsp",
            "Type": "other",
            "CreatedAt": "2013-08-01T21:46:54+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的應用程式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeApps。

describe-commands

以下程式碼範例顯示如何使用 describe-commands。

AWS CLI

描述命令

下列describe-commands命令說明指定執行個體中的命令。

```
aws opsworks describe-commands \
    --instance-id 8c2673b9-3fe5-420d-9cfa-78d875ee7687 \
    --region us-east-1
```

輸出：

```
{
    "Commands": [
        {
            "Status": "successful",
            "CompletedAt": "2013-07-25T18:57:47+00:00",
            "InstanceId": "8c2673b9-3fe5-420d-9cfa-78d875ee7687",
            "DeploymentId": "6ed0df4c-9ef7-4812-8dac-d54a05be1029",
            "AcknowledgedAt": "2013-07-25T18:57:41+00:00",
            "LogUrl": "https://s3.amazonaws.com/<bucket-name>/logs/008c1a91-
ec59-4d51-971d-3adff54b00cc?AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
 &Expires=1375394373&Signature=HkXil6UuNfxTCC37EPQAa462E1E%3D&response-cache-
control=private&response-content-encoding=gzip&response-content- type=text%2Fplain",
            "Type": "undeploy",
            "CommandId": "008c1a91-ec59-4d51-971d-3adff54b00cc",
            "CreatedAt": "2013-07-25T18:57:34+00:00",
            "ExitCode": 0
        },
        {
            "Status": "successful",
            "CompletedAt": "2013-07-25T18:55:40+00:00",
            "InstanceId": "8c2673b9-3fe5-420d-9cfa-78d875ee7687",
            "DeploymentId": "19d3121e-d949-4ff2-9f9d-94eac087862a",
            "AcknowledgedAt": "2013-07-25T18:55:32+00:00",
            "LogUrl": "https://s3.amazonaws.com/<bucket-name>/
logs/899d3d64-0384-47b6-a586-33433aad117c?AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
 &Expires=1375394373&Signature=xMsJvtLuUqWmsr8s%2FAjVru0BtRs%3D&response-cache-
control=private&response-content-encoding=gzip&response-conten t-type=text%2Fplain",
```

```
            "Type": "deploy",
            "CommandId": "899d3d64-0384-47b6-a586-33433aad117c",
            "CreatedAt": "2013-07-25T18:55:29+00:00",
            "ExitCode": 0
        }
    ]
}
```

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的 OpsWorks 生命週期事件。 AWS OpsWorks

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCommands。

describe-deployments

以下程式碼範例顯示如何使用 describe-deployments。

AWS CLI

描述部署

下列describe-deployments命令說明指定堆疊中的部署。

```
aws opsworks --region us-east-1 describe-deployments --stack-id 38ee91e2-abdc-4208-a107-0b7168b3cc7a
```

輸出：

```
{
  "Deployments": [
      {
          "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
          "Status": "successful",
          "CompletedAt": "2013-07-25T18:57:49+00:00",
          "DeploymentId": "6ed0df4c-9ef7-4812-8dac-d54a05be1029",
          "Command": {
              "Args": {},
              "Name": "undeploy"
          },
          "CreatedAt": "2013-07-25T18:57:34+00:00",
          "Duration": 15,
          "InstanceIds": [
              "8c2673b9-3fe5-420d-9cfa-78d875ee7687",
```

```
                "9e588a25-35b2-4804-bd43-488f85ebe5b7"
            ]
        },
        {
            "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
            "Status": "successful",
            "CompletedAt": "2013-07-25T18:56:41+00:00",
            "IamUserArn": "arn:aws:iam::123456789012:user/someuser",
            "DeploymentId": "19d3121e-d949-4ff2-9f9d-94eac087862a",
            "Command": {
                "Args": {},
                "Name": "deploy"
            },
            "InstanceIds": [
                "8c2673b9-3fe5-420d-9cfa-78d875ee7687",
                "9e588a25-35b2-4804-bd43-488f85ebe5b7"
            ],
            "Duration": 72,
            "CreatedAt": "2013-07-25T18:55:29+00:00"
        }
    ]
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的部署應用程式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDeployments。

describe-elastic-ips

以下程式碼範例顯示如何使用 describe-elastic-ips。

AWS CLI

描述彈性 IP 執行個體

下列describe-elastic-ips命令說明指定執行個體中的彈性 IP 地址。

```
aws opsworks --region us-east-1 describe-elastic-ips --instance-id b62f3e04-e9eb-436c-a91f-d9e9a396b7b0
```

輸出：

```
{
  "ElasticIps": [
      {
          "Ip": "192.0.2.0",
          "Domain": "standard",
          "Region": "us-west-2"
      }
  ]
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeElasticIps。

describe-elastic-load-balancers

以下程式碼範例顯示如何使用 describe-elastic-load-balancers。

AWS CLI

描述堆疊的彈性負載平衡器

下列describe-elastic-load-balancers命令說明指定堆疊的負載平衡器。

```
aws opsworks --region us-west-2 describe-elastic-load-balancers --stack-
id 6f4660e5-37a6-4e42-bfa0-1358ebd9c182
```

輸出：此特定堆疊有一個負載平衡器。

```
{
  "ElasticLoadBalancers": [
      {
          "SubnetIds": [
              "subnet-60e4ea04",
              "subnet-66e1c110"
          ],
          "Ec2InstanceIds": [],
          "ElasticLoadBalancerName": "my-balancer",
          "Region": "us-west-2",
          "LayerId": "344973cb-bf2b-4cd0-8d93-51cd819bab04",
          "AvailabilityZones": [
```

```
                "us-west-2a",
                "us-west-2b"
            ],
            "VpcId": "vpc-b319f9d4",
            "StackId": "6f4660e5-37a6-4e42-bfa0-1358ebd9c182",
            "DnsName": "my-balancer-2094040179.us-west-2.elb.amazonaws.com"
        }
    ]
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的應用程式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeElasticLoadBalancers。

describe-instances

以下程式碼範例顯示如何使用 describe-instances。

AWS CLI

描述執行個體

下列describe-instances命令說明指定堆疊中的執行個體：

```
aws opsworks --region us-east-1 describe-instances --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

輸出：下列輸出範例適用於具有兩個執行個體的堆疊。第一個是已註冊的 EC2 執行個體，第二個由 AWS OpsWorks 建立。

```
{
  "Instances": [
    {
      "StackId": "71c7ca72-55ae-4b6a-8ee1-a8dcded3fa0f",
      "PrivateDns": "ip-10-31-39-66.us-west-2.compute.internal",
      "LayerIds": [
        "26cf1d32-6876-42fa-bbf1-9cadc0bff938"
      ],
      "EbsOptimized": false,
      "ReportedOs": {
        "Version": "14.04",
```

```
        "Name": "ubuntu",
        "Family": "debian"
      },
      "Status": "online",
      "InstanceId": "4d6d1710-ded9-42a1-b08e-b043ad7af1e2",
      "SshKeyName": "US-West-2",
      "InfrastructureClass": "ec2",
      "RootDeviceVolumeId": "vol-d08ec6c1",
      "SubnetId": "subnet-b8de0ddd",
      "InstanceType": "t1.micro",
      "CreatedAt": "2015-02-24T20:52:49+00:00",
      "AmiId": "ami-35501205",
      "Hostname": "ip-192-0-2-0",
      "Ec2InstanceId": "i-5cd23551",
      "PublicDns": "ec2-192-0-2-0.us-west-2.compute.amazonaws.com",
      "SecurityGroupIds": [
        "sg-c4d3f0a1"
      ],
      "Architecture": "x86_64",
      "RootDeviceType": "ebs",
      "InstallUpdatesOnBoot": true,
      "Os": "Custom",
      "VirtualizationType": "paravirtual",
      "AvailabilityZone": "us-west-2a",
      "PrivateIp": "10.31.39.66",
      "PublicIp": "192.0.2.06",
      "RegisteredBy": "arn:aws:iam::123456789102:user/AWS/OpsWorks/OpsWorks-
EC2Register-i-5cd23551"
    },
    {
      "StackId": "71c7ca72-55ae-4b6a-8ee1-a8dcded3fa0f",
      "PrivateDns": "ip-10-31-39-158.us-west-2.compute.internal",
      "SshHostRsaKeyFingerprint": "69:6b:7b:8b:72:f3:ed:23:01:00:05:bc:9f:a4:60:c1",
      "LayerIds": [
        "26cf1d32-6876-42fa-bbf1-9cadc0bff938"
      ],
      "EbsOptimized": false,
      "ReportedOs": {},
      "Status": "booting",
      "InstanceId": "9b137a0d-2f5d-4cc0-9704-13da4b31fdcb",
      "SshKeyName": "US-West-2",
      "InfrastructureClass": "ec2",
      "RootDeviceVolumeId": "vol-e09dd5f1",
      "SubnetId": "subnet-b8de0ddd",
```

```
      "InstanceProfileArn": "arn:aws:iam::123456789102:instance-profile/aws-
opsworks-ec2-role",
      "InstanceType": "c3.large",
      "CreatedAt": "2015-02-24T21:29:33+00:00",
      "AmiId": "ami-9fc29baf",
      "SshHostDsaKeyFingerprint": "fc:87:95:c3:f5:e1:3b:9f:d2:06:6e:62:9a:35:27:e8",
      "Ec2InstanceId": "i-8d2dca80",
      "PublicDns": "ec2-192-0-2-1.us-west-2.compute.amazonaws.com",
      "SecurityGroupIds": [
        "sg-b022add5",
        "sg-b122add4"
      ],
      "Architecture": "x86_64",
      "RootDeviceType": "ebs",
      "InstallUpdatesOnBoot": true,
      "Os": "Amazon Linux 2014.09",
      "VirtualizationType": "paravirtual",
      "AvailabilityZone": "us-west-2a",
      "Hostname": "custom11",
      "PrivateIp": "10.31.39.158",
      "PublicIp": "192.0.2.0"
    }
  ]
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的執行個體。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeInstances。

describe-layers

以下程式碼範例顯示如何使用 describe-layers。

AWS CLI

描述堆疊的圖層

下列describe-layers命令說明指定堆疊中的圖層：

```
aws opsworks --region us-east-1 describe-layers --stack-id 38ee91e2-abdc-4208-
a107-0b7168b3cc7a
```

輸出：

```
{
  "Layers": [
      {
          "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
          "Type": "db-master",
          "DefaultSecurityGroupNames": [
              "AWS-OpsWorks-DB-Master-Server"
          ],
          "Name": "MySQL",
          "Packages": [],
          "DefaultRecipes": {
              "Undeploy": [],
              "Setup": [
                  "opsworks_initial_setup",
                  "ssh_host_keys",
                  "ssh_users",
                  "mysql::client",
                  "dependencies",
                  "ebs",
                  "opsworks_ganglia::client",
                  "mysql::server",
                  "dependencies",
                  "deploy::mysql"
              ],
              "Configure": [
                  "opsworks_ganglia::configure-client",
                  "ssh_users",
                  "agent_version",
                  "deploy::mysql"
              ],
              "Shutdown": [
                  "opsworks_shutdown::default",
                  "mysql::stop"
              ],
              "Deploy": [
                  "deploy::default",
                  "deploy::mysql"
              ]
          },
          "CustomRecipes": {
              "Undeploy": [],
              "Setup": [],
```

```
            "Configure": [],
            "Shutdown": [],
            "Deploy": []
        },
        "EnableAutoHealing": false,
        "LayerId": "41a20847-d594-4325-8447-171821916b73",
        "Attributes": {
            "MysqlRootPasswordUbiquitous": "true",
            "RubygemsVersion": null,
            "RailsStack": null,
            "HaproxyHealthCheckMethod": null,
            "RubyVersion": null,
            "BundlerVersion": null,
            "HaproxyStatsPassword": null,
            "PassengerVersion": null,
            "MemcachedMemory": null,
            "EnableHaproxyStats": null,
            "ManageBundler": null,
            "NodejsVersion": null,
            "HaproxyHealthCheckUrl": null,
            "MysqlRootPassword": "*****FILTERED*****",
            "GangliaPassword": null,
            "GangliaUser": null,
            "HaproxyStatsUrl": null,
            "GangliaUrl": null,
            "HaproxyStatsUser": null
        },
        "Shortname": "db-master",
        "AutoAssignElasticIps": false,
        "CustomSecurityGroupIds": [],
        "CreatedAt": "2013-07-25T18:11:19+00:00",
        "VolumeConfigurations": [
            {
                "MountPoint": "/vol/mysql",
                "Size": 10,
                "NumberOfDisks": 1
            }
        ]
    },
    {
        "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
        "Type": "custom",
        "DefaultSecurityGroupNames": [
            "AWS-OpsWorks-Custom-Server"
```

```
            ],
            "Name": "TomCustom",
            "Packages": [],
            "DefaultRecipes": {
                "Undeploy": [],
                "Setup": [
                    "opsworks_initial_setup",
                    "ssh_host_keys",
                    "ssh_users",
                    "mysql::client",
                    "dependencies",
                    "ebs",
                    "opsworks_ganglia::client"
                ],
                "Configure": [
                    "opsworks_ganglia::configure-client",
                    "ssh_users",
                    "agent_version"
                ],
                "Shutdown": [
                    "opsworks_shutdown::default"
                ],
                "Deploy": [
                    "deploy::default"
                ]
            },
            "CustomRecipes": {
                "Undeploy": [],
                "Setup": [
                    "tomcat::setup"
                ],
                "Configure": [
                    "tomcat::configure"
                ],
                "Shutdown": [],
                "Deploy": [
                    "tomcat::deploy"
                ]
            },
            "EnableAutoHealing": true,
            "LayerId": "e6cbcd29-d223-40fc-8243-2eb213377440",
            "Attributes": {
                "MysqlRootPasswordUbiquitous": null,
                "RubygemsVersion": null,
```

```
                "RailsStack": null,
                "HaproxyHealthCheckMethod": null,
                "RubyVersion": null,
                "BundlerVersion": null,
                "HaproxyStatsPassword": null,
                "PassengerVersion": null,
                "MemcachedMemory": null,
                "EnableHaproxyStats": null,
                "ManageBundler": null,
                "NodejsVersion": null,
                "HaproxyHealthCheckUrl": null,
                "MysqlRootPassword": null,
                "GangliaPassword": null,
                "GangliaUser": null,
                "HaproxyStatsUrl": null,
                "GangliaUrl": null,
                "HaproxyStatsUser": null
            },
            "Shortname": "tomcustom",
            "AutoAssignElasticIps": false,
            "CustomSecurityGroupIds": [],
            "CreatedAt": "2013-07-25T18:12:53+00:00",
            "VolumeConfigurations": []
        }
    ]
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的圖層。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLayers。

describe-load-based-auto-scaling

以下程式碼範例顯示如何使用 `describe-load-based-auto-scaling`。

AWS CLI

描述 layer 的負載型擴展組態

下列範例說明指定 layer 的負載型擴展組態。圖層的識別方式是其圖層 ID，您可以在圖層的詳細資訊頁面或執行 找到該圖層`describe-layers`。

```
aws opsworks describe-load-based-auto-scaling --region us-east-1 --layer-
ids 6bec29c9-c866-41a0-aba5-fa3e374ce2a1
```

輸出：範例 layer 具有單一負載型執行個體。

```
{
  "LoadBasedAutoScalingConfigurations": [
    {
      "DownScaling": {
        "IgnoreMetricsTime": 10,
        "ThresholdsWaitTime": 10,
        "InstanceCount": 1,
        "CpuThreshold": 30.0
      },
      "Enable": true,
      "UpScaling": {
        "IgnoreMetricsTime": 5,
        "ThresholdsWaitTime": 5,
        "InstanceCount": 1,
        "CpuThreshold": 80.0
      },
      "LayerId": "6bec29c9-c866-41a0-aba5-fa3e374ce2a1"
    }
  ]
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的 Automatic Load-based Scaling 的運作方式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLoadBasedAutoScaling。

describe-my-user-profile

以下程式碼範例顯示如何使用 describe-my-user-profile。

AWS CLI

取得使用者的設定檔

下列範例顯示如何取得執行 命令之 AWS Identity and Access Management (IAM) 使用者的設定檔。

```
aws opsworks --region us-east-1 describe-my-user-profile
```

輸出：為了簡潔起見，大部分使用者的 SSH 公有金鑰會被省略符號 (...) 取代。

```
{
  "UserProfile": {
    "IamUserArn": "arn:aws:iam::123456789012:user/myusername",
    "SshPublicKey": "ssh-rsa AAAAB3NzaC1yc2EAAAABJQ...3LQ4aX9jpxQw== rsa-
key-20141104",
    "Name": "myusername",
    "SshUsername": "myusername"
  }
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的將使用者匯入至 OpsWorks。 AWS OpsWorks

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeMyUserProfile。

describe-permissions

以下程式碼範例顯示如何使用 describe-permissions。

AWS CLI

取得使用者的 per-stack AWS OpsWorks 許可層級

下列範例示範如何在指定的堆疊上取得 AWS Identity and Access Management (IAM) 使用者的許可層級。

```
aws opsworks --region us-east-1 describe-permissions --iam-user-
arn arn:aws:iam::123456789012:user/cli-user-test --stack-id d72553d4-8727-448c-9b00-
f024f0ba1b06
```

輸出：

```
{
  "Permissions": [
    {
      "StackId": "d72553d4-8727-448c-9b00-f024f0ba1b06",
      "IamUserArn": "arn:aws:iam::123456789012:user/cli-user-test",
      "Level": "manage",
      "AllowSudo": true,
      "AllowSsh": true
    }
  ]
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的授予每個堆疊許可層級。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePermissions。

describe-raid-arrays

以下程式碼範例顯示如何使用 describe-raid-arrays。

AWS CLI

描述 RAID 陣列

下列範例說明連接至指定堆疊中執行個體的 RAID 陣列。

```
aws opsworks --region us-east-1 describe-raid-arrays --stack-
id d72553d4-8727-448c-9b00-f024f0ba1b06
```

輸出：以下是具有一個 RAID 陣列之堆疊的輸出。

```
{
  "RaidArrays": [
    {
      "StackId": "d72553d4-8727-448c-9b00-f024f0ba1b06",
      "AvailabilityZone": "us-west-2a",
      "Name": "Created for php-app1",
      "NumberOfDisks": 2,
      "InstanceId": "9f14adbc-ced5-43b6-bf01-e7d0db6cf2f7",
```

```
      "RaidLevel": 0,
      "VolumeType": "standard",
      "RaidArrayId": "f2d4e470-5972-4676-b1b8-bae41ec3e51c",
      "Device": "/dev/md0",
      "MountPoint": "/mnt/workspace",
      "CreatedAt": "2015-02-26T23:53:09+00:00",
      "Size": 100
    }
  ]
}
```

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的 EBS 磁碟區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRaidArrays。

describe-rds-db-instances

以下程式碼範例顯示如何使用 describe-rds-db-instances。

AWS CLI

描述堆疊的已註冊 Amazon RDS 執行個體

下列範例說明使用指定堆疊註冊的 Amazon RDS 執行個體。

```
aws opsworks --region us-east-1 describe-rds-db-instances --stack-
id d72553d4-8727-448c-9b00-f024f0ba1b06
```

輸出：以下是具有一個已註冊 RDS 執行個體之堆疊的輸出。

```
{
  "RdsDbInstances": [
    {
      "Engine": "mysql",
      "StackId": "d72553d4-8727-448c-9b00-f024f0ba1b06",
      "MissingOnRds": false,
      "Region": "us-west-2",
      "RdsDbInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:clitestdb",
      "DbPassword": "*****FILTERED*****",
      "Address": "clitestdb.cdlqlk5uwd0k.us-west-2.rds.amazonaws.com",
      "DbUser": "cliuser",
      "DbInstanceIdentifier": "clitestdb"
```

```
    }
  ]
}
```

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的資源管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRdsDbInstances。

describe-stack-provisioning-parameters

以下程式碼範例顯示如何使用 describe-stack-provisioning-parameters。

AWS CLI

傳回堆疊的佈建參數

下列describe-stack-provisioning-parameters範例會傳回指定堆疊的佈建參數。佈建參數包括設定，例如 OpsWorks 用來在堆疊中的執行個體上管理代理程式的代理程式安裝位置和公有金鑰。

```
aws opsworks describe-stack-provisioning-parameters \
    --stack-id 62744d97-6faf-4ecb-969b-a086fEXAMPLE
```

輸出：

```
{
    "AgentInstallerUrl": "https://opsworks-instance-agent-us-
west-2.s3.amazonaws.com/ID_number/opsworks-agent-installer.tgz",
    "Parameters": {
        "agent_installer_base_url": "https://opsworks-instance-agent-us-
west-2.s3.amazonaws.com",
        "agent_installer_tgz": "opsworks-agent-installer.tgz",
        "assets_download_bucket": "opsworks-instance-assets-us-
west-2.s3.amazonaws.com",
        "charlie_public_key": "-----BEGIN PUBLIC KEY-----PUBLIC_KEY_EXAMPLE\n-----
END PUBLIC KEY-----",
        "instance_service_endpoint": "opsworks-instance-service.us-
west-2.amazonaws.com",
        "instance_service_port": "443",
        "instance_service_region": "us-west-2",
        "instance_service_ssl_verify_peer": "true",
        "instance_service_use_ssl": "true",
```

```
        "ops_works_endpoint": "opsworks.us-west-2.amazonaws.com",
        "ops_works_port": "443",
        "ops_works_region": "us-west-2",
        "ops_works_ssl_verify_peer": "true",
        "ops_works_use_ssl": "true",
        "verbose": "false",
        "wait_between_runs": "30"
    }
}
```

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的執行堆疊命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStackProvisioningParameters。

describe-stack-summary

以下程式碼範例顯示如何使用 `describe-stack-summary`。

AWS CLI

描述堆疊的組態

下列`describe-stack-summary`命令會傳回指定堆疊組態的摘要。

```
aws opsworks --region us-east-1 describe-stack-summary --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

輸出：

```
{
  "StackSummary": {
    "StackId": "8c428b08-a1a1-46ce-a5f8-feddc43771b8",
    "InstancesCount": {
      "Booting": 1
    },
    "Name": "CLITest",
    "AppsCount": 1,
    "LayersCount": 1,
    "Arn": "arn:aws:opsworks:us-west-2:123456789012:stack/8c428b08-a1a1-46ce-a5f8-feddc43771b8/"
  }
```

```
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的 Stacks。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStackSummary。

describe-stacks

以下程式碼範例顯示如何使用 describe-stacks。

AWS CLI

描述堆疊

下列describe-stacks命令說明 帳戶的堆疊。

```
aws opsworks --region us-east-1 describe-stacks
```

輸出：

```
{
  "Stacks": [
    {
      "ServiceRoleArn": "arn:aws:iam::444455556666:role/aws-opsworks-service-role",
      "StackId": "aeb7523e-7c8b-49d4-b866-03aae9d4fbcb",
      "DefaultRootDeviceType": "instance-store",
      "Name": "TomStack-sd",
      "ConfigurationManager": {
        "Version": "11.4",
        "Name": "Chef"
      },
      "UseCustomCookbooks": true,
      "CustomJson": "{\n  \"tomcat\": {\n    \"base_version\": 7,\n    \"java_opts
\": \"-Djava.awt.headless=true -Xmx256m\"\n  },\n  \"datasources\": {\n    \"ROOT\":
 \"jdbc/mydb\"\n  }\n}",
      "Region": "us-east-1",
      "DefaultInstanceProfileArn": "arn:aws:iam::444455556666:instance-profile/aws-
opsworks-ec2-role",
      "CustomCookbooksSource": {
        "Url": "git://github.com/example-repo/tomcustom.git",
        "Type": "git"
```

```
      },
      "DefaultAvailabilityZone": "us-east-1a",
      "HostnameTheme": "Layer_Dependent",
      "Attributes": {
        "Color": "rgb(45, 114, 184)"
      },
      "DefaultOs": "Amazon Linux",
      "CreatedAt": "2013-08-01T22:53:42+00:00"
    },
    {
      "ServiceRoleArn": "arn:aws:iam::444455556666:role/aws-opsworks-service-role",
      "StackId": "40738975-da59-4c5b-9789-3e422f2cf099",
      "DefaultRootDeviceType": "instance-store",
      "Name": "MyStack",
      "ConfigurationManager": {
        "Version": "11.4",
        "Name": "Chef"
      },
      "UseCustomCookbooks": false,
      "Region": "us-east-1",
      "DefaultInstanceProfileArn": "arn:aws:iam::444455556666:instance-profile/aws-
opsworks-ec2-role",
      "CustomCookbooksSource": {},
      "DefaultAvailabilityZone": "us-east-1a",
      "HostnameTheme": "Layer_Dependent",
      "Attributes": {
        "Color": "rgb(45, 114, 184)"
      },
      "DefaultOs": "Amazon Linux",
      "CreatedAt": "2013-10-25T19:24:30+00:00"
    }
  ]
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的 Stacks。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStacks。

describe-timebased-auto-scaling

以下程式碼範例顯示如何使用 describe-timebased-auto-scaling。

AWS CLI

描述執行個體的時間型擴展組態

下列範例說明指定執行個體的時間型擴展組態。執行個體是以其執行個體 ID 識別，您可以在執行個體的詳細資訊頁面或執行 找到。 describe-instances

```
aws opsworks describe-time-based-auto-scaling --region us-east-1 --instance-
ids 701f2ffe-5d8e-4187-b140-77b75f55de8d
```

輸出：此範例具有單一以時間為基礎的執行個體。

```
{
  "TimeBasedAutoScalingConfigurations": [
   {
      "InstanceId": "701f2ffe-5d8e-4187-b140-77b75f55de8d",
      "AutoScalingSchedule": {
        "Monday": {
          "11": "on",
          "10": "on",
          "13": "on",
          "12": "on"
        },
        "Tuesday": {
          "11": "on",
          "10": "on",
          "13": "on",
          "12": "on"
        }
      }
    }
  ]
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的自動時間擴展的運作方式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTimebasedAutoScaling。

describe-user-profiles

以下程式碼範例顯示如何使用 describe-user-profiles。

AWS CLI

描述使用者設定檔

下列describe-user-profiles命令說明帳戶的使用者設定檔。

```
aws opsworks --region us-east-1 describe-user-profiles
```

輸出：

```
{
  "UserProfiles": [
    {
      "IamUserArn": "arn:aws:iam::123456789012:user/someuser",
      "SshPublicKey": "ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEAkOuP7i80q3Cko...",
      "AllowSelfManagement": true,
      "Name": "someuser",
      "SshUsername": "someuser"
    },
    {
      "IamUserArn": "arn:aws:iam::123456789012:user/cli-user-test",
      "AllowSelfManagement": true,
      "Name": "cli-user-test",
      "SshUsername": "myusername"
    }
  ]
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的 ManagingOpsWorks 使用者。 AWS OpsWorks

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeUserProfiles。

describe-volumes

以下程式碼範例顯示如何使用 describe-volumes。

AWS CLI

描述堆疊的磁碟區

下列範例說明堆疊的 EBS 磁碟區。

```
aws opsworks --region us-east-1 describe-volumes --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

輸出：

```
{
  "Volumes": [
    {
      "Status": "in-use",
      "AvailabilityZone": "us-west-2a",
      "Name": "CLITest",
      "InstanceId": "dfe18b02-5327-493d-91a4-c5c0c448927f",
      "VolumeType": "standard",
      "VolumeId": "56b66fbd-e1a1-4aff-9227-70f77118d4c5",
      "Device": "/dev/sdi",
      "Ec2VolumeId": "vol-295c1638",
      "MountPoint": "/mnt/myvolume",
      "Size": 1
    }
  ]
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的資源管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeVolumes。

detach-elastic-load-balancer

以下程式碼範例顯示如何使用 detach-elastic-load-balancer。

AWS CLI

從其層分離負載平衡器

下列範例會將以其名稱識別的負載平衡器從其 layer 分離。

```
aws opsworks --region us-east-1 detach-elastic-load-balancer --elastic-load-balancer-name Java-LB --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的 Elastic Load Balancing。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetachElasticLoadBalancer。

`disassociate-elastic-ip`

以下程式碼範例顯示如何使用 `disassociate-elastic-ip`。

AWS CLI

取消彈性 IP 地址與執行個體的關聯

下列範例會取消彈性 IP 地址與指定執行個體的關聯。

```
aws opsworks --region us-east-1 disassociate-elastic-ip --elastic-ip 54.148.130.96
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的資源管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateElasticIp。

`get-hostname-suggestion`

以下程式碼範例顯示如何使用 `get-hostname-suggestion`。

AWS CLI

取得 layer 的下一個主機名稱

下列範例會取得指定 layer 的下一個產生的主機名稱。此範例使用的 layer 是具有一個執行個體的 Java Application Server layer。堆疊的主機名稱主題是預設值 Layer_Dependent。

```
aws opsworks --region us-east-1 get-hostname-suggestion --layer-
id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

輸出：

```
{
  "Hostname": "java-app2",
  "LayerId": "888c5645-09a5-4d0e-95a8-812ef1db76a4"
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的建立新堆疊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetHostnameSuggestion。

reboot-instance

以下程式碼範例顯示如何使用 reboot-instance。

AWS CLI

重新啟動執行個體

下列範例會重新啟動執行個體。

```
aws opsworks --region us-east-1 reboot-instance --instance-
id dfe18b02-5327-493d-91a4-c5c0c448927f
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的重新啟動執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RebootInstance。

register-elastic-ip

以下程式碼範例顯示如何使用 register-elastic-ip。

AWS CLI

向堆疊註冊彈性 IP 地址

下列範例會使用指定的堆疊，註冊以其 IP 地址識別的彈性 IP 地址。

注意：彈性 IP 地址必須與堆疊位於相同的區域。

```
aws opsworks register-elastic-ip --region us-east-1 --stack-
id d72553d4-8727-448c-9b00-f024f0ba1b06 --elastic-ip 54.148.130.96
```

輸出

```
{
  "ElasticIp": "54.148.130.96"
}
```

詳細資訊

如需詳細資訊，請參閱 OpsWorks 使用者指南中的向堆疊註冊彈性 IP 地址。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterElasticIp。

register-rds-db-instance

以下程式碼範例顯示如何使用 register-rds-db-instance。

AWS CLI

向堆疊註冊 Amazon RDS 執行個體

下列範例會使用指定的堆疊註冊由其 Amazon Resource Name (ARN) 識別的 Amazon RDS 資料庫執行個體。它也會指定執行個體的主使用者名稱和密碼。請注意， AWS OpsWorks 不會驗證其中一個值。如果其中一個不正確，您的應用程式將無法連線到資料庫。

```
aws opsworks register-rds-db-instance --region us-east-1 --stack-
id d72553d4-8727-448c-9b00-f024f0ba1b06 --rds-db-instance-arn arn:aws:rds:us-
west-2:123456789012:db:clitestdb  --db-user cliuser --db-password some23!pwd
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的使用堆疊註冊 Amazon RDS 執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterRdsDbInstance。

register-volume

以下程式碼範例顯示如何使用 register-volume。

AWS CLI

向堆疊註冊 Amazon EBS 磁碟區

下列範例會使用指定的堆疊，註冊以其磁碟區 ID 識別的 Amazon EBS 磁碟區。

```
aws opsworks register-volume --region us-east-1 --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06 --ec-2-volume-id vol-295c1638
```

輸出：

```
{
  "VolumeId": "ee08039c-7cb7-469f-be10-40fb7f0c05e8"
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的使用堆疊註冊 Amazon EBS 磁碟區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterVolume。

register

以下程式碼範例顯示如何使用 register。

AWS CLI

向堆疊註冊執行個體

下列範例顯示向 AWS Opsworks 外部建立的堆疊註冊執行個體的各種方式。您可以從要註冊register的執行個體執行 ，或從單獨的工作站執行 。如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的註冊 Amazon EC2 和內部部署執行個體。

注意：為了簡潔起見，範例會省略 region引數。

註冊 Amazon EC2 執行個體

若要表示您正在註冊 EC2 執行個體，請將 `--infrastructure-class`引數設定為 `ec2`。

下列範例會從個別工作站向指定的堆疊註冊 EC2 執行個體。執行個體會以其 EC2 ID 識別`i-12345678`。此範例使用工作站的預設 SSH 使用者名稱，並嘗試使用不需要密碼的身分驗證技術登入執行個體，例如預設私有 SSH 金鑰。如果失敗， 會`register`查詢密碼。

```
aws opsworks register --infrastructure-class=ec2 --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb i-12345678
```

下列範例會從個別工作站向指定的堆疊註冊 EC2 執行個體。它使用 `--ssh-username`和 `--ssh-private-key`引數來明確指定命令用來登入執行個體的 SSH 使用者名稱和私有金鑰檔案。 `ec2-user`是 Amazon Linux 執行個體的標準使用者名稱。將 `ubuntu`用於 Ubuntu 執行個體。

```
aws opsworks register --infrastructure-class=ec2 --stack-
id 935450cc-61e0-4b03-a3e0-160ac817d2bb --ssh-username ec2-user --ssh-private-
key ssh_private_key i-12345678
```

下列範例會註冊執行 `register`命令的 EC2 執行個體。使用 SSH 登入執行個體，並使用 `register --local`引數執行，而非執行個體 ID 或主機名稱。

```
aws opsworks register --infrastructure-class ec2 --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --local
```

註冊現場部署執行個體

若要表示您正在註冊現場部署執行個體，請將 `--infrastructure-class`引數設定為 `on-premises`。

下列範例會從個別工作站使用指定的堆疊註冊現有的現場部署執行個體。執行個體是透過其 IP 地址來識別`192.0.2.3`。此範例使用工作站的預設 SSH 使用者名稱，並嘗試使用不需要密碼的身分驗證技術登入執行個體，例如預設私有 SSH 金鑰。如果失敗， 會`register`查詢密碼。

```
aws opsworks register --infrastructure-class on-premises --stack-
id 935450cc-61e0-4b03-a3e0-160ac817d2bb 192.0.2.3
```

下列範例會向個別工作站的指定堆疊註冊現場部署執行個體。執行個體的識別方式是其主機名稱 `host1`。`--override-...` 引數 direct AWS OpsWorks 會分別顯示`webserver1`為主機名稱`192.0.2.3`和`10.0.0.2`執行個體的公有和私有 IP 地址。

```
aws opsworks register --infrastructure-class on-premises --stack-
id 935450cc-61e0-4b03-a3e0-160ac817d2bb --override-hostname webserver1 --override-
public-ip 192.0.2.3 --override-private-ip 10.0.0.2 host1
```

下列範例會向個別工作站的指定堆疊註冊現場部署執行個體。執行個體由其 IP 地址識別。 會使用指定的 SSH 使用者名稱和私有金鑰檔案register登入執行個體。

```
aws opsworks register --infrastructure-class on-premises --stack-
id 935450cc-61e0-4b03-a3e0-160ac817d2bb --ssh-username admin --ssh-private-
key ssh_private_key 192.0.2.3
```

下列範例會從個別工作站使用指定的堆疊註冊現有的現場部署執行個體。命令會使用指定 SSH 密碼和執行個體 IP 地址的自訂 SSH 命令字串登入執行個體。

```
aws opsworks register --infrastructure-class on-premises --stack-
id 935450cc-61e0-4b03-a3e0-160ac817d2bb --override-ssh "sshpass -p 'mypassword' ssh
 your-user@192.0.2.3"
```

下列範例會註冊執行 register命令的現場部署執行個體。使用 SSH 登入執行個體，並使用 register --local引數執行，而非執行個體 ID 或主機名稱。

```
aws opsworks register --infrastructure-class on-premises --stack-
id 935450cc-61e0-4b03-a3e0-160ac817d2bb --local
```

輸出：下列是註冊 EC2 執行個體的典型輸出。

```
Warning: Permanently added '52.11.41.206' (ECDSA) to the list of known hosts.
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                               Dload  Upload   Total   Spent    Left  Speed
100 6403k  100 6403k    0     0  2121k      0  0:00:03  0:00:03 --:--:-- 2121k
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Initializing AWS OpsWorks
 environment
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Running on Ubuntu
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Checking if OS is supported
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Running on supported OS
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Setup motd
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Executing: ln -sf --backup /etc/
motd.opsworks-static /etc/motd
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Enabling multiverse repositories
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Customizing APT environment
```

```
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Installing system packages
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Executing: dpkg --configure -a
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Executing with retry: apt-get
 update
[Tue, 24 Feb 2015 20:49:13 +0000] opsworks-init: Executing: apt-get install -y ruby
 ruby-dev libicu-dev libssl-dev libxslt-dev libxml2-dev libyaml-dev monit
[Tue, 24 Feb 2015 20:50:13 +0000] opsworks-init: Using assets bucket from
 environment: 'opsworks-instance-assets-us-east-1.s3.amazonaws.com'.
[Tue, 24 Feb 2015 20:50:13 +0000] opsworks-init: Installing Ruby for the agent
[Tue, 24 Feb 2015 20:50:13 +0000] opsworks-init: Executing: /tmp/opsworks-
agent-installer.YgGq8wF3UUre6yDy/opsworks-agent-installer/opsworks-agent/bin/
installer_wrapper.sh -r -R opsworks-instance-assets-us-east-1.s3.amazonaws.com
[Tue, 24 Feb 2015 20:50:44 +0000] opsworks-init: Starting the installer
Instance successfully registered. Instance ID: 4d6d1710-ded9-42a1-b08e-b043ad7af1e2
Connection to 52.11.41.206 closed.
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks AWS OpsWorks Stack 註冊執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的註冊。

`set-load-based-auto-scaling`

以下程式碼範例顯示如何使用 `set-load-based-auto-scaling`。

AWS CLI

設定 layer 的負載型擴展組態

下列範例會啟用指定 layer 的負載型擴展，並設定該 layer 的組態。您必須使用 `create-instance` 將負載型執行個體新增至 layer。

```
aws opsworks --region us-east-1 set-load-based-auto-scaling --layer-
id 523569ae-2faf-47ac-b39e-f4c4b381f36d --enable --up-scaling file://upscale.json --
down-scaling file://downscale.json
```

此範例會將升級閾值設定放在名為 的工作目錄中的個別檔案中`upscale.json`，其中包含下列項目。

```
{
  "InstanceCount": 2,
```

```
  "ThresholdsWaitTime": 3,
  "IgnoreMetricsTime": 3,
  "CpuThreshold": 85,
  "MemoryThreshold": 85,
  "LoadThreshold": 85
}
```

此範例會將縮減閾值設定放在名為 的工作目錄中的個別檔案中downscale.json，其中包含下列項目。

```
{
"InstanceCount": 2,
"ThresholdsWaitTime": 3,
"IgnoreMetricsTime": 3,
"CpuThreshold": 35,
"MemoryThreshold": 30,
"LoadThreshold": 30
}
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的使用自動負載型擴展。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetLoadBasedAutoScaling。

set-permission

以下程式碼範例顯示如何使用 set-permission。

AWS CLI

授予每個stack AWS OpsWorks 許可層級

當您透過呼叫 將 AWS Identity and Access Management (IAM) 使用者匯入至 AWS OpsWorks 時create-user-profile，該使用者只有連接 IAM 政策授予的那些許可。您可以修改使用者的政策來授予 AWS OpsWorks 許可。不過，匯入使用者通常會比較容易，然後使用 set-permission命令，為使用者需要存取的每個堆疊授予使用者其中一個標準許可層級。

下列範例會為 Amazon Resource Name (ARN) 識別的使用者授予指定堆疊的許可。此範例會授予使用者管理許可層級，以及堆疊執行個體上的 sudo 和 SSH 權限。

```
aws opsworks set-permission --region us-east-1 --stack-id 71c7ca72-55ae-4b6a-8ee1-a8dcded3fa0f --level manage  --iam-user-arn arn:aws:iam::123456789102:user/cli-user-test --allow-ssh --allow-sudo
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的 GrantingOpsWorks User Per-Stack Permissions。 AWS OpsWorks

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetPermission。

set-time-based-auto-scaling

以下程式碼範例顯示如何使用 set-time-based-auto-scaling。

AWS CLI

設定 layer 的時間式擴展組態

下列範例會設定指定執行個體的時間型組態。您必須先使用 create-instance 將執行個體新增至 layer。

```
aws opsworks --region us-east-1 set-time-based-auto-scaling --instance-id 69b6237c-08c0-4edb-a6af-78f3d01cedf2 --auto-scaling-schedule file://schedule.json
```

此範例會將排程放在名為 的工作目錄中的個別檔案中schedule.json。在此範例中，執行個體會在週一和週二的中午 UTC （國際標準時間） 左右開啟幾個小時。

```
{
  "Monday": {
    "10": "on",
    "11": "on",
    "12": "on",
    "13": "on"
  },
  "Tuesday": {
    "10": "on",
    "11": "on",
    "12": "on",
```

```
    "13": "on"
  }
}
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的使用自動時間擴展。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetTimeBasedAutoScaling。

start-instance

以下程式碼範例顯示如何使用 start-instance。

AWS CLI

啟動執行個體

下列start-instance命令會啟動指定的全年無休執行個體。

```
aws opsworks start-instance --instance-id f705ee48-9000-4890-8bd3-20eb05825aaf
```

輸出：無。使用 describe-instances 檢查執行個體的狀態。

秘訣 您可以呼叫 start-stack，以一個命令啟動堆疊中的每個離線執行個體。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的手動啟動、停止和重新啟動全年無休的執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartInstance。

start-stack

以下程式碼範例顯示如何使用 start-stack。

AWS CLI

啟動堆疊的執行個體

下列範例會啟動堆疊的所有全年無休執行個體。若要啟動特定執行個體，請使用 start-instance。

```
aws opsworks --region us-east-1 start-stack --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的啟動執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartStack。

stop-instance

以下程式碼範例顯示如何使用 stop-instance。

AWS CLI

停止執行個體

下列範例會停止指定的執行個體，以其執行個體 ID 識別。您可以前往 AWS OpsWorks 主控台上的執行個體詳細資訊頁面，或執行 describe-instances命令來取得執行個體 ID。

```
aws opsworks stop-instance --region us-east-1 --instance-id 3a21cfac-4a1f-4ce2-a921-b2cfba6f7771
```

您可以呼叫 來重新啟動已停止的執行個體，start-instance或呼叫 來刪除執行個體delete-instance。

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的停止執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopInstance。

stop-stack

以下程式碼範例顯示如何使用 stop-stack。

AWS CLI

停止堆疊的執行個體

下列範例會停止堆疊的所有全年無休執行個體。若要停止特定執行個體，請使用 stop-instance。

```
aws opsworks --region us-east-1 stop-stack --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

輸出：無輸出。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的停止執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopStack。

unassign-instance

以下程式碼範例顯示如何使用 unassign-instance。

AWS CLI

從其 layer 取消指派已註冊的執行個體

下列unassign-instance命令會從其連接的層取消指派執行個體。

```
aws opsworks --region us-east-1 unassign-instance --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的取消指派已註冊的執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UnassignInstance。

unassign-volume

以下程式碼範例顯示如何使用 unassign-volume。

AWS CLI

從其執行個體取消指派磁碟區

下列範例會從其執行個體取消指派已註冊的 Amazon Elastic Block Store (Amazon EBS) 磁碟區。磁碟區是以磁碟區 ID 識別，這是當您向堆疊註冊磁碟區時， AWS OpsWorks 指派的 GUID，而不是 Amazon Elastic Compute Cloud (Amazon EC2) 磁碟區 ID。

```
aws opsworks --region us-east-1 unassign-volume --volume-id 8430177d-52b7-4948-9c62-e195af4703df
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的取消指派 Amazon EBS 磁碟區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UnassignVolume。

update-app

以下程式碼範例顯示如何使用 update-app。

AWS CLI

更新應用程式

下列範例會更新指定的應用程式以變更其名稱。

```
aws opsworks --region us-east-1 update-app --app-id 26a61ead-d201-47e3-b55c-2a7c666942f8 --name NewAppName
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的編輯應用程式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateApp。

update-elastic-ip

以下程式碼範例顯示如何使用 update-elastic-ip。

AWS CLI

更新彈性 IP 地址名稱

下列範例會更新指定彈性 IP 地址的名稱。

```
aws opsworks --region us-east-1 update-elastic-ip --elastic-ip 54.148.130.96 --
name NewIPName
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的資源管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateElasticIp。

update-instance

以下程式碼範例顯示如何使用 update-instance。

AWS CLI

更新執行個體

下列範例會更新指定執行個體的類型。

```
aws opsworks --region us-east-1 update-instance --instance-
id dfe18b02-5327-493d-91a4-c5c0c448927f --instance-type c3.xlarge
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的編輯執行個體組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateInstance。

update-layer

以下程式碼範例顯示如何使用 update-layer。

AWS CLI

更新圖層

下列範例會更新指定的 layer 以使用 Amazon EBS 最佳化執行個體。

```
aws opsworks --region us-east-1 update-layer --layer-
id 888c5645-09a5-4d0e-95a8-812ef1db76a4 --use-ebs-optimized-instances
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 OpsWorks AWS 使用者指南中的編輯 OpsWorks Layer 的組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateLayer。

update-my-user-profile

以下程式碼範例顯示如何使用 update-my-user-profile。

AWS CLI

更新使用者的設定檔

下列範例會將development使用者的設定檔更新為使用指定的 SSH 公有金鑰。使用者的 AWS 登入資料由 credentials 檔案 (~\.aws\credentials) 中的development設定檔表示，而金鑰位於工作目錄中的.pem檔案中。

```
aws opsworks --region us-east-1 --profile development update-my-user-profile --ssh-
public-key file://development_key.pem
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的 EditingOpsWorks 使用者設定。 AWS OpsWorks

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateMyUserProfile。

update-rds-db-instance

以下程式碼範例顯示如何使用 update-rds-db-instance。

AWS CLI

更新已註冊的 Amazon RDS 資料庫執行個體

下列範例會更新 Amazon RDS 執行個體的主密碼值。請注意，此命令不會變更 RDS 執行個體的主密碼，只會變更您提供給 to AWS OpsWorks 的密碼。如果此密碼不符合 RDS 執行個體的密碼，您的應用程式將無法連線至資料庫。

```
aws opsworks --region us-east-1 update-rds-db-instance --db-password 123456789
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的使用堆疊註冊 Amazon RDS 執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRdsDbInstance。

update-volume

以下程式碼範例顯示如何使用 update-volume。

AWS CLI

更新已註冊的磁碟區

下列範例會更新已註冊的 Amazon Elastic Block Store (Amazon EBS) 磁碟區的掛載點。磁碟區是以磁碟區 ID 識別，這是當您向堆疊註冊磁碟區時， AWS OpsWorks 指派給磁碟區的 GUID，而不是 Amazon Elastic Compute Cloud (Amazon EC2) 磁碟區 ID：

```
aws opsworks --region us-east-1 update-volume --volume-id 8430177d-52b7-4948-9c62-
e195af4703df --mount-point /mnt/myvol
```

輸出：無。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的將 Amazon EBS 磁碟區指派給執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateVolume。

AWS OpsWorks CM 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS OpsWorks CM。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-node`

以下程式碼範例顯示如何使用 `associate-node`。

AWS CLI

建立節點的關聯

下列`associate-node`命令會將名為 的節點`i-44de882p`與名為 的 Chef Automate 伺服器建立關聯`automate-06`，這表示`automate-06`伺服器管理節點，並透過由 associate-node 命令安裝在節點上的`chef-client`代理程式軟體，將配方命令傳達給節點。有效的節點名稱為 EC2 IDs.:

```
aws opsworks-cm associate-node --server-name "automate-06" --node-name "i-43de882p"
 --engine-attributes "Name=CHEF_ORGANIZATION,Value='MyOrganization'
 Name=CHEF_NODE_PUBLIC_KEY,Value='Public_key_contents'"
```

命令傳回的輸出如下所示。輸出：

```
{
 "NodeAssociationStatusToken": "AHUY8wFe4pdXtZC5DiJa5SOLp5o14DH//
rHRqHDWXxwVoNBxcEy4V7R0NOFymh7E/1HumOBPsemPQFE6dcGaiFk"
```

```
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks AWS OpsWorks for Chef Automate。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateNode。

create-backup

以下程式碼範例顯示如何使用 create-backup。

AWS CLI

建立備份

下列create-backup命令會啟動automate-06us-east-1區域中名為 之 Chef Automate 伺服器的手動備份。命令會將描述性訊息新增至 --description 參數中的備份。

```
aws opsworks-cm create-backup \
    --server-name 'automate-06' \
    --description "state of my infrastructure at launch"
```

輸出會顯示與下列有關新備份的資訊。

輸出：

```
{
    "Backups": [
        {
            "BackupArn": "string",
            "BackupId": "automate-06-20160729133847520",
            "BackupType": "MANUAL",
            "CreatedAt": 2016-07-29T13:38:47.520Z,
            "Description": "state of my infrastructure at launch",
            "Engine": "Chef",
            "EngineModel": "Single",
            "EngineVersion": "12",
            "InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/
automate-06-1010V4UU2WRM2",
            "InstanceType": "m4.large",
            "KeyPair": "",
```

```
            "PreferredBackupWindow": "",
            "PreferredMaintenanceWindow": "",
            "S3LogUrl": "https://s3.amazonaws.com/<bucket-name>/
automate-06-20160729133847520",
            "SecurityGroupIds": [ "sg-1a24c270" ],
            "ServerName": "automate-06",
            "ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-
service-role.1114810729735",
            "Status": "OK",
            "StatusDescription": "",
            "SubnetIds": [ "subnet-49436a18" ],
            "ToolsVersion": "string",
            "UserArn": "arn:aws:iam::1019881987024:user/opsworks-user"
        }
    ],
}
```

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的備份和還原 Chef Automate Server 的 anOpsWorks。 AWS OpsWorks

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateBackup。

create-server

以下程式碼範例顯示如何使用 create-server。

AWS CLI

建立伺服器

下列create-server範例automate-06會在您的預設區域中建立名為 的新 Chef Automate 伺服器。請注意，預設值用於大多數其他設定，例如要保留的備份數量，以及維護和備份開始時間。執行create-server命令之前，請先完成《AWS OpsWorks for Chef Automate 使用者指南》中的 OpsWorks AWS for Chef Automate 入門中的先決條件。

```
aws opsworks-cm create-server \
    --engine "Chef" \
    --engine-model "Single" \
    --engine-version "12" \
    --server-name "automate-06" \
    --instance-profile-arn "arn:aws:iam::1019881987024:instance-profile/aws-
opsworks-cm-ec2-role" \
```

```
    --instance-type "t2.medium" \
    --key-pair "amazon-test" \
    --service-role-arn "arn:aws:iam::044726508045:role/aws-opsworks-cm-service-role"
```

輸出會顯示與新伺服器類似的資訊：

```
{
    "Server": {
        "BackupRetentionCount": 10,
        "CreatedAt": 2016-07-29T13:38:47.520Z,
        "DisableAutomatedBackup": FALSE,
        "Endpoint": "https://opsworks-cm.us-east-1.amazonaws.com",
        "Engine": "Chef",
        "EngineAttributes": [
            {
                "Name": "CHEF_DELIVERY_ADMIN_PASSWORD",
                "Value": "1Password1"
            }
        ],
        "EngineModel": "Single",
        "EngineVersion": "12",
        "InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/aws-
opsworks-cm-ec2-role",
        "InstanceType": "t2.medium",
        "KeyPair": "amazon-test",
        "MaintenanceStatus": "",
        "PreferredBackupWindow": "Sun:02:00",
        "PreferredMaintenanceWindow": "00:00",
        "SecurityGroupIds": [ "sg-1a24c270" ],
        "ServerArn": "arn:aws:iam::1019881987024:instance/
automate-06-1010V4UU2WRM2",
        "ServerName": "automate-06",
        "ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-service-
role",
        "Status": "CREATING",
        "StatusReason": "",
        "SubnetIds": [ "subnet-49436a18" ]
    }
}
```

如需詳細資訊，請參閱 AWS OpsWorks for Chef Automate API 參考中的 UpdateServer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateServer。

delete-backup

以下程式碼範例顯示如何使用 delete-backup。

AWS CLI

刪除備份

下列delete-backup命令會刪除以備份 ID 識別的 Chef Automate 伺服器的手動或自動備份。當您接近可儲存的備份數量上限，或想要將 Amazon S3 儲存成本降至最低時，此命令很有用：

```
aws opsworks-cm delete-backup --backup-id "automate-06-2016-11-19T23:42:40.240Z"
```

輸出會顯示備份刪除是否成功。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的備份和還原 Chef Automate Server 的 anOpsWorks。 AWS OpsWorks

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBackup。

delete-server

以下程式碼範例顯示如何使用 delete-server。

AWS CLI

刪除伺服器

下列delete-server命令會刪除由伺服器名稱識別的 Chef Automate 伺服器。刪除伺服器後，請求不會再傳回該伺服器DescribeServer。：

```
aws opsworks-cm delete-server --server-name "automate-06"
```

輸出會顯示伺服器刪除是否成功。

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks AWS OpsWorks for Chef Automate Server。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteServer。

`describe-account-attributes`

以下程式碼範例顯示如何使用 `describe-account-attributes`。

AWS CLI

描述帳戶屬性

下列`describe-account-attributes`命令會傳回您帳戶使用 AWS OpsWorks for Chef Automate 資源的相關資訊：

```
aws opsworks-cm describe-account-attributes
```

命令傳回的每個帳戶屬性項目的輸出如下所示。輸出：

```
{
 "Attributes": [
    {
       "Maximum": 5,
       "Name": "ServerLimit",
       "Used": 2
    }
 ]
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks for Chef Automate API 參考中的 DescribeAccountAttributes。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAccountAttributes。

`describe-backups`

以下程式碼範例顯示如何使用 `describe-backups`。

AWS CLI

描述備份

下列describe-backups命令會傳回與您預設區域中帳戶相關聯的所有備份的相關資訊。

```
aws opsworks-cm describe-backups
```

命令傳回的每個備份項目的輸出如下所示。

輸出：

```
{
    "Backups": [
        {
            "BackupArn": "string",
            "BackupId": "automate-06-20160729133847520",
            "BackupType": "MANUAL",
            "CreatedAt": 2016-07-29T13:38:47.520Z,
            "Description": "state of my infrastructure at launch",
            "Engine": "Chef",
            "EngineModel": "Single",
            "EngineVersion": "12",
            "InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/
automate-06-1010V4UU2WRM2",
            "InstanceType": "m4.large",
            "KeyPair": "",
            "PreferredBackupWindow": "",
            "PreferredMaintenanceWindow": "",
            "S3LogUrl": "https://s3.amazonaws.com/<bucket-name>/
automate-06-20160729133847520",
            "SecurityGroupIds": [ "sg-1a24c270" ],
            "ServerName": "automate-06",
            "ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-
service-role.1114810729735",
            "Status": "Successful",
            "StatusDescription": "",
            "SubnetIds": [ "subnet-49436a18" ],
            "ToolsVersion": "string",
            "UserArn": "arn:aws:iam::1019881987024:user/opsworks-user"
        }
    ],
}
```

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的備份和還原 Chef Automate Server 的 anOpsWorks。 AWS OpsWorks

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeBackups。

describe-events

以下程式碼範例顯示如何使用 describe-events。

AWS CLI

描述事件

下列describe-events範例會傳回與指定 Chef Automate 伺服器相關聯的所有事件的相關資訊。

```
aws opsworks-cm describe-events \
    --server-name 'automate-06'
```

命令傳回的每個事件項目的輸出類似下列範例：

```
{
    "ServerEvents": [
        {
            "CreatedAt": 2016-07-29T13:38:47.520Z,
            "LogUrl": "https://s3.amazonaws.com/<bucket-name>/
automate-06-20160729133847520",
            "Message": "Updates successfully installed.",
            "ServerName": "automate-06"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的一般故障診斷秘訣。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEvents。

describe-node-association-status

以下程式碼範例顯示如何使用 describe-node-association-status。

AWS CLI

描述節點關聯狀態

下列describe-node-association-status命令會傳回請求的狀態，以將節點與名為 的 Chef Automate 伺服器建立關聯automate-06：

```
aws opsworks-cm describe-node-association-status --server-
name "automate-06" --node-association-status-token "AflJKl+/
GoKLZJBdDQEx0O65CDi57blQe9nKM8joSok0pQ9xr8DqApBN9/1O6sLdSvlfDEKkEx+eoCHvjoWHaOs="
```

命令傳回的每個帳戶屬性項目的輸出如下所示。輸出：

```
{
 "NodeAssociationStatus": "IN_PROGRESS"
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks for Chef Automate API 參考中的 DescribeNodeAssociationStatus。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeNodeAssociationStatus。

describe-servers

以下程式碼範例顯示如何使用 describe-servers。

AWS CLI

描述伺服器

下列describe-servers命令會傳回與您的帳戶相關聯的所有伺服器，以及您預設區域中的所有伺服器的相關資訊：

```
aws opsworks-cm describe-servers
```

命令傳回的每個伺服器項目的輸出如下所示。輸出：

```
{
 "Servers": [
    {
       "BackupRetentionCount": 8,
       "CreatedAt": 2016-07-29T13:38:47.520Z,
```

```
        "DisableAutomatedBackup": FALSE,
        "Endpoint": "https://opsworks-cm.us-east-1.amazonaws.com",
        "Engine": "Chef",
        "EngineAttributes": [
           {
              "Name": "CHEF_DELIVERY_ADMIN_PASSWORD",
              "Value": "1Password1"
           }
        ],
        "EngineModel": "Single",
        "EngineVersion": "12",
        "InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/
automate-06-1010V4UU2WRM2",
        "InstanceType": "m4.large",
        "KeyPair": "",
        "MaintenanceStatus": "SUCCESS",
        "PreferredBackupWindow": "03:00",
        "PreferredMaintenanceWindow": "Mon:09:00",
        "SecurityGroupIds": [ "sg-1a24c270" ],
        "ServerArn": "arn:aws:iam::1019881987024:instance/automate-06-1010V4UU2WRM2",
        "ServerName": "automate-06",
        "ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-service-
role.1114810729735",
        "Status": "HEALTHY",
        "StatusReason": "",
        "SubnetIds": [ "subnet-49436a18" ]
    }
 ]
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks for Chef Automate API 指南中的 DescribeServers。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeServers。

disassociate-node

以下程式碼範例顯示如何使用 disassociate-node。

AWS CLI

取消節點的關聯

下列disassociate-node命令會取消與名為 的節點的關聯i-44de882p，由名為 的 Chef Automate 伺服器從管理中移除節點automate-06。有效的節點名稱為 EC2 IDs.:

```
aws opsworks-cm disassociate-node --server-name "automate-06" --node-
name "i-43de882p" --engine-attributes "Name=CHEF_ORGANIZATION,Value='MyOrganization'
 Name=CHEF_NODE_PUBLIC_KEY,Value='Public_key_contents'"
```

命令傳回的輸出如下所示。輸出：

```
{
 "NodeAssociationStatusToken": "AHUY8wFe4pdXtZC5DiJa5SOLp5o14DH//
rHRqHDWXxwVoNBxcEy4V7R0NOFymh7E/1HumOBPsemPQFE6dcGaiFk"
}
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks AWS OpsWorks for Chef Automate Server。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateNode。

restore-server

以下程式碼範例顯示如何使用 restore-server。

AWS CLI

還原伺服器

下列restore-server命令會從 ID 為 的備份執行名為 automate-06 的 Chef Automate 伺服器就地還原automate-06-2016-11-22T16:13:27.998Z。還原伺服器會還原對 Chef Automate 伺服器在執行指定備份時所管理節點的連線。

aws opsworks-cm restore-server --backup-id "automate-06-2016-11-22T16：13：27.998Z" --server-name "automate-06"

輸出僅為命令 ID。輸出：

```
(None)
```

詳細資訊

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的為 Chef Automate Server 還原 FailedOpsWorks。 AWS OpsWorks

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreServer。

`start-maintenance`

以下程式碼範例顯示如何使用 `start-maintenance`。

AWS CLI

開始維護

下列`start-maintenance`範例會在預設區域中指定的 Chef Automate 或 Puppet Enterprise 伺服器上手動開始維護。如果較早的自動維護嘗試失敗，且維護失敗的根本原因已解決，則此命令很有用。

```
aws opsworks-cm start-maintenance \
    --server-name 'automate-06'
```

輸出：

```
{
    "Server": {
        "AssociatePublicIpAddress": true,
        "BackupRetentionCount": 10,
        "ServerName": "automate-06",
        "CreatedAt": 1569229584.842,
        "CloudFormationStackArn": "arn:aws:cloudformation:us-
west-2:123456789012:stack/aws-opsworks-cm-instance-automate-06-1606611794746/
EXAMPLE0-31de-11eb-bdb0-0a5b0a1353b8",
        "DisableAutomatedBackup": false,
        "Endpoint": "automate-06-EXAMPLEvr8gjfk5f.us-west-2.opsworks-cm.io",
        "Engine": "ChefAutomate",
        "EngineModel": "Single",
        "EngineAttributes": [],
        "EngineVersion": "2020-07",
        "InstanceProfileArn": "arn:aws:iam::123456789012:instance-profile/aws-
opsworks-cm-ec2-role",
        "InstanceType": "m5.large",
        "PreferredMaintenanceWindow": "Sun:01:00",
        "PreferredBackupWindow": "Sun:15:00",
        "SecurityGroupIds": [
```

```
            "sg-EXAMPLE"
        ],
        "ServiceRoleArn": "arn:aws:iam::123456789012:role/service-role/aws-opsworks-
cm-service-role",
        "Status": "UNDER_MAINTENANCE",
        "SubnetIds": [
            "subnet-EXAMPLE"
        ],
        "ServerArn": "arn:aws:opsworks-cm:us-west-2:123456789012:server/
automate-06/0148382d-66b0-4196-8274-d1a2b6dff8d1"
    }
}
```

如需詳細資訊，請參閱 AWS OpsWorks 使用者指南中的系統維護 (Puppet Enterprise 伺服器) 或系統維護 (Chef Automate 伺服器)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartMaintenance。

update-server-engine-attributes

以下程式碼範例顯示如何使用 update-server-engine-attributes。

AWS CLI

更新伺服器引擎屬性

下列update-server-engine-attributes命令會更新名為 之 Chef Automate 伺服器的CHEF_PIVOTAL_KEY引擎屬性值automate-06。目前無法變更其他引擎屬性的值。

```
aws opsworks-cm update-server-engine-attributes \
    --attribute-name CHEF_PIVOTAL_KEY \
    --attribute-value "new key value" \
    --server-name "automate-06"
```

輸出會顯示與下列有關已更新伺服器的資訊。

```
{
    "Server": {
        "BackupRetentionCount": 2,
        "CreatedAt": 2016-07-29T13:38:47.520Z,
        "DisableAutomatedBackup": FALSE,
        "Endpoint": "https://opsworks-cm.us-east-1.amazonaws.com",
        "Engine": "Chef",
```

```
        "EngineAttributes": [
            {
                "Name": "CHEF_PIVOTAL_KEY",
                "Value": "new key value"
            }
        ],
        "EngineModel": "Single",
        "EngineVersion": "12",
        "InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/
automate-06-1010V4UU2WRM2",
        "InstanceType": "m4.large",
        "KeyPair": "",
        "MaintenanceStatus": "SUCCESS",
        "PreferredBackupWindow": "Mon:09:15",
        "PreferredMaintenanceWindow": "03:00",
        "SecurityGroupIds": [ "sg-1a24c270" ],
        "ServerArn": "arn:aws:iam::1019881987024:instance/
automate-06-1010V4UU2WRM2",
        "ServerName": "automate-06",
        "ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-service-
role.1114810729735",
        "Status": "HEALTHY",
        "StatusReason": "",
        "SubnetIds": [ "subnet-49436a18" ]
    }
}
```

如需詳細資訊，請參閱 AWS OpsWorks for Chef Automate API 參考中的 UpdateServerEngineAttributes。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateServerEngineAttributes。

update-server

以下程式碼範例顯示如何使用 update-server。

AWS CLI

更新伺服器

下列update-server命令會更新預設區域中指定 Chef Automate 伺服器的維護開始時間。新增 --preferred-maintenance-window 參數，將伺服器維護的開始日期和時間變更為週一上午 9：15。UTC：

```
aws opsworks-cm update-server \
    --server-name "automate-06" \
    --preferred-maintenance-window "Mon:09:15"
```

輸出會顯示與下列有關已更新伺服器的資訊。

```
{
    "Server": {
        "BackupRetentionCount": 8,
        "CreatedAt": 2016-07-29T13:38:47.520Z,
        "DisableAutomatedBackup": TRUE,
        "Endpoint": "https://opsworks-cm.us-east-1.amazonaws.com",
        "Engine": "Chef",
        "EngineAttributes": [
            {
                "Name": "CHEF_DELIVERY_ADMIN_PASSWORD",
                "Value": "1Password1"
            }
        ],
        "EngineModel": "Single",
        "EngineVersion": "12",
        "InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/
automate-06-1010V4UU2WRM2",
        "InstanceType": "m4.large",
        "KeyPair": "",
        "MaintenanceStatus": "OK",
        "PreferredBackupWindow": "Mon:09:15",
        "PreferredMaintenanceWindow": "03:00",
        "SecurityGroupIds": [ "sg-1a24c270" ],
        "ServerArn": "arn:aws:iam::1019881987024:instance/
automate-06-1010V4UU2WRM2",
        "ServerName": "automate-06",
        "ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-service-
role.1114810729735",
        "Status": "HEALTHY",
        "StatusReason": "",
        "SubnetIds": [ "subnet-49436a18" ]
    }
}
```

如需詳細資訊，請參閱 AWS OpsWorks for Chef Automate API 參考中的 UpdateServer。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateServer。

使用 的組織範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Organizations 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`accept-handshake`

以下程式碼範例顯示如何使用 `accept-handshake`。

AWS CLI

接受來自另一個帳戶的交握

組織擁有者 Bill 先前已邀請 Juan 的帳戶加入他的組織。下列範例顯示 Juan 的帳戶接受交握，因此同意邀請。

```
aws organizations accept-handshake --handshake-id h-examplehandshakeid111
```

輸出會顯示下列項目：

```
{
        "Handshake": {
                "Action": "INVITE",
                "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
                "RequestedTimestamp": 1481656459.257,
                "ExpirationTimestamp": 1482952459.257,
                "Id": "h-examplehandshakeid111",
                "Parties": [
                        {
                                "Id": "o-exampleorgid",
                                "Type": "ORGANIZATION"
```

```
                        },
                        {
                                "Id": "juan@example.com",
                                "Type": "EMAIL"
                        }
                ],
                "Resources": [
                        {
                                "Resources": [
                                        {
                                                "Type": "MASTER_EMAIL",
                                                "Value": "bill@amazon.com"
                                        },
                                        {
                                                "Type": "MASTER_NAME",
                                                "Value": "Org Master Account"
                                        },
                                        {
                                                "Type": "ORGANIZATION_FEATURE_SET",
                                                "Value": "ALL"
                                        }
                                ],
                                "Type": "ORGANIZATION",
                                "Value": "o-exampleorgid"
                        },
                        {
                                "Type": "EMAIL",
                                "Value": "juan@example.com"
                        }
                ],
                "State": "ACCEPTED"
        }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptHandshake。

attach-policy

以下程式碼範例顯示如何使用 attach-policy。

AWS CLI

將政策連接至根帳戶、OU 或帳戶

範例 1

下列範例示範如何將服務控制政策 (SCP) 連接至 OU：

```
aws organizations attach-policy
                --policy-id p-examplepolicyid111
                --target-id ou-examplerootid111-exampleouid111
```

範例 2

下列範例示範如何將服務控制政策直接連接至 帳戶：

```
aws organizations attach-policy
                --policy-id p-examplepolicyid111
                --target-id 333333333333
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AttachPolicy。

cancel-handshake

以下程式碼範例顯示如何使用 cancel-handshake。

AWS CLI

若要取消從另一個帳戶傳送的交握

Bill 先前傳送邀請給 Susan 的帳戶加入他的組織。他改變主意，決定在 Susan 接受邀請之前取消邀請。下列範例顯示 Bill 的取消：

```
aws organizations cancel-handshake --handshake-id h-examplehandshakeid111
```

輸出包含交握物件，顯示狀態現在為 CANCELED：

```
{
        "Handshake": {
                "Id": "h-examplehandshakeid111",
                "State":"CANCELED",
                "Action": "INVITE",
                "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
                "Parties": [
```

```
                        {
                                "Id": "o-exampleorgid",
                                "Type": "ORGANIZATION"
                        },
                        {
                                "Id": "susan@example.com",
                                "Type": "EMAIL"
                        }
                ],
                "Resources": [
                        {
                                "Type": "ORGANIZATION",
                                "Value": "o-exampleorgid",
                                "Resources": [
                                        {
                                                "Type": "MASTER_EMAIL",
                                                "Value": "bill@example.com"
                                        },
                                        {
                                                "Type": "MASTER_NAME",
                                                "Value": "Master Account"
                                        },
                                        {
                                                "Type": "ORGANIZATION_FEATURE_SET",
                                                "Value": "CONSOLIDATED_BILLING"
                                        }
                                ]
                        },
                        {
                                "Type": "EMAIL",
                                "Value": "anika@example.com"
                        },
                        {
                                "Type": "NOTES",
                                "Value": "This is a request for Susan's account to
 join Bob's organization."
                        }
                ],
                "RequestedTimestamp": 1.47008383521E9,
                "ExpirationTimestamp": 1.47137983521E9
        }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelHandshake。

create-account

以下程式碼範例顯示如何使用 create-account。

AWS CLI

建立自動成為組織一部分的成員帳戶

下列範例示範如何在 組織中建立成員帳戶。成員帳戶是以生產帳戶名稱和 susan@example.com 的電子郵件地址來設定。Organizations 會使用 OrganizationAccountAccessRole 的預設名稱自動建立 IAM 角色，因為未指定 roleName 參數。此外，由於未指定 IamUserAccessToBilling 參數，因此允許具有足夠許可存取帳戶帳單資料的 IAM 使用者或角色的設定會設定為預設值 ALLOW。Organizations 會自動傳送「歡迎 AWS」電子郵件給 Susan：

```
aws organizations create-account --email susan@example.com --account-
name "Production Account"
```

輸出包含一個請求物件，顯示狀態現在為 IN_PROGRESS：

```
{
        "CreateAccountStatus": {
                "State": "IN_PROGRESS",
                "Id": "car-examplecreateaccountrequestid111"
        }
}
```

您稍後可以將 Id 回應值作為 create-account-request-id 參數的值提供給 describe-create-account-status 命令，以查詢請求的目前狀態。

如需詳細資訊，請參閱《 AWS Organizations 使用者指南》中的在您的組織中建立 AWS 帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAccount。

create-organization

以下程式碼範例顯示如何使用 create-organization。

AWS CLI

範例 1：建立新組織

Bill 想要使用帳戶 111111111111 的登入資料來建立組織。下列範例顯示帳戶成為新組織中的主要帳戶。由於他未指定功能集，因此新組織會預設為啟用所有功能，並在根帳戶上啟用服務控制政策。

```
aws organizations create-organization
```

輸出包含組織物件，其中包含新組織的詳細資訊：

```
{
        "Organization": {
                "AvailablePolicyTypes": [
                        {
                                "Status": "ENABLED",
                                "Type": "SERVICE_CONTROL_POLICY"
                        }
                ],
                "MasterAccountId": "111111111111",
                "MasterAccountArn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/111111111111",
                "MasterAccountEmail": "bill@example.com",
                "FeatureSet": "ALL",
                "Id": "o-exampleorgid",
                "Arn": "arn:aws:organizations::111111111111:organization/o-
exampleorgid"
        }
}
```

範例 2：建立僅啟用合併帳單功能的新組織

下列範例會建立僅支援合併帳單功能的組織：

```
aws organizations create-organization --feature-set CONSOLIDATED_BILLING
```

輸出包含組織物件，其中包含新組織的詳細資訊：

```
{
        "Organization": {
                "Arn": "arn:aws:organizations::111111111111:organization/o-
exampleorgid",
                "AvailablePolicyTypes": [],
                "Id": "o-exampleorgid",
```

```
                "MasterAccountArn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/111111111111",
                "MasterAccountEmail": "bill@example.com",
                "MasterAccountId": "111111111111",
                "FeatureSet": "CONSOLIDATED_BILLING"
        }
}
```

如需詳細資訊，請參閱《 AWS Organizations 使用者指南》中的建立組織。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateOrganization。

create-organizational-unit

以下程式碼範例顯示如何使用 create-organizational-unit。

AWS CLI

在根或父 OU 中建立 OU

下列範例示範如何建立名為 AccountingOU 的 OU：

```
aws organizations create-organizational-unit --parent-id r-examplerootid111 --
name AccountingOU
```

輸出包含 organizationalUnit 物件，其中包含新 OU 的詳細資訊：

```
{
        "OrganizationalUnit": {
                "Id": "ou-examplerootid111-exampleouid111",
                "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-
examplerootid111-exampleouid111",
                "Name": "AccountingOU"
        }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateOrganizationalUnit。

create-policy

以下程式碼範例顯示如何使用 create-policy。

AWS CLI

範例 1：使用 JSON 政策的文字來源檔案建立政策

下列範例示範如何建立名為 的服務控制政策 (SCP)AllowAllS3Actions。政策內容取自本機電腦上名為 的檔案policy.json。

```
aws organizations create-policy --content file://policy.json --
name AllowAllS3Actions, --type SERVICE_CONTROL_POLICY --description "Allows
 delegation of all S3 actions"
```

輸出包含政策物件，其中包含新政策的詳細資訊：

```
{
        "Policy": {
                "Content": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Action\":[\"s3:*\"],\"Resource\":[\"*\"]}]}",
                "PolicySummary": {
                        "Arn": "arn:aws:organizations::o-exampleorgid:policy/
service_control_policy/p-examplepolicyid111",
                        "Description": "Allows delegation of all S3 actions",
                        "Name": "AllowAllS3Actions",
                        "Type":"SERVICE_CONTROL_POLICY"
                }
        }
}
```

範例 2：使用 JSON 政策做為參數來建立政策

下列範例示範如何建立相同的 SCP，這次在 參數中將政策內容內嵌為 JSON 字串。字串必須在雙引號之前以反斜線逸出，以確保它們在 參數中被視為常值，其本身被雙引號包圍：

```
aws organizations create-policy --content "{\"Version\":\"2012-10-17\",\"Statement
\":[{\"Effect\":\"Allow\",\"Action\":[\"s3:*\"],\"Resource\":[\"*\"]}]}" --
name AllowAllS3Actions --type SERVICE_CONTROL_POLICY --description "Allows
 delegation of all S3 actions"
```

如需在組織中建立和使用政策的詳細資訊，請參閱《 AWS Organizations 使用者指南》中的管理組織政策。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreatePolicy。

decline-handshake

以下程式碼範例顯示如何使用 decline-handshake。

AWS CLI

拒絕從另一個帳戶傳送的交握

下列範例顯示身為帳戶 222222222222 擁有者的管理員 Susan 拒絕加入 Bill 組織的邀請。DeclineHandshake 操作會傳回交握物件，顯示狀態現在為 DECLINED：

```
aws organizations decline-handshake --handshake-id h-examplehandshakeid111
```

輸出包含顯示 新狀態的交握物件DECLINED：

```
{
        "Handshake": {
                "Id": "h-examplehandshakeid111",
                "State": "DECLINED",
                "Resources": [
                        {
                                "Type": "ORGANIZATION",
                                "Value": "o-exampleorgid",
                                "Resources": [
                                        {
                                                "Type": "MASTER_EMAIL",
                                                "Value": "bill@example.com"
                                        },
                                        {
                                                "Type": "MASTER_NAME",
                                                "Value": "Master Account"
                                        }
                                ]
                        },
                        {
                                "Type": "EMAIL",
                                "Value": "susan@example.com"
                        },
                        {
                                "Type": "NOTES",
                                "Value": "This is an invitation to Susan's account
 to join the Bill's organization."
                        }
```

```
                ],
                "Parties": [
                        {
                                "Type": "EMAIL",
                                "Id": "susan@example.com"
                        },
                        {
                                "Type": "ORGANIZATION",
                                "Id": "o-exampleorgid"
                        }
                ],
                "Action": "INVITE",
                "RequestedTimestamp": 1470684478.687,
                "ExpirationTimestamp": 1471980478.687,
                "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111"
        }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeclineHandshake。

delete-organization

以下程式碼範例顯示如何使用 delete-organization。

AWS CLI

刪除組織

下列範例示範如何刪除組織。若要執行此操作，您必須是組織中主帳戶的管理員。此範例假設您先前已從組織移除所有成員帳戶、OUs 和政策：

```
aws organizations delete-organization
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteOrganization。

delete-organizational-unit

以下程式碼範例顯示如何使用 delete-organizational-unit。

AWS CLI

刪除 OU

下列範例顯示如何刪除 OU。此範例假設您先前已從 OUs 移除所有帳戶和其他 OU：

```
aws organizations delete-organizational-unit --organizational-unit-id ou-examplerootid111-exampleouid111
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteOrganizationalUnit。

delete-policy

以下程式碼範例顯示如何使用 delete-policy。

AWS CLI

刪除政策

下列範例顯示如何從組織刪除政策。此範例假設您先前已從所有實體分離政策：

```
aws organizations delete-policy --policy-id p-examplepolicyid111
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeletePolicy。

describe-account

以下程式碼範例顯示如何使用 describe-account。

AWS CLI

取得帳戶的詳細資訊

下列範例示範如何請求 帳戶的詳細資訊：

```
aws organizations describe-account --account-id 555555555555
```

輸出會顯示帳戶物件，其中包含帳戶的詳細資訊：

```
{
        "Account": {
```

```
                "Id": "555555555555",
                "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/555555555555",
                "Name": "Beta account",
                "Email": "anika@example.com",
                "JoinedMethod": "INVITED",
                "JoinedTimeStamp": 1481756563.134,
                "Status": "ACTIVE"
        }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAccount。

describe-create-account-status

以下程式碼範例顯示如何使用 describe-create-account-status。

AWS CLI

若要取得建立 帳戶之請求的最新狀態

下列範例顯示如何請求先前請求的最新狀態，以在組織中建立 帳戶。指定的 --request-id 來自原始呼叫 create-account 的回應。帳戶建立請求會依狀態欄位顯示 Organizations 已成功完成帳戶建立。

命令：

```
aws organizations describe-create-account-status --create-account-request-id car-
examplecreateaccountrequestid111
```

輸出：

```
{
  "CreateAccountStatus": {
    "State": "SUCCEEDED",
    "AccountId": "555555555555",
    "AccountName": "Beta account",
    "RequestedTimestamp": 1470684478.687,
    "CompletedTimestamp": 1470684532.472,
    "Id": "car-examplecreateaccountrequestid111"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCreateAccountStatus。

describe-handshake

以下程式碼範例顯示如何使用 describe-handshake。

AWS CLI

取得交握的相關資訊

下列範例示範如何請求有關交握的詳細資訊。交握 ID 來自對 的原始呼叫InviteAccountToOrganization，或來自對 ListHandshakesForAccount或 的呼叫ListHandshakesForOrganization：

```
aws organizations describe-handshake --handshake-id h-examplehandshakeid111
```

輸出包含交握物件，其中包含所請求交握的所有詳細資訊：

```
{
        "Handshake": {
                "Id": "h-examplehandshakeid111",
                "State": "OPEN",
                "Resources": [
                        {
                                "Type": "ORGANIZATION",
                                "Value": "o-exampleorgid",
                                "Resources": [
                                        {
                                                "Type": "MASTER_EMAIL",
                                                "Value": "bill@example.com"
                                        },
                                        {
                                                "Type": "MASTER_NAME",
                                                "Value": "Master Account"
                                        }
                                ]
                        },
                        {
                                "Type": "EMAIL",
                                "Value": "anika@example.com"
                        }
                ],
```

```
                "Parties": [
                        {
                                "Type": "ORGANIZATION",
                                "Id": "o-exampleorgid"
                        },
                        {
                                "Type": "EMAIL",
                                "Id": "anika@example.com"
                        }
                ],
                "Action": "INVITE",
                "RequestedTimestamp": 1470158698.046,
                "ExpirationTimestamp": 1471454698.046,
                "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111"
        }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeHandshake。

describe-organization

以下程式碼範例顯示如何使用 describe-organization。

AWS CLI

取得目前組織的相關資訊

下列範例說明如何請求有關組織的詳細資訊：

```
aws organizations describe-organization
```

輸出包含具有組織詳細資訊的組織物件：

```
{
        "Organization": {
                "MasterAccountArn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/111111111111",
                "MasterAccountEmail": "bill@example.com",
                "MasterAccountId": "111111111111",
                "Id": "o-exampleorgid",
                "FeatureSet": "ALL",
```

```
                "Arn": "arn:aws:organizations::111111111111:organization/o-
exampleorgid",
                "AvailablePolicyTypes": [
                        {
                                "Status": "ENABLED",
                                "Type": "SERVICE_CONTROL_POLICY"
                        }
                ]
        }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeOrganization。

describe-organizational-unit

以下程式碼範例顯示如何使用 describe-organizational-unit。

AWS CLI

取得 OU 的相關資訊

下列describe-organizational-unit範例會請求有關 OU 的詳細資訊。

```
aws organizations describe-organizational-unit \
    --organizational-unit-id ou-examplerootid111-exampleouid111
```

輸出：

```
{
    "OrganizationalUnit": {
        "Name": "Accounting Group",
        "Arn": "arn:aws:organizations::123456789012:ou/o-exampleorgid/ou-
examplerootid111-exampleouid111",
        "Id": "ou-examplerootid111-exampleouid111"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeOrganizationalUnit。

describe-policy

以下程式碼範例顯示如何使用 describe-policy。

AWS CLI

取得政策的相關資訊

下列範例示範如何請求政策的相關資訊：

```
aws organizations describe-policy --policy-id p-examplepolicyid111
```

輸出包含政策物件，其中包含政策的詳細資訊：

```
{
        "Policy": {
                "Content": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement\": [\n
    {\n      \"Effect\": \"Allow\",\n      \"Action\": \"*\",\n      \"Resource\":
 \"*\"\n    }\n  ]\n}",
                "PolicySummary": {
                        "Arn": "arn:aws:organizations::111111111111:policy/o-
exampleorgid/service_control_policy/p-examplepolicyid111",
                        "Type": "SERVICE_CONTROL_POLICY",
                        "Id": "p-examplepolicyid111",
                        "AwsManaged": false,
                        "Name": "AllowAllS3Actions",
                        "Description": "Enables admins to delegate S3 permissions"
                }
        }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePolicy。

detach-policy

以下程式碼範例顯示如何使用 detach-policy。

AWS CLI

從根帳戶、OU 或帳戶分離政策

下列範例示範如何從 OU 分離政策：

```
aws organizations  detach-policy  --target-id ou-examplerootid111-exampleouid111 --
policy-id p-examplepolicyid111
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetachPolicy。

disable-policy-type

以下程式碼範例顯示如何使用 disable-policy-type。

AWS CLI

在根目錄中停用政策類型

下列範例示範如何在根目錄中停用服務控制政策 (SCP) 政策類型：

```
aws organizations disable-policy-type --root-id r-examplerootid111 --policy-
type SERVICE_CONTROL_POLICY
```

輸出顯示 PolicyTypes 回應元素不再包含 SERVICE_CONTROL_POLICY：

```
{
        "Root": {
                "PolicyTypes": [],
                "Name": "Root",
                "Id": "r-examplerootid111",
                "Arn": "arn:aws:organizations::111111111111:root/o-exampleorgid/r-
examplerootid111"
        }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisablePolicyType。

enable-all-features

以下程式碼範例顯示如何使用 enable-all-features。

AWS CLI

啟用組織中的所有功能

此範例顯示管理員要求組織中的所有受邀帳戶核准組織中已啟用的所有功能。 AWS 組織會傳送電子郵件到向每個受邀成員帳戶註冊的地址，要求擁有者透過接受傳送的交握來核准所有功能的變更。在所有獲邀請的成員帳戶接受交握後，組織管理員可以完成所有功能的變更，具有適當許可的人員可以建立政策並將其套用至根帳戶、OUs 和帳戶：

```
aws organizations enable-all-features
```

輸出是交握物件，會傳送至所有受邀成員帳戶進行核准：

```
{
        "Handshake": {
                "Action": "ENABLE_ALL_FEATURES",
                "Arn":"arn:aws:organizations::111111111111:handshake/o-exampleorgid/
enable_all_features/h-examplehandshakeid111",
                "ExpirationTimestamp":1.483127868609E9,
                "Id":"h-examplehandshakeid111",
                "Parties": [
                        {
                                "id":"o-exampleorgid",
                                "type":"ORGANIZATION"
                        }
                ],
                "requestedTimestamp":1.481831868609E9,
                "resources": [
                        {
                                "type":"ORGANIZATION",
                                "value":"o-exampleorgid"
                        }
                ],
                "state":"REQUESTED"
        }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableAllFeatures。

enable-policy-type

以下程式碼範例顯示如何使用 enable-policy-type。

AWS CLI

在根目錄中啟用政策類型的使用

下列範例顯示如何在根中啟用服務控制政策 (SCP) 政策類型：

```
aws organizations enable-policy-type --root-id r-examplerootid111 --policy-
type SERVICE_CONTROL_POLICY
```

輸出會顯示根物件，其中包含 policyTypes 回應元素，指出現在已啟用 SCPs：

```
{
        "Root": {
                "PolicyTypes": [
                        {
                                "Status":"ENABLED",
                                "Type":"SERVICE_CONTROL_POLICY"
                        }
                ],
                "Id": "r-examplerootid111",
                "Name": "Root",
                "Arn": "arn:aws:organizations::111111111111:root/o-exampleorgid/r-
examplerootid111"
        }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnablePolicyType。

`invite-account-to-organization`

以下程式碼範例顯示如何使用 `invite-account-to-organization`。

AWS CLI

邀請帳戶加入組織

下列範例顯示 bill@example.com 擁有的主帳戶邀請 juan@example.com 擁有的帳戶加入組織：

```
aws organizations invite-account-to-organization --target '{"Type": "EMAIL", "Id":
 "juan@example.com"}' --notes "This is a request for Juan's account to join Bill's
 organization."
```

輸出包含交握結構，顯示傳送至受邀帳戶的內容：

```
{
        "Handshake": {
                "Action": "INVITE",
                "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
                "ExpirationTimestamp": 1482952459.257,
```

```
                "Id": "h-examplehandshakeid111",
                "Parties": [
                        {
                                "Id": "o-exampleorgid",
                                "Type": "ORGANIZATION"
                        },
                        {
                                "Id": "juan@example.com",
                                "Type": "EMAIL"
                        }
                ],
                "RequestedTimestamp": 1481656459.257,
                "Resources": [
                        {
                                "Resources": [
                                        {
                                                "Type": "MASTER_EMAIL",
                                                "Value": "bill@amazon.com"
                                        },
                                        {
                                                "Type": "MASTER_NAME",
                                                "Value": "Org Master Account"
                                        },
                                        {
                                                "Type": "ORGANIZATION_FEATURE_SET",
                                                "Value": "FULL"
                                        }
                                ],
                                "Type": "ORGANIZATION",
                                "Value": "o-exampleorgid"
                        },
                        {
                                "Type": "EMAIL",
                                "Value": "juan@example.com"
                        }
                ],
                "State": "OPEN"
        }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 InviteAccountToOrganization。

leave-organization

以下程式碼範例顯示如何使用 leave-organization。

AWS CLI

將組織保留為成員帳戶

下列範例顯示請求離開組織的成員帳戶的管理員，其目前是 的成員：

```
aws organizations leave-organization
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 LeaveOrganization。

list-accounts-for-parent

以下程式碼範例顯示如何使用 list-accounts-for-parent。

AWS CLI

擷取指定父根或 OU 中所有帳戶的清單

下列範例顯示如何請求 OU 中的帳戶清單：

```
aws organizations list-accounts-for-parent --parent-id ou-examplerootid111-
exampleouid111
```

輸出包含帳戶摘要物件的清單。

```
{
        "Accounts": [
                {
                        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/333333333333",
                        "JoinedMethod": "INVITED",
                        "JoinedTimestamp": 1481835795.536,
                        "Id": "333333333333",
                        "Name": "Development Account",
                        "Email": "juan@example.com",
                        "Status": "ACTIVE"
                },
```

```
                {
                        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/444444444444",
                        "JoinedMethod": "INVITED",
                        "JoinedTimestamp": 1481835812.143,
                        "Id": "444444444444",
                        "Name": "Test Account",
                        "Email": "anika@example.com",
                        "Status": "ACTIVE"
                }
        ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAccountsForParent。

`list-accounts`

以下程式碼範例顯示如何使用 `list-accounts`。

AWS CLI

擷取組織中所有帳戶的清單

下列範例示範如何請求組織中的帳戶清單：

```
aws organizations list-accounts
```

輸出包含帳戶摘要物件的清單。

```
{
        "Accounts": [
                {
                        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/111111111111",
                        "JoinedMethod": "INVITED",
                        "JoinedTimestamp": 1481830215.45,
                        "Id": "111111111111",
                        "Name": "Master Account",
                        "Email": "bill@example.com",
                        "Status": "ACTIVE"
                },
                {
```

```
                        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/222222222222",
                        "JoinedMethod": "INVITED",
                        "JoinedTimestamp": 1481835741.044,
                        "Id": "222222222222",
                        "Name": "Production Account",
                        "Email": "alice@example.com",
                        "Status": "ACTIVE"
                },
                {
                        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/333333333333",
                        "JoinedMethod": "INVITED",
                        "JoinedTimestamp": 1481835795.536,
                        "Id": "333333333333",
                        "Name": "Development Account",
                        "Email": "juan@example.com",
                        "Status": "ACTIVE"
                },
                {
                        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/444444444444",
                        "JoinedMethod": "INVITED",
                        "JoinedTimestamp": 1481835812.143,
                        "Id": "444444444444",
                        "Name": "Test Account",
                        "Email": "anika@example.com",
                        "Status": "ACTIVE"
                }
        ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAccounts。

`list-children`

以下程式碼範例顯示如何使用 `list-children`。

AWS CLI

擷取父 OUs 或根的子帳戶和 OU

下列範例說明如何列出包含該帳戶 444444444444 的根或 OU：

```
aws organizations list-children --child-type ORGANIZATIONAL_UNIT --parent-id ou-examplerootid111-exampleouid111
```

輸出顯示父系包含的兩個子 OUs：

```
{
        "Children": [
                {
                        "Id": "ou-examplerootid111-exampleouid111",
                        "Type":"ORGANIZATIONAL_UNIT"
                },
                {
                        "Id":"ou-examplerootid111-exampleouid222",
                        "Type":"ORGANIZATIONAL_UNIT"
                }
        ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListChildren。

list-create-account-status

以下程式碼範例顯示如何使用 list-create-account-status。

AWS CLI

範例 1：擷取目前組織中提出的帳戶建立請求清單

下列範例顯示如何為成功完成的組織請求帳戶建立請求清單：

```
aws organizations list-create-account-status --states SUCCEEDED
```

輸出包含 物件陣列，其中包含每個請求的相關資訊。

```
{
        "CreateAccountStatuses": [
                {
                        "AccountId": "444444444444",
                        "AccountName": "Developer Test Account",
                        "CompletedTimeStamp": 1481835812.143,
                        "Id": "car-examplecreateaccountrequestid111",
                        "RequestedTimeStamp": 1481829432.531,
```

```
                        "State": "SUCCEEDED"
                }
        ]
}
```

範例 2：擷取目前組織中提出的進行中帳戶建立請求清單

下列範例會取得組織的進行中帳戶建立請求清單：

```
aws organizations list-create-account-status --states IN_PROGRESS
```

輸出包含 物件陣列，其中包含每個請求的相關資訊。

```
{
        "CreateAccountStatuses": [
                {
                  "State": "IN_PROGRESS",
                  "Id": "car-examplecreateaccountrequestid111",
                  "RequestedTimeStamp": 1481829432.531,
                  "AccountName": "Production Account"
                }
        ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCreateAccountStatus。

`list-handshakes-for-account`

以下程式碼範例顯示如何使用 `list-handshakes-for-account`。

AWS CLI

擷取傳送至 帳戶的交握清單

下列範例示範如何取得與用來呼叫 操作的登入資料帳戶相關聯的所有交握清單：

```
aws organizations list-handshakes-for-account
```

輸出包含交握結構清單，其中包含每個交握的相關資訊，包括其目前狀態：

```
{
        "Handshake": {
```

```
                "Action": "INVITE",
                "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
                "ExpirationTimestamp": 1482952459.257,
                "Id": "h-examplehandshakeid111",
                "Parties": [
                        {
                                "Id": "o-exampleorgid",
                                "Type": "ORGANIZATION"
                        },
                        {
                                "Id": "juan@example.com",
                                "Type": "EMAIL"
                        }
                ],
                "RequestedTimestamp": 1481656459.257,
                "Resources": [
                        {
                                "Resources": [
                                        {
                                                "Type": "MASTER_EMAIL",
                                                "Value": "bill@amazon.com"
                                        },
                                        {
                                                "Type": "MASTER_NAME",
                                                "Value": "Org Master Account"
                                        },
                                        {
                                                "Type": "ORGANIZATION_FEATURE_SET",
                                                "Value": "FULL"
                                        }
                                ],
                                "Type": "ORGANIZATION",
                                "Value": "o-exampleorgid"
                        },
                        {
                                "Type": "EMAIL",
                                "Value": "juan@example.com"
                        }
                ],
                "State": "OPEN"
        }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListHandshakesForAccount。

list-handshakes-for-organization

以下程式碼範例顯示如何使用 list-handshakes-for-organization。

AWS CLI

擷取與組織相關聯的交握清單

下列範例顯示如何取得與目前組織相關聯的交握清單：

```
aws organizations list-handshakes-for-organization
```

輸出會顯示兩個交握。第一個是 Juan 帳戶的邀請，並顯示 OPEN 狀態。第二個是 Anika 帳戶的邀請，並顯示接受狀態：

```
{
        "Handshakes": [
                {
                        "Action": "INVITE",
                        "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
                        "ExpirationTimestamp": 1482952459.257,
                        "Id": "h-examplehandshakeid111",
                        "Parties": [
                                {
                                        "Id": "o-exampleorgid",
                                        "Type": "ORGANIZATION"
                                },
                                {
                                        "Id": "juan@example.com",
                                        "Type": "EMAIL"
                                }
                        ],
                        "RequestedTimestamp": 1481656459.257,
                        "Resources": [
                                {
                                        "Resources": [
                                                {
                                                        "Type": "MASTER_EMAIL",
                                                        "Value": "bill@amazon.com"
                                                },
```

```
                                                {
                                                        "Type": "MASTER_NAME",
                                                        "Value": "Org Master
 Account"
                                                },
                                                {
                                                        "Type":
 "ORGANIZATION_FEATURE_SET",
                                                        "Value": "FULL"
                                                }
                                        ],
                                        "Type": "ORGANIZATION",
                                        "Value": "o-exampleorgid"
                                },
                                {
                                        "Type": "EMAIL",
                                        "Value": "juan@example.com"
                                },
                                {
                                        "Type":"NOTES",
                                        "Value":"This is an invitation to Juan's
 account to join Bill's organization."
                                }
                        ],
                        "State": "OPEN"
                },
                {
                        "Action": "INVITE",
                        "State":"ACCEPTED",
                        "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
                        "ExpirationTimestamp": 1.471797437427E9,
                        "Id": "h-examplehandshakeid222",
                        "Parties": [
                                {
                                        "Id": "o-exampleorgid",
                                        "Type": "ORGANIZATION"
                                },
                                {
                                        "Id": "anika@example.com",
                                        "Type": "EMAIL"
                                }
                        ],
                        "RequestedTimestamp": 1.469205437427E9,
```

```
                        "Resources": [
                                {
                                        "Resources": [
                                                {
                                                        "Type":"MASTER_EMAIL",
                                                        "Value":"bill@example.com"
                                                },
                                                {
                                                        "Type":"MASTER_NAME",
                                                        "Value":"Master Account"
                                                }
                                        ],
                                        "Type":"ORGANIZATION",
                                        "Value":"o-exampleorgid"
                                },
                                {
                                        "Type":"EMAIL",
                                        "Value":"anika@example.com"
                                },
                                {
                                        "Type":"NOTES",
                                        "Value":"This is an invitation to Anika's
 account to join Bill's organization."
                                }
                        ]
                }
        ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListHandshakesForOrganization。

list-organizational-units-for-parent

以下程式碼範例顯示如何使用 list-organizational-units-for-parent。

AWS CLI

擷取父 OUs 或根中的 OU 清單

下列範例示範如何取得指定根目錄中的 OUs 清單：

```
aws organizations list-organizational-units-for-parent --parent-id r-
examplerootid111
```

輸出顯示指定的根包含兩個 OUs並顯示每個 的詳細資訊：

```
{
        "OrganizationalUnits": [
                {
                        "Name": "AccountingDepartment",
                        "Arn": "arn:aws:organizations::o-exampleorgid:ou/r-
examplerootid111/ou-examplerootid111-exampleouid111"
                },
                {
                        "Name": "ProductionDepartment",
                        "Arn": "arn:aws:organizations::o-exampleorgid:ou/r-
examplerootid111/ou-examplerootid111-exampleouid222"
                }
        ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListOrganizationalUnitsForParent。

`list-parents`

以下程式碼範例顯示如何使用 `list-parents`。

AWS CLI

列出帳戶或子 OUs 的父 OU 或根

下列範例說明如何列出包含該帳戶 444444444444 的根或父 OU：

```
aws organizations list-parents --child-id 444444444444
```

輸出顯示指定的帳戶位於具有指定 ID 的 OU 中：

```
{
  "Parents": [
        {
          "Id": "ou-examplerootid111-exampleouid111",
          "Type": "ORGANIZATIONAL_UNIT"
        }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListParents。

list-policies-for-target

以下程式碼範例顯示如何使用 list-policies-for-target。

AWS CLI

擷取直接連接到帳戶的 SCPs 清單

下列範例顯示如何取得直接連接至 帳戶之篩選條件參數所指定之所有服務控制政策 (SCPs) 的清單：

```
aws organizations list-policies-for-target --filter SERVICE_CONTROL_POLICY --target-
id 444444444444
```

輸出包含政策結構的清單，其中包含政策的摘要資訊。此清單不包含套用到帳戶的政策，因為繼承自 OU 階層中的位置：

```
{
        "Policies": [
                {
                        "Type": "SERVICE_CONTROL_POLICY",
                        "Name": "AllowAllEC2Actions",
                        "AwsManaged", false,
                        "Id": "p-examplepolicyid222",
                        "Arn": "arn:aws:organizations::o-exampleorgid:policy/
service_control_policy/p-examplepolicyid222",
                        "Description": "Enables account admins to delegate
 permissions for any EC2 actions to users and roles in their accounts."
                }
        ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPoliciesForTarget。

list-policies

以下程式碼範例顯示如何使用 list-policies。

AWS CLI

擷取特定類型組織中所有政策的清單

下列範例示範如何取得篩選條件參數所指定的 SCPs 清單：

```
aws organizations list-policies --filter SERVICE_CONTROL_POLICY
```

輸出包含具有摘要資訊的政策清單：

```
{
        "Policies": [
                {
                        "Type": "SERVICE_CONTROL_POLICY",
                        "Name": "AllowAllS3Actions",
                        "AwsManaged": false,
                        "Id": "p-examplepolicyid111",
                        "Arn": "arn:aws:organizations::111111111111:policy/
service_control_policy/p-examplepolicyid111",
                        "Description": "Enables account admins to delegate
 permissions for any S3 actions to users and roles in their accounts."
                },
                {
                        "Type": "SERVICE_CONTROL_POLICY",
                        "Name": "AllowAllEC2Actions",
                        "AwsManaged": false,
                        "Id": "p-examplepolicyid222",
                        "Arn": "arn:aws:organizations::111111111111:policy/
service_control_policy/p-examplepolicyid222",
                        "Description": "Enables account admins to delegate
 permissions for any EC2 actions to users and roles in their accounts."
                },
                {
                        "AwsManaged": true,
                        "Description": "Allows access to every operation",
                        "Type": "SERVICE_CONTROL_POLICY",
                        "Id": "p-FullAWSAccess",
                        "Arn": "arn:aws:organizations::aws:policy/
service_control_policy/p-FullAWSAccess",
                        "Name": "FullAWSAccess"
                }
        ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListPolicies。

list-roots

以下程式碼範例顯示如何使用 list-roots。

AWS CLI

擷取組織中根的清單

此範例說明如何取得組織的根目錄清單：

```
aws organizations list-roots
```

輸出包含具有摘要資訊的根結構清單：

```
{
        "Roots": [
                {
                        "Name": "Root",
                        "Arn": "arn:aws:organizations::111111111111:root/o-
exampleorgid/r-examplerootid111",
                        "Id": "r-examplerootid111",
                        "PolicyTypes": [
                                {
                                        "Status":"ENABLED",
                                        "Type":"SERVICE_CONTROL_POLICY"
                                }
                        ]
                }
        ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRoots。

list-targets-for-policy

以下程式碼範例顯示如何使用 list-targets-for-policy。

AWS CLI

擷取政策連接的根目錄、OUs 和帳戶清單

下列範例示範如何取得所指定政策所連接的根目錄、OUs 和帳戶的清單：

```
aws organizations list-targets-for-policy --policy-id p-FullAWSAccess
```

輸出包含連接物件的清單，其中包含政策所連接之根、OUs 和帳戶的摘要資訊：

```
{
        "Targets": [
                {
                        "Arn": "arn:aws:organizations::111111111111:root/o-
exampleorgid/r-examplerootid111",
                        "Name": "Root",
                        "TargetId":"r-examplerootid111",
                        "Type":"ROOT"
                },
                {
                        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/333333333333;",
                        "Name": "Developer Test Account",
                        "TargetId": "333333333333",
                        "Type": "ACCOUNT"
                },
                {
                        "Arn":"arn:aws:organizations::111111111111:ou/o-
exampleorgid/ou-examplerootid111-exampleouid111",
                        "Name":"Accounting",
                        "TargetId":"ou-examplerootid111-exampleouid111",
                        "Type":"ORGANIZATIONAL_UNIT"
                }
        ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTargetsForPolicy。

`move-account`

以下程式碼範例顯示如何使用 `move-account`。

AWS CLI

在根或 OUs 之間移動帳戶

下列範例說明如何將組織中的主帳戶從根目錄移至 OU：

```
aws organizations move-account --account-id 333333333333 --source-parent-id r-examplerootid111 --destination-parent-id ou-examplerootid111-exampleouid111
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 MoveAccount。

remove-account-from-organization

以下程式碼範例顯示如何使用 remove-account-from-organization。

AWS CLI

將 帳戶從組織中移除為主帳戶

下列範例示範如何從組織中移除 帳戶：

```
aws organizations remove-account-from-organization --account-id 333333333333
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveAccountFromOrganization。

update-organizational-unit

以下程式碼範例顯示如何使用 update-organizational-unit。

AWS CLI

重新命名 OU

此範例說明如何重新命名 OU：在此範例中，OU 已重新命名為「AccountingOU」：

```
aws organizations update-organizational-unit --organizational-unit-id ou-examplerootid111-exampleouid111 --name AccountingOU
```

輸出會顯示新名稱：

```
{
        "OrganizationalUnit": {
                "Id": "ou-examplerootid111-exampleouid111"
```

```
                "Name": "AccountingOU",
                "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-
examplerootid111-exampleouid111""
        }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateOrganizationalUnit。

update-policy

以下程式碼範例顯示如何使用 update-policy。

AWS CLI

範例 1：重新命名政策

下列update-policy範例會重新命名政策並提供新的描述。

```
aws organizations update-policy \
    --policy-id p-examplepolicyid111 \
    --name Renamed-Policy \
    --description "This description replaces the original."
```

輸出會顯示新名稱和描述。

```
{
    "Policy": {
        "Content": "{\n  \"Version\":\"2012-10-17\",\n  \"Statement\":{\n
  \"Effect\":\"Allow\",\n    \"Action\":\"ec2:*\",\n    \"Resource\":\"*\"\n
  }\n}\n",
        "PolicySummary": {
            "Id": "p-examplepolicyid111",
            "AwsManaged": false,
            "Arn":"arn:aws:organizations::111111111111:policy/o-exampleorgid/
service_control_policy/p-examplepolicyid111",
            "Description": "This description replaces the original.",
            "Name": "Renamed-Policy",
            "Type": "SERVICE_CONTROL_POLICY"
        }
    }
}
```

範例 2：取代政策的 JSON 文字內容

下列範例說明如何使用允許 S3 而非 EC2 的新 JSON 政策文字字串取代先前範例中 SCP 的 JSON 文字：

```
aws organizations update-policy \
    --policy-id p-examplepolicyid111 \
    --content "{\"Version\":\"2012-10-17\",\"Statement\":{\"Effect\":\"Allow\",
\"Action\":\"s3:*\",\"Resource\":\"*\"}}"
```

輸出會顯示新內容：

```
{
    "Policy": {
        "Content": "{ \"Version\": \"2012-10-17\", \"Statement\": { \"Effect\":
 \"Allow\", \"Action\": \"s3:*\", \"Resource\": \"*\" } }",
        "PolicySummary": {
            "Arn": "arn:aws:organizations::111111111111:policy/o-exampleorgid/
service_control_policy/p-examplepolicyid111",
            "AwsManaged": false;
            "Description": "This description replaces the original.",
            "Id": "p-examplepolicyid111",
            "Name": "Renamed-Policy",
            "Type": "SERVICE_CONTROL_POLICY"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePolicy。

AWS Outposts 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS Outposts。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

get-outpost-instance-types

以下程式碼範例顯示如何使用 get-outpost-instance-types。

AWS CLI

取得 Outpost 上的執行個體類型

下列get-outpost-instance-types範例會取得指定 Outpost 的執行個體類型。

```
aws outposts get-outpost-instance-types \
    --outpost-id op-0ab23c4567EXAMPLE
```

輸出：

```
{
    "InstanceTypes": [
        {
            "InstanceType": "c5d.large"
        },
        {
            "InstanceType": "i3en.24xlarge"
        },
        {
            "InstanceType": "m5d.large"
        },
        {
            "InstanceType": "r5d.large"
        }
    ],
    "OutpostId": "op-0ab23c4567EXAMPLE",
    "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-0ab23c4567EXAMPLE"
}
```

如需詳細資訊，請參閱 Outpost 使用者指南中的在 Outpost 上啟動執行個體。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetOutpostInstanceTypes。

get-outpost

以下程式碼範例顯示如何使用 get-outpost。

AWS CLI

取得 Outpost 詳細資訊

下列get-outpost範例顯示指定 Outpost 的詳細資訊。

```
aws outposts get-outpost \
    --outpost-id op-0ab23c4567EXAMPLE
```

輸出：

```
{
    "Outpost": {
        "OutpostId": "op-0ab23c4567EXAMPLE",
        "OwnerId": "123456789012",
        "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-0ab23c4567EXAMPLE",
        "SiteId": "os-0ab12c3456EXAMPLE",
        "Name": "EXAMPLE",
        "LifeCycleStatus": "ACTIVE",
        "AvailabilityZone": "us-west-2a",
        "AvailabilityZoneId": "usw2-az1",
        "Tags": {}
    }
}
```

如需詳細資訊，請參閱 Outpost 使用者指南中的使用 AWS Outpost。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetOutpost。

list-outposts

以下程式碼範例顯示如何使用 list-outposts。

AWS CLI

列出 Outpost

下列list-outposts範例列出您 AWS 帳戶中的 Outpost。

```
aws outposts list-outposts
```

輸出：

```
{
    "Outposts": [
        {
            "OutpostId": "op-0ab23c4567EXAMPLE",
            "OwnerId": "123456789012",
            "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-0ab23c4567EXAMPLE",
            "SiteId": "os-0ab12c3456EXAMPLE",
            "Name": "EXAMPLE",
            "Description": "example",
            "LifeCycleStatus": "ACTIVE",
            "AvailabilityZone": "us-west-2a",
            "AvailabilityZoneId": "usw2-az1",
            "Tags": {
                "Name": "EXAMPLE"
            }
        },
        {
            "OutpostId": "op-4fe3dc21baEXAMPLE",
            "OwnerId": "123456789012",
            "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-4fe3dc21baEXAMPLE",
            "SiteId": "os-0ab12c3456EXAMPLE",
            "Name": "EXAMPLE2",
            "LifeCycleStatus": "ACTIVE",
            "AvailabilityZone": "us-west-2a",
            "AvailabilityZoneId": "usw2-az1",
            "Tags": {}
        }
    ]
}
```

如需詳細資訊，請參閱 Outpost 使用者指南中的使用 AWS Outpost。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListOutposts。

`list-sites`

以下程式碼範例顯示如何使用 `list-sites`。

AWS CLI

列出網站

下列`list-sites`範例列出您 AWS 帳戶中可用的 Outpost 網站。

```
aws outposts list-sites
```

輸出：

```
{
    "Sites": [
        {
            "SiteId": "os-0ab12c3456EXAMPLE",
            "AccountId": "123456789012",
            "Name": "EXAMPLE",
            "Description": "example",
            "Tags": {}
        }
    ]
}
```

如需詳細資訊，請參閱 Outpost 使用者指南中的使用 AWS Outpost。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSites。

AWS Payment Cryptography 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS Payment Cryptography。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-alias`

以下程式碼範例顯示如何使用 `create-alias`。

AWS CLI

建立金鑰的別名

下列`create-alias`範例會建立金鑰的別名。

```
aws payment-cryptography create-alias \
    --alias-name alias/sampleAlias1 \
    --key-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

輸出：

```
{
    "Alias": {
        "AliasName": "alias/sampleAlias1",
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h"
    }
}
```

如需詳細資訊，請參閱《 AWS 付款密碼編譯使用者指南》中的關於別名。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateAlias。

`create-key`

以下程式碼範例顯示如何使用 `create-key`。

AWS CLI

建立金鑰

下列`create-key`範例會產生 2KEY TDES 金鑰，可用來產生和驗證 CVV/CVV2 值。

```
aws payment-cryptography create-key \
    --exportable \
```

```
    --key-
attributes KeyAlgorithm=TDES_2KEY, KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC
```

輸出：

```
{
    "Key": {
        "CreateTimestamp": "1686800690",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_2KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": false,
                "DeriveKey": false,
                "Encrypt": false,
                "Generate": true,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": false,
                "Verify": true,
                "Wrap": false
            },
            "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"
        },
        "KeyCheckValue": "F2E50F",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "1686800690"
    }
}
```

如需詳細資訊，請參閱《AWS 付款密碼編譯使用者指南》中的產生金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateKey。

delete-alias

以下程式碼範例顯示如何使用 delete-alias。

AWS CLI

刪除別名

下列delete-alias範例會刪除別名。它不會影響金鑰。

```
aws payment-cryptography delete-alias \
    --alias-name alias/sampleAlias1
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 AWS 付款密碼編譯使用者指南》中的關於別名。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteAlias。

delete-key

以下程式碼範例顯示如何使用 delete-key。

AWS CLI

刪除金鑰

下列delete-key範例會排程 7 天後刪除的金鑰，這是預設的等待期間。

```
aws payment-cryptography delete-key \
    --key-identifier arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h
```

輸出：

```
{
    "Key": {
        "CreateTimestamp": "1686801198",
        "DeletePendingTimestamp": "1687405998",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_2KEY",
            "KeyClass": "SYMMETRIC_KEY",
```

```
            "KeyModesOfUse": {
                "Decrypt": false,
                "DeriveKey": false,
                "Encrypt": false,
                "Generate": true,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": false,
                "Verify": true,
                "Wrap": false
            },
            "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"
        },
        "KeyCheckValue": "F2E50F",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "DELETE_PENDING",
        "UsageStartTimestamp": "1686801190"
    }
}
```

如需詳細資訊，請參閱《AWS 付款密碼編譯使用者指南》中的刪除金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteKey。

export-key

以下程式碼範例顯示如何使用 export-key。

AWS CLI

匯出金鑰

下列export-key範例匯出金鑰。

```
aws payment-cryptography export-key \
    --export-key-identifier arn:aws:payment-cryptography:us-west-2:123456789012:key/lco3w6agsk7zgu2l \
    --key-material '{"Tr34KeyBlock": { \
        "CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-cryptography:us-west-2:123456789012:key/ftobshq7pvioc5fx", \
        "ExportToken": "export-token-cu4lg26ofcziixny", \
        "KeyBlockFormat": "X9_TR34_2012", \
```

```
        "WrappingKeyCertificate": file://wrapping-key-certificate.pem }}'
```

wrapping-key-certificate.pem 的內容：

```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2VENDQXFXZ0F3SUJBZ0lSQU1ZZS8xMXFUK2svVzlRUDJQOElV
```

輸出：

```
{
    "WrappedKey": {
        "KeyMaterial":
 "308205A106092A864886F70D010702A08205923082058E020101310D300B0609608648016503040201308203F
        "WrappedKeyMaterialFormat": "TR34_KEY_BLOCK"
    }
}
```

如需詳細資訊，請參閱《AWS 付款密碼編譯使用者指南》中的匯出金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ExportKey。

get-alias

以下程式碼範例顯示如何使用 get-alias。

AWS CLI

取得別名

下列get-alias範例會傳回與別名相關聯的金鑰 ARN。

```
aws payment-cryptography get-alias \
    --alias-name alias/sampleAlias1
```

輸出：

```
{
    "Alias": {
        "AliasName": "alias/sampleAlias1",
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h"
    }
```

```
}
```

如需詳細資訊，請參閱AWS 《付款密碼編譯使用者指南》中的關於別名。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetAlias。

get-key

以下程式碼範例顯示如何使用 get-key。

AWS CLI

取得金鑰的中繼資料

下列get-key範例會傳回與別名相關聯的金鑰中繼資料。此操作不會傳回密碼編譯資料。

```
aws payment-cryptography get-key \
    --key-identifier alias/sampleAlias1
```

輸出：

```
{
    "Key": {
        "CreateTimestamp": "1686800690",
        "DeletePendingTimestamp": "1687405998",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_2KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": false,
                "DeriveKey": false,
                "Encrypt": false,
                "Generate": true,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": false,
                "Verify": true,
                "Wrap": false
            },
```

```
            "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"
        },
        "KeyCheckValue": "F2E50F",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "DELETE_PENDING",
        "UsageStartTimestamp": "1686801190"
    }
}
```

如需詳細資訊，請參閱《AWS 付款密碼編譯使用者指南》中的取得金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetKey。

get-parameters-for-export

以下程式碼範例顯示如何使用 get-parameters-for-export。

AWS CLI

初始化匯出程序

下列get-parameters-for-export範例會產生金鑰對、簽署金鑰，然後傳回憑證和憑證根。

```
aws payment-cryptography get-parameters-for-export \
    --signing-key-algorithm RSA_2048 \
    --key-material-type TR34_KEY_BLOCK
```

輸出：

```
{
    "ExportToken": "export-token-ep5cwyzune7oya53",
    "ParametersValidUntilTimestamp": "1687415640",
    "SigningKeyAlgorithm": "RSA_2048",
    "SigningKeyCertificate":

    "MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
```

```
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
    "SigningKeyCertificateChain":
    "NIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
}
```

如需詳細資訊，請參閱《AWS 付款密碼編譯使用者指南》中的匯出金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetParametersForExport。

get-parameters-for-import

以下程式碼範例顯示如何使用 get-parameters-for-import。

AWS CLI

初始化匯入程序

下列get-parameters-for-import範例會產生金鑰對、簽署金鑰，然後傳回憑證和憑證根。

```
aws payment-cryptography get-parameters-for-import \
    --key-material-type TR34_KEY_BLOCK \
    --wrapping-key-algorithm RSA_2048
```

輸出：

```
{
    "ImportToken": "import-token-qgmafpaa7nt2kfbb",
    "ParametersValidUntilTimestamp": "1687415640",
    "WrappingKeyAlgorithm": "RSA_2048",
    "WrappingKeyCertificate":
    "MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
    "WrappingKeyCertificateChain":
    "NIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
}
```

如需詳細資訊，請參閱《AWS 付款密碼編譯使用者指南》中的匯入金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetParametersForImport。

get-public-key-certificate

以下程式碼範例顯示如何使用 get-public-key-certificate。

AWS CLI

傳回公有金鑰

下列get-public-key-certificate範例會傳回金鑰對的公有金鑰部分。

```
aws payment-cryptography get-public-key-certificate \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

輸出：

```
{
    "KeyCertificate":
    "MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
    "KeyCertificateChain":
    "NIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
```

```
}
```

如需詳細資訊，請參閱《 AWS 付款密碼編譯使用者指南》中的取得與金鑰對相關聯的公有金鑰/憑證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPublicKeyCertificate。

import-key

以下程式碼範例顯示如何使用 import-key。

AWS CLI

匯入 TR-34 金鑰

下列import-key範例會匯入 TR-34 金鑰。

```
aws payment-cryptography import-key \
        --key-material='{ "Tr34KeyBlock": {" \
            CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-
cryptography:us-west-2:123456789012:key/rmm5wn2q564njnjm", \
            "ImportToken": "import-token-5ott6ho5nts7bbcg", \
            "KeyBlockFormat": "X9_TR34_2012", \
            "SigningKeyCertificate": file://signing-key-certificate.pem, \
            "WrappedKeyBlock": file://wrapped-key-block.pem }}'
```

signing-key-certificate.pem 的內容：

```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2RENDQXFTZ0F3SUJBZ0lRYWVCK25IbE1WZU1PR1ZiNjU1Q2Jz
```

wrapped-key-block.pem 的內容：

```
3082059806092A864886F70D010702A0820589308205850201013 10D300B0609608648016503040201308203160 6
```

輸出：

```
{
    "Key": {
        "CreateTimestamp": "2023-06-09T16:56:27.621000-07:00",
        "Enabled": true,
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
bzmvgyxdg3sktwxd",
```

```
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_2KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": false,
                "DeriveKey": false,
                "Encrypt": false,
                "Generate": true,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": false,
                "Verify": true,
                "Wrap": false
            },
            "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"
        },
        "KeyCheckValue": "D9B20E",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "EXTERNAL",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2023-06-09T16:56:27.621000-07:00"
    }
}
```

如需詳細資訊，請參閱《AWS 付款密碼編譯使用者指南》中的匯入金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ImportKey。

list-aliases

以下程式碼範例顯示如何使用 list-aliases。

AWS CLI

取得別名清單

下列list-aliases範例顯示您帳戶中在此區域的所有別名。

```
aws payment-cryptography list-aliases
```

輸出：

```
{
```

```
    "Aliases": [
        {
            "AliasName": "alias/sampleAlias1",
            "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h"
        },
        {
            "AliasName": "alias/sampleAlias2",
            "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h"
        }
    ]
}
```

如需詳細資訊，請參閱《 AWS 付款密碼編譯使用者指南》中的關於別名。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAliases。

list-keys

以下程式碼範例顯示如何使用 list-keys。

AWS CLI

取得金鑰清單

下列list-keys範例顯示您帳戶中在此區域的所有金鑰。

```
aws payment-cryptography list-keys
```

輸出：

```
{
    "Keys": [
    {
        "CreateTimestamp": "1666506840",
        "Enabled": false,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_3KEY",
            "KeyClass": "SYMMETRIC_KEY",
```

```
            "KeyModesOfUse": {
                "Decrypt": true,
                "DeriveKey": false,
                "Encrypt": true,
                "Generate": false,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": true,
                "Verify": false,
                "Wrap": true
            },
            "KeyUsage": "TR31_P1_PIN_GENERATION_KEY"
        },
        "KeyCheckValue": "369D",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStopTimestamp": "1666938840"
    }
    ]
}
```

如需詳細資訊，請參閱《AWS 付款密碼編譯使用者指南》中的列出金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListKeys。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

取得金鑰的標籤清單

下列list-tags-for-resource範例會取得金鑰的標籤。

```
aws payment-cryptography list-tags-for-resource \
    --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

輸出：

```
{
```

```
    "Tags": [
        {
            "Key": "BIN",
            "Value": "20151120"
        },
        {
            "Key": "Project",
            "Value": "Production"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS 付款密碼編譯使用者指南》中的使用 API 操作管理金鑰標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

restore-key

以下程式碼範例顯示如何使用 restore-key。

AWS CLI

還原排程刪除的金鑰

下列restore-key範例會取消刪除金鑰。

```
aws payment-cryptography restore-key \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

輸出：

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyUsage": "TR31_V2_VISA_PIN_VERIFICATION_KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyAlgorithm": "TDES_3KEY",
            "KeyModesOfUse": {
                "Encrypt": false,
                "Decrypt": false,
```

```
                "Wrap": false,
                "Unwrap": false,
                "Generate": true,
                "Sign": false,
                "Verify": true,
                "DeriveKey": false,
                "NoRestrictions": false
            }
        },
        "KeyCheckValue": "",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "Enabled": false,
        "Exportable": true,
        "KeyState": "CREATE_COMPLETE",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "CreateTimestamp": "1686800690",
        "UsageStopTimestamp": "1687405998"
    }
}
```

如需詳細資訊，請參閱《AWS 付款密碼編譯使用者指南》中的刪除金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreKey。

start-key-usage

以下程式碼範例顯示如何使用 start-key-usage。

AWS CLI

啟用金鑰

下列start-key-usage範例會啟用金鑰。

```
aws payment-cryptography start-key-usage \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

輸出：

```
{
    "Key": {
        "CreateTimestamp": "1686800690",
```

```
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
alsuwfxug3pgy6xh",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_3KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": true,
                "DeriveKey": false,
                "Encrypt": true,
                "Generate": false,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": true,
                "Verify": false,
                "Wrap": true
            },
            "KeyUsage": "TR31_P1_PIN_GENERATION_KEY"
        },
        "KeyCheckValue": "369D",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "1686800690"
    }
}
```

如需詳細資訊，請參閱AWS 《付款密碼編譯使用者指南》中的啟用和停用金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartKeyUsage。

stop-key-usage

以下程式碼範例顯示如何使用 stop-key-usage。

AWS CLI

停用金鑰

下列stop-key-usage範例會停用金鑰。

```
aws payment-cryptography stop-key-usage \
```

```
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

輸出：

```
{
    "Key": {
        "CreateTimestamp": "1686800690",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
alsuwfxug3pgy6xh",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_3KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": true,
                "DeriveKey": false,
                "Encrypt": true,
                "Generate": false,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": true,
                "Verify": false,
                "Wrap": true
            },
            "KeyUsage": "TR31_P1_PIN_GENERATION_KEY"
        },
        "KeyCheckValue": "369D",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "1686800690"
    }
}
```

如需詳細資訊，請參閱AWS 《付款密碼編譯使用者指南》中的啟用和停用金鑰。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopKeyUsage。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

標記金鑰

下列tag-resource範例會標記金鑰。

```
aws payment-cryptography tag-resource \
    --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
    --tags Key=sampleTag,Value=sampleValue
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS 付款密碼編譯使用者指南》中的管理金鑰標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從金鑰移除標籤

下列untag-resource範例會從金鑰移除標籤。

```
aws payment-cryptography untag-resource \
    --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
    --tag-keys sampleTag
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS 付款密碼編譯使用者指南》中的管理金鑰標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-alias

以下程式碼範例顯示如何使用 update-alias。

AWS CLI

更新別名

下列update-alias範例會將別名與不同的金鑰建立關聯。

```
aws payment-cryptography update-alias \
    --alias-name alias/sampleAlias1 \
    --key-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
tqv5yij6wtxx64pi
```

輸出：

```
{
    "Alias": {
        "AliasName": "alias/sampleAlias1",
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
tqv5yij6wtxx64pi "
    }
}
```

如需詳細資訊，請參閱《 AWS 付款密碼編譯使用者指南》中的關於別名。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateAlias。

AWS Payment Cryptography 使用 的資料平面範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 AWS Payment Cryptography 資料平面來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

decrypt-data

以下程式碼範例顯示如何使用 decrypt-data。

AWS CLI

解密加密文字

下列decrypt-data範例使用對稱金鑰解密加密文字資料。對於此操作，金鑰必須KeyModesOfUse設定為 Decrypt，並將 KeyUsage設定為 TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY。

```
aws payment-cryptography-data decrypt-data \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
    --cipher-text 33612AB9D6929C3A828EB6030082B2BD \
    --decryption-attributes 'Symmetric={Mode=CBC}'
```

輸出：

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
    "KeyCheckValue": "71D7AE",
    "PlainText": "313233343132333431323334313233334"
}
```

如需詳細資訊，請參閱AWS 《付款密碼編譯使用者指南》中的解密資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DecryptData。

encrypt-data

以下程式碼範例顯示如何使用 encrypt-data。

AWS CLI

加密資料

下列encrypt-data範例使用對稱金鑰加密純文字資料。對於此操作，金鑰必須KeyModesOfUse設定為 Encrypt，並將 KeyUsage設定為 TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY。

```
aws payment-cryptography-data encrypt-data \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
    --plain-text 31323334313233343132333431323334 \
    --encryption-attributes 'Symmetric={Mode=CBC}'
```

輸出：

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
    "KeyCheckValue": "71D7AE",
    "CipherText": "33612AB9D6929C3A828EB6030082B2BD"
}
```

如需詳細資訊，請參閱AWS 《付款密碼編譯使用者指南》中的加密資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EncryptData。

generate-card-validation-data

以下程式碼範例顯示如何使用 generate-card-validation-data。

AWS CLI

產生 CVV

下列generate-card-validation-data範例會產生 CVV/CVV2。

```
aws payment-cryptography-data generate-card-validation-data \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
    --primary-account-number=171234567890123 \
    --generation-attributes CardVerificationValue2={CardExpiryDate=0123}
```

輸出：

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
    "KeyCheckValue": "CADDA1",
    "ValidationData": "801"
}
```

如需詳細資訊，請參閱AWS 《付款密碼編譯使用者指南》中的產生卡片資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GenerateCardValidationData。

generate-mac

以下程式碼範例顯示如何使用 generate-mac。

AWS CLI

產生 MAC

下列generate-card-validation-data範例會使用演算法 HMAC_SHA256 和 HMAC 加密金鑰，為卡片資料身分驗證產生雜湊型訊息驗證碼 (HMAC)。金鑰必須KeyUsage設定為 TR31_M7_HMAC_KEY，並將 KeyModesOfUse設定為 Generate。

```
aws payment-cryptography-data generate-mac \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
    --message-
data "3b31303838343930303130373333934313533d32343038323236303030373030303f33" \
    --generation-attributes Algorithm=HMAC_SHA256
```

輸出：

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h,
    "KeyCheckValue": "2976E7",
    "Mac": "ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C"
}
```

如需詳細資訊，請參閱《AWS 付款密碼編譯使用者指南》中的產生 MAC。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GenerateMac。

generate-pin-data

以下程式碼範例顯示如何使用 generate-pin-data。

AWS CLI

產生 PIN

以下generate-card-validation-data範例使用 Visa PIN 配置產生新的隨機 PIN。

```
aws payment-cryptography-data generate-pin-data \
    --generation-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/37y2tsl45p5zjbh2 \
    --encryption-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/ivi5ksfsuplneuyt \
    --primary-account-number 171234567890123 \
    --pin-block-format ISO_FORMAT_0 \
    --generation-attributes VisaPin={PinVerificationKeyIndex=1}
```

輸出：

```
{
    "GenerationKeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/37y2tsl45p5zjbh2",
    "GenerationKeyCheckValue": "7F2363",
    "EncryptionKeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt",
    "EncryptionKeyCheckValue": "7CC9E2",
    "EncryptedPinBlock": "AC17DC148BDA645E",
    "PinData": {
        "VerificationValue": "5507"
    }
}
```

如需詳細資訊，請參閱AWS 《付款密碼編譯使用者指南》中的產生 PIN 資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GeneratePinData。

re-encrypt-data

以下程式碼範例顯示如何使用 re-encrypt-data。

AWS CLI

使用不同的金鑰重新加密資料

下列re-encrypt-data範例會解密使用 AES 對稱金鑰加密的密碼文字，並使用衍生的每筆交易唯一金鑰 (DUKPT) 金鑰重新加密。

```
aws payment-cryptography-data re-encrypt-data \
    --incoming-key-identifier arn:aws:payment-cryptography:us-
west-2:111122223333:key/hyvv7ymboitd4vfy \
    --outgoing-key-identifier arn:aws:payment-cryptography:us-
west-2:111122223333:key/jl6ythkcvzesbxen \
    --cipher-
text 4D2B0BDBA192D5AEFEAA5B3EC28E4A65383C313FFA25140101560F75FE1B99F27192A90980AB9334
 \
    --incoming-encryption-
attributes "Dukpt={Mode=ECB,KeySerialNumber=0123456789111111}" \
    --outgoing-encryption-attributes '{"Symmetric": {"Mode": "ECB"}}'
```

輸出：

```
{
    "CipherText":
 "F94959DA30EEFF0C035483C6067667CF6796E3C1AD28C2B61F9CFEB772A8DD41C0D6822931E0D3B1",
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:111122223333:key/
jl6ythkcvzesbxen",
    "KeyCheckValue": "2E8CD9"
}
```

如需詳細資訊，請參閱AWS 《付款密碼編譯使用者指南》中的加密和解密資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReEncryptData。

translate-pin-data

以下程式碼範例顯示如何使用 translate-pin-data。

AWS CLI

若要翻譯 PIN 資料

下列translate-pin-data範例使用 DUKPT 演算法，將 PIN 從使用 ISO 0 PIN 區塊的 PEK TDES 加密轉譯為 AES ISO 4 PIN 區塊。

```
aws payment-cryptography-data translate-pin-data \
    --encrypted-pin-block "AC17DC148BDA645E" \
    --incoming-translation-
attributes=IsoFormat0='{PrimaryAccountNumber=171234567890123}' \
    --incoming-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/ivi5ksfsuplneuyt \
    --outgoing-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/4pmyquwjs3yj4vwe \
    --outgoing-translation-attributes
 IsoFormat4="{PrimaryAccountNumber=171234567890123}" \
    --outgoing-dukpt-attributes KeySerialNumber="FFFF9876543210E00008"
```

輸出：

```
{
    "PinBlock": "1F4209C670E49F83E75CC72E81B787D9",
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt
    "KeyCheckValue": "7CC9E2"
}
```

如需詳細資訊，請參閱AWS 《付款密碼編譯使用者指南》中的翻譯 PIN 資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TranslatePinData。

verify-auth-request-cryptogram

以下程式碼範例顯示如何使用 verify-auth-request-cryptogram。

AWS CLI

驗證身分驗證請求

下列verify-auth-request-cryptogram範例會驗證授權請求密碼編譯 (ARQC)。

```
aws payment-cryptography-data verify-auth-request-cryptogram \
    --auth-request-cryptogram F6E1BD1E6037FB3E \
    --auth-response-attributes '{"ArpcMethod1": {"AuthResponseCode": "1111"}}' \
    --key-identifier arn:aws:payment-cryptography:us-west-2:111122223333:key/
pboipdfzd4mdklya \
    --major-key-derivation-mode "EMV_OPTION_A" \
```

```
    --session-key-derivation-attributes '{"EmvCommon":
 {"ApplicationTransactionCounter": "1234","PanSequenceNumber":
 "01","PrimaryAccountNumber": "471234567890123"}}' \
    --transaction-data "123456789ABCDEF"
```

輸出：

```
{
    "AuthResponseValue": "D899B8C6FBF971AA",
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:111122223333:key/
pboipdfzd4mdklya",
    "KeyCheckValue": "985792"
}
```

如需詳細資訊，請參閱《付款密碼編譯使用者指南》中的驗證身分驗證請求 (ARQC) 密碼編譯。AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 VerifyAuthRequestCryptogram。

verify-card-validation-data

以下程式碼範例顯示如何使用 verify-card-validation-data。

AWS CLI

驗證 CVV

下列verify-card-validation-data範例會驗證 PAN 的 CVV/CVV2。

```
aws payment-cryptography-data verify-card-validation-data \
    --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi \
    --primary-account-number=171234567890123 \
    --verification-attributes CardVerificationValue2={CardExpiryDate=0123} \
    --validation-data 801
```

輸出：

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
```

```
    "KeyCheckValue": "CADDA1"
}
```

如需詳細資訊，請參閱AWS 《付款密碼編譯使用者指南》中的驗證卡片資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 VerifyCardValidationData。

verify-mac

以下程式碼範例顯示如何使用 verify-mac。

AWS CLI

驗證 MAC

下列verify-mac範例會使用演算法 HMAC_SHA256 和 HMAC 加密金鑰，驗證卡片資料身分驗證的雜湊型訊息驗證碼 (HMAC)。

```
aws payment-cryptography-data verify-mac \
    --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
qnobl5lghrzunce6 \
    --message-
data "3b343038383439303031303733393431353d323430383232363030303730303f33" \
    --verification-attributes='Algorithm=HMAC_SHA256' \
    --mac ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C
```

輸出：

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
qnobl5lghrzunce6,
    "KeyCheckValue": "2976E7",
}
```

如需詳細資訊，請參閱《 AWS 付款密碼編譯使用者指南》中的驗證 MAC。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 VerifyMac。

verify-pin-data

以下程式碼範例顯示如何使用 verify-pin-data。

AWS CLI

驗證 PIN 碼

下列verify-pin-data範例會驗證 PAN 的 PIN。

```
aws payment-cryptography-data verify-pin-data \
    --verification-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/37y2tsl45p5zjbh2 \
    --encryption-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/ivi5ksfsuplneuyt \
    --primary-account-number 171234567890123 \
    --pin-block-format ISO_FORMAT_0 \
    --verification-attributes
 VisaPin="{PinVerificationKeyIndex=1,VerificationValue=5507}" \
    --encrypted-pin-block AC17DC148BDA645E
```

輸出：

```
{
    "VerificationKeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/37y2tsl45p5zjbh2",
    "VerificationKeyCheckValue": "7F2363",
    "EncryptionKeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt",
    "EncryptionKeyCheckValue": "7CC9E2",
}
```

如需詳細資訊，請參閱AWS 《付款密碼編譯使用者指南》中的驗證 PIN 資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 VerifyPinData。

使用 的 Amazon Pinpoint 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon Pinpoint 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-app`

以下程式碼範例顯示如何使用 `create-app`。

AWS CLI

範例 1：建立應用程式

以下 `create-app` 範例會建立新應用程式 (專案)。

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp
```

輸出：

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
}
```

範例 2：建立已標記的應用程式

下列 `create-app` 範例會建立新的應用程式 (專案)，並將標籤 (金鑰和值) 與應用程式產生關聯。

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp,tags={"Stack"="Test"}
```

輸出：

```
{
    "ApplicationResponse": {
```

```
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {
            "Stack": "Test"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateApp。

create-sms-template

以下程式碼範例顯示如何使用 create-sms-template。

AWS CLI

為透過簡訊管道傳送的訊息建立訊息範本

下列create-sms-template範例會建立簡訊範本。

```
aws pinpoint create-sms-template \
    --template-name TestTemplate \
    --sms-template-request file://myfile.json \
    --region us-east-1
```

myfile.json 的內容：

```
{
    "Body": "hello\n how are you?\n food is good",
    "TemplateDescription": "Test SMS Template"
}
```

輸出：

```
{
    "CreateTemplateMessageBody": {
        "Arn": "arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:templates/
TestTemplate/SMS",
        "Message": "Created",
        "RequestID": "8c36b17f-a0b0-400f-ac21-29e9b62a975d"
```

```
    }
}
```

如需詳細資訊，請參閱《Amazon Pinpoint 使用者指南》中的 Amazon Pinpoint 訊息範本。Amazon Pinpoint

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSmsTemplate。

delete-app

以下程式碼範例顯示如何使用 delete-app。

AWS CLI

如欲刪除應用程式

以下 delete-app 範例會刪除應用程式 (專案)。

```
aws pinpoint delete-app \
    --application-id 810c7aab86d42fb2b56c8c966example
```

輸出：

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteApp。

get-apns-channel

以下程式碼範例顯示如何使用 get-apns-channel。

AWS CLI

擷取應用程式 APNs 頻道狀態和設定的相關資訊

下列get-apns-channel範例會擷取應用程式 APNs頻道狀態和設定的相關資訊。

```
aws pinpoint get-apns-channel \
    --application-id 9ab1068eb0a6461c86cce7f27ce0efd7 \
    --region us-east-1
```

輸出：

```
{
    "APNSChannelResponse": {
        "ApplicationId": "9ab1068eb0a6461c86cce7f27ce0efd7",
        "CreationDate": "2019-05-09T21:54:45.082Z",
        "DefaultAuthenticationMethod": "CERTIFICATE",
        "Enabled": true,
        "HasCredential": true,
        "HasTokenKey": false,
        "Id": "apns",
        "IsArchived": false,
        "LastModifiedDate": "2019-05-09T22:04:01.067Z",
        "Platform": "APNS",
        "Version": 2
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetApnsChannel。

get-app

以下程式碼範例顯示如何使用 get-app。

AWS CLI

擷取應用程式 （ 專案 ） 的相關資訊

下列get-app範例會擷取應用程式 （ 專案 ） 的相關資訊。

```
aws pinpoint get-app \
    --application-id 810c7aab86d42fb2b56c8c966example \
    --region us-east-1
```

輸出：

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {
                "Year": "2019",
                "Stack": "Production"
            }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetApp。

get-apps

以下程式碼範例顯示如何使用 get-apps。

AWS CLI

擷取所有應用程式的相關資訊

下列get-apps範例會擷取所有應用程式（專案）的相關資訊。

```
aws pinpoint get-apps
```

輸出：

```
{
    "ApplicationsResponse": {
        "Item": [
            {
                "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
                "Id": "810c7aab86d42fb2b56c8c966example",
                "Name": "ExampleCorp",
                "tags": {
                    "Year": "2019",
                    "Stack": "Production"
                }
            },
```

```
            {
                "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/42d8c7eb0990a57ba1d5476a3example",
                "Id": "42d8c7eb0990a57ba1d5476a3example",
                "Name": "AnyCompany",
                "tags": {}
            },
            {
                "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/80f5c382b638ffe5ad12376bbexample",
                "Id": "80f5c382b638ffe5ad12376bbexample",
                "Name": "ExampleCorp_Test",
                "tags": {
                    "Year": "2019",
                    "Stack": "Test"
                }
            }
        ],
        "NextToken":
 "eyJDcmVhdGlvbkRhdGUiOiIyMDE5LTA3LTE2VDE0OjM4OjUzLjkwM1oiLCJBY2NvdW50SWQiOiI1MTIzOTcxODM4Nz
    }
}
```

NextToken 回應值的存在表示有更多可用的輸出。再次呼叫 命令，並提供該值做為NextToken輸入參數。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetApps。

get-campaign

以下程式碼範例顯示如何使用 get-campaign。

AWS CLI

擷取行銷活動狀態、組態和其他設定的相關資訊

下列get-campaign範例會擷取行銷活動的狀態、組態和其他設定的相關資訊。

```
aws pinpoint get-campaign \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --campaign-id a1e63c6cc0eb43ed826ffcc3cc90b30d \
    --region us-east-1
```

輸出：


```
{
    "CampaignResponse": {
        "AdditionalTreatments": [],
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
        "Arn": "arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/6e0b7591a90841d2b5d93fa11143e5a7/campaigns/
a1e63c6cc0eb43ed826ffcc3cc90b30d",
        "CreationDate": "2019-10-08T18:40:16.581Z",
        "Description": " ",
        "HoldoutPercent": 0,
        "Id": "a1e63c6cc0eb43ed826ffcc3cc90b30d",
        "IsPaused": false,
        "LastModifiedDate": "2019-10-08T18:40:16.581Z",
        "Limits": {
            "Daily": 0,
            "MaximumDuration": 60,
            "MessagesPerSecond": 50,
            "Total": 0
        },
        "MessageConfiguration": {
            "EmailMessage": {
                "FromAddress": "sender@example.com",
                "HtmlBody": "<!DOCTYPE html>\n  <html lang=\"en\">\n    <head>\n
 <meta http-equiv=\"Content-Type\" content=\"text/html; charset=utf-8\" />\n</head>
\n<body>Hello</body>\n</html>",
                "Title": "PinpointDemo"
            }
        },
        "Name": "MyCampaign",
        "Schedule": {
            "IsLocalTime": false,
            "StartTime": "IMMEDIATE",
            "Timezone": "utc"
        },
        "SegmentId": "b66c9e42f71444b2aa2e0ffc1df28f60",
        "SegmentVersion": 1,
        "State": {
            "CampaignStatus": "COMPLETED"
        },
        "tags": {},
        "TemplateConfiguration": {},
        "Version": 1
```

```
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCampaign。

get-campaigns

以下程式碼範例顯示如何使用 get-campaigns。

AWS CLI

擷取與應用程式相關聯之所有行銷活動的狀態、組態和其他設定的相關資訊

下列get-campaigns範例會擷取與應用程式相關聯之所有行銷活動的狀態、組態和其他設定的相關資訊。

```
aws pinpoint get-campaigns \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

輸出：

```
{
    "CampaignsResponse": {
        "Item": [
            {
                "AdditionalTreatments": [],
                "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
                "Arn": "arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/6e0b7591a90841d2b5d93fa11143e5a7/
campaigns/7e1280344c8f4a9aa40a00b006fe44f1",
                "CreationDate": "2019-10-08T18:40:22.905Z",
                "Description": " ",
                "HoldoutPercent": 0,
                "Id": "7e1280344c8f4a9aa40a00b006fe44f1",
                "IsPaused": false,
                "LastModifiedDate": "2019-10-08T18:40:22.905Z",
                "Limits": {},
                "MessageConfiguration": {
                    "EmailMessage": {
                        "FromAddress": "sender@example.com",
```

```
                        "HtmlBody": "<!DOCTYPE html>\n    <html lang=\"en
\">\n    <head>\n    <meta http-equiv=\"Content-Type\" content=\"text/html;
 charset=utf-8\" />\n</head>\n<body>Hello</body>\n</html>",
                        "Title": "PInpointDemo Test"
                    }
                },
                "Name": "MyCampaign1",
                "Schedule": {
                    "IsLocalTime": false,
                    "QuietTime": {},
                    "StartTime": "IMMEDIATE",
                    "Timezone": "UTC"
                },
                "SegmentId": "b66c9e42f71444b2aa2e0ffc1df28f60",
                "SegmentVersion": 1,
                "State": {
                    "CampaignStatus": "COMPLETED"
                },
                "tags": {},
                "TemplateConfiguration": {},
                "Version": 1
            },
            {
                "AdditionalTreatments": [],
                "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
                "Arn": "arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/6e0b7591a90841d2b5d93fa11143e5a7/campaigns/
a1e63c6cc0eb43ed826ffcc3cc90b30d",
                "CreationDate": "2019-10-08T18:40:16.581Z",
                "Description": " ",
                "HoldoutPercent": 0,
                "Id": "a1e63c6cc0eb43ed826ffcc3cc90b30d",
                "IsPaused": false,
                "LastModifiedDate": "2019-10-08T18:40:16.581Z",
                "Limits": {
                    "Daily": 0,
                    "MaximumDuration": 60,
                    "MessagesPerSecond": 50,
                    "Total": 0
                },
                "MessageConfiguration": {
                    "EmailMessage": {
                        "FromAddress": "sender@example.com",
```

```
                        "HtmlBody": "<!DOCTYPE html>\n    <html lang=\"en
\">\n    <head>\n    <meta http-equiv=\"Content-Type\" content=\"text/html;
 charset=utf-8\" />\n</head>\n<body>Demo</body>\n</html>",
                        "Title": "PinpointDemo"
                    }
                },
                "Name": "MyCampaign2",
                "Schedule": {
                    "IsLocalTime": false,
                    "StartTime": "IMMEDIATE",
                    "Timezone": "utc"
                },
                "SegmentId": "b66c9e42f71444b2aa2e0ffc1df28f60",
                "SegmentVersion": 1,
                "State": {
                    "CampaignStatus": "COMPLETED"
                },
                "tags": {},
                "TemplateConfiguration": {},
                "Version": 1
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCampaigns。

get-channels

以下程式碼範例顯示如何使用 get-channels。

AWS CLI

擷取應用程式每個頻道的歷史記錄和狀態的相關資訊

下列get-channels範例會擷取應用程式每個頻道的歷史記錄和狀態的相關資訊。

```
aws pinpoint get-channels \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

輸出：

```
{
    "ChannelsResponse": {
        "Channels": {
            "GCM": {
                "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
                "CreationDate": "2019-10-08T18:28:23.182Z",
                "Enabled": true,
                "HasCredential": true,
                "Id": "gcm",
                "IsArchived": false,
                "LastModifiedDate": "2019-10-08T18:28:23.182Z",
                "Version": 1
            },
            "SMS": {
                "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
                "CreationDate": "2019-10-08T18:39:18.511Z",
                "Enabled": true,
                "Id": "sms",
                "IsArchived": false,
                "LastModifiedDate": "2019-10-08T18:39:18.511Z",
                "Version": 1
            },
            "EMAIL": {
                "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
                "CreationDate": "2019-10-08T18:27:23.990Z",
                "Enabled": true,
                "Id": "email",
                "IsArchived": false,
                "LastModifiedDate": "2019-10-08T18:27:23.990Z",
                "Version": 1
            },
            "IN_APP": {
                "Enabled": true,
                "IsArchived": false,
                "Version": 0
            }
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetChannels。

get-email-channel

以下程式碼範例顯示如何使用 get-email-channel。

AWS CLI

擷取應用程式電子郵件管道狀態和設定的相關資訊

下列get-email-channel範例會擷取應用程式的電子郵件管道狀態和設定。

```
aws pinpoint get-email-channel \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

輸出：

```
{
    "EmailChannelResponse": {
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
        "CreationDate": "2019-10-08T18:27:23.990Z",
        "Enabled": true,
        "FromAddress": "sender@example.com",
        "Id": "email",
        "Identity": "arn:aws:ses:us-east-1:AIDACKCEVSQ6C2EXAMPLE:identity/
sender@example.com",
        "IsArchived": false,
        "LastModifiedDate": "2019-10-08T18:27:23.990Z",
        "MessagesPerSecond": 1,
        "Platform": "EMAIL",
        "RoleArn": "arn:aws:iam::AIDACKCEVSQ6C2EXAMPLE:role/pinpoint-events",
        "Version": 1
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetEmailChannel。

get-endpoint

以下程式碼範例顯示如何使用 get-endpoint。

AWS CLI

如要擷取應用程式之特定端點的設定和屬性相關資訊

下列 get-endpoint 範例會擷取應用程式之特定端點的設定和屬性相關資訊。

```
aws pinpoint get-endpoint \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --endpoint-id testendpoint \
    --region us-east-1
```

輸出：

```
{
    "EndpointResponse": {
        "Address": "+11234567890",
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "Attributes": {},
        "ChannelType": "SMS",
        "CohortId": "63",
        "CreationDate": "2019-01-28T23:55:11.534Z",
        "EffectiveDate": "2021-08-06T00:04:51.763Z",
        "EndpointStatus": "ACTIVE",
        "Id": "testendpoint",
        "Location": {
            "Country": "USA"
        },
        "Metrics": {
            "SmsDelivered": 1.0
        },
        "OptOut": "ALL",
        "RequestId": "a204b1f2-7e26-48a7-9c80-b49a2143489d",
        "User": {
            "UserAttributes": {
                "Age": [
                    "24"
                ]
            },
        "UserId": "testuser"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetEndpoint。

get-gcm-channel

以下程式碼範例顯示如何使用 get-gcm-channel。

AWS CLI

擷取應用程式 GCM 頻道狀態和設定的相關資訊

下列get-gcm-channel範例會擷取應用程式 GCM 頻道狀態和設定的相關資訊。

```
aws pinpoint get-gcm-channel \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

輸出：

```
{
    "GCMChannelResponse": {
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
        "CreationDate": "2019-10-08T18:28:23.182Z",
        "Enabled": true,
        "HasCredential": true,
        "Id": "gcm",
        "IsArchived": false,
        "LastModifiedDate": "2019-10-08T18:28:23.182Z",
        "Platform": "GCM",
        "Version": 1
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetGcmChannel。

get-sms-channel

以下程式碼範例顯示如何使用 get-sms-channel。

AWS CLI

如要擷取應用程式 SMS 管道的狀態和設定相關資訊

下列 get-sms-channel 範例會擷取應用程式 SMS 管道的狀態和設定。

```
aws pinpoint get-sms-channel \
```

```
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

輸出：

```
{
    "SMSChannelResponse": {
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
        "CreationDate": "2019-10-08T18:39:18.511Z",
        "Enabled": true,
        "Id": "sms",
        "IsArchived": false,
        "LastModifiedDate": "2019-10-08T18:39:18.511Z",
        "Platform": "SMS",
        "PromotionalMessagesPerSecond": 20,
        "TransactionalMessagesPerSecond": 20,
        "Version": 1
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetSmsChannel。

get-sms-template

以下程式碼範例顯示如何使用 get-sms-template。

AWS CLI

針對透過簡訊管道傳送的訊息，擷取訊息範本的內容和設定

下列get-sms-template範例會擷取簡訊範本的內容和設定。

```
aws pinpoint get-sms-template \
    --template-name TestTemplate \
    --region us-east-1
```

輸出：

```
{
    "SMSTemplateResponse": {
        "Arn": "arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:templates/
TestTemplate/SMS",
```

```
        "Body": "hello\n how are you?\n food is good",
        "CreationDate": "2023-06-20T21:37:30.124Z",
        "LastModifiedDate": "2023-06-20T21:37:30.124Z",
        "tags": {},
        "TemplateDescription": "Test SMS Template",
        "TemplateName": "TestTemplate",
        "TemplateType": "SMS",
        "Version": "1"
    }
}
```

如需詳細資訊，請參閱《Amazon Pinpoint 使用者指南》中的 Amazon Pinpoint 訊息範本。Amazon Pinpoint

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSmsTemplate。

get-voice-channel

以下程式碼範例顯示如何使用 get-voice-channel。

AWS CLI

擷取應用程式語音頻道狀態和設定的相關資訊

下列get-voice-channel範例會擷取應用程式的語音頻道狀態和設定。

```
aws pinpoint get-voice-channel \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

輸出：

```
{
    "VoiceChannelResponse": {
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
        "CreationDate": "2022-04-28T00:17:03.836Z",
        "Enabled": true,
        "Id": "voice",
        "IsArchived": false,
        "LastModifiedDate": "2022-04-28T00:17:03.836Z",
        "Platform": "VOICE",
        "Version": 1
```

```
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetVoiceChannel。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

擷取資源的標籤清單

下列`list-tags-for-resource`範例會擷取與指定資源相關聯的所有標籤（金鑰名稱和值）。

```
aws pinpoint list-tags-for-resource \
    --resource-arn arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example
```

輸出：

```
{
    "TagsModel": {
        "tags": {
            "Year": "2019",
            "Stack": "Production"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Pinpoint 開發人員指南》中的「標記 Amazon Pinpoint 資源 <https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>'__。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`phone-number-validate`

以下程式碼範例顯示如何使用 `phone-number-validate`。

AWS CLI

擷取電話號碼的相關資訊

以下內容會phone-number-validate擷取電話號碼的相關資訊。

```
aws pinpoint phone-number-validate \
    --number-validate-request PhoneNumber="+12065550142" \
    --region us-east-1
```

輸出：

```
{
    "NumberValidateResponse": {
        "Carrier": "ExampleCorp Mobile",
        "City": "Seattle",
        "CleansedPhoneNumberE164": "+12065550142",
        "CleansedPhoneNumberNational": "2065550142",
        "Country": "United States",
        "CountryCodeIso2": "US",
        "CountryCodeNumeric": "1",
        "OriginalPhoneNumber": "+12065550142",
        "PhoneType": "MOBILE",
        "PhoneTypeCode": 0,
        "Timezone": "America/Los_Angeles",
        "ZipCode": "98101"
    }
}
```

詳情請參閱《Amazon Pinpoint SMS 使用者指南》中的 Amazon Pinpoint SMS 管道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PhoneNumberValidate。

send-messages

以下程式碼範例顯示如何使用 send-messages。

AWS CLI

如要使用應用程式的端點傳送 SMS 訊息

下列 send-messages 範例會針對具有端點的應用程式傳送直接訊息。

```
aws pinpoint send-messages \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
```

```
    --message-request file://myfile.json \
    --region us-west-2
```

myfile.json 的內容：

```
{
    "MessageConfiguration": {
        "SMSMessage": {
            "Body": "hello, how are you?"
        }
    },
    "Endpoints": {
        "testendpoint": {}
    }
}
```

輸出：

```
{
    "MessageResponse": {
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "EndpointResult": {
            "testendpoint": {
                "Address": "+12345678900",
                "DeliveryStatus": "SUCCESSFUL",
                "MessageId": "itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0",
                "StatusCode": 200,
                "StatusMessage": "MessageId:
 itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0"
            }
        },
        "RequestId": "c7e23264-04b2-4a46-b800-d24923f74753"
    }
}
```

詳情請參閱《Amazon Pinpoint SMS 使用者指南》中的 Amazon Pinpoint SMS 管道。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 SendMessages。

send-users-messages

以下程式碼範例顯示如何使用 send-users-messages。

AWS CLI

為應用程式的使用者傳送簡訊

下列send-users-messages範例會為 應用程式的 使用者傳送直接訊息。

```
aws pinpoint send-users-messages \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --send-users-message-request file://myfile.json \
    --region us-west-2
```

myfile.json 的內容：

```
{
    "MessageConfiguration": {
        "SMSMessage": {
            "Body": "hello, how are you?"
        }
    },
    "Users": {
        "testuser": {}
    }
}
```

輸出：

```
{
    "SendUsersMessageResponse": {
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "RequestId": "e0b12cf5-2359-11e9-bb0b-d5fb91876b25",
        "Result": {
            "testuser": {
                "testuserendpoint": {
                    "DeliveryStatus": "SUCCESSFUL",
                    "MessageId": "7qu4hk5bqhda3i7i2n4pjf98qcuh8b7p45ifsmo0",
                    "StatusCode": 200,
                    "StatusMessage": "MessageId:
 7qu4hk5bqhda3i7i2n4pjf98qcuh8b7p45ifsmo0",
                    "Address": "+12345678900"
                }
            }
        }
```

```
    }
}
```

詳情請參閱《Amazon Pinpoint SMS 使用者指南》中的 Amazon Pinpoint SMS 管道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SendUsersMessages。

`tag-resource`

以下程式碼範例顯示如何使用 `tag-resource`。

AWS CLI

將標籤新增至資源

下列範例會將兩個標籤（索引鍵名稱和值）新增至資源。

```
aws pinpoint list-tags-for-resource \
    --resource-arn arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \
    --tags-model tags={Stack=Production,Year=2019}
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Pinpoint 開發人員指南》中的「標記 Amazon Pinpoint 資源 <https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>'__。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

`untag-resource`

以下程式碼範例顯示如何使用 `untag-resource`。

AWS CLI

範例 1：從資源移除標籤

下列`untag-resource`範例會從資源中移除指定的標籤（索引鍵名稱和值）。

```
aws pinpoint untag-resource \
    --resource-arn arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \
```

```
    --tag-keys Year
```

此命令不會產生輸出。

範例 2：從資源中移除多個標籤

下列untag-resource範例會從資源中移除指定的標籤（索引鍵名稱和值）。

```
aws pinpoint untag-resource \
    --resource-arn arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \
    --tag-keys Year Stack
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Pinpoint 開發人員指南》中的「標記 Amazon Pinpoint 資源 <https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>'__。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-sms-channel

以下程式碼範例顯示如何使用 update-sms-channel。

AWS CLI

啟用簡訊管道或更新應用程式的簡訊管道狀態和設定。

下列update-sms-channel範例會為應用程式啟用 SMS 通道的 SMS 通道。

```
aws pinpoint update-sms-channel \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --sms-channel-request Enabled=true \
    --region us-west-2
```

輸出：

```
{
    "SMSChannelResponse": {
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "CreationDate": "2019-01-28T23:25:25.224Z",
```

```
        "Enabled": true,
        "Id": "sms",
        "IsArchived": false,
        "LastModifiedDate": "2023-05-18T23:22:50.977Z",
        "Platform": "SMS",
        "PromotionalMessagesPerSecond": 20,
        "TransactionalMessagesPerSecond": 20,
        "Version": 3
    }
}
```

詳情請參閱《Amazon Pinpoint SMS 使用者指南》中的 Amazon Pinpoint SMS 管道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSmsChannel。

使用 的 Amazon Polly 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon Polly 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`delete-lexicon`

以下程式碼範例顯示如何使用 `delete-lexicon`。

AWS CLI

刪除語彙

下列`delete-lexicon`範例會刪除指定的語彙。

```
aws polly delete-lexicon \
```

```
    --name w3c
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Polly 開發人員指南》中的使用 DeleteLexicon 操作。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLexicon。

get-lexicon

以下程式碼範例顯示如何使用 get-lexicon。

AWS CLI

擷取語彙的內容

下列get-lexicon範例會擷取指定發音語彙的內容。

```
aws polly get-lexicon \
    --name w3c
```

輸出：

```
{
    "Lexicon": {
        "Content": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<lexicon version=
\"1.0\" \n      xmlns=    \"http://www.w3.org/2005/01/pronunciation-lexicon
\"\n      xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" \n
 xsi:schemaLocation=\"http://www.w3.org/2005/01/pronunciation-lexicon \n
 http://www.w3.org/TR/2007/CR-pronunciation-    lexicon-20071212/pls.xsd\"\n
   alphabet=\"ipa\" \n      xml:lang=\"en-US\">\n  <lexeme>\n    <grapheme>W3C</
grapheme>\n        <alias>World Wide Web Consortium</alias>\n  </lexeme>\n</lexicon>
\n",
        "Name": "w3c"
    },
    "LexiconAttributes": {
        "Alphabet": "ipa",
        "LanguageCode": "en-US",
        "LastModified": 1603908910.99,
        "LexiconArn": "arn:aws:polly:us-west-2:880185128111:lexicon/w3c",
        "LexemesCount": 1,
        "Size": 492
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon Polly 開發人員指南》中的使用 GetLexicon 操作。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLexicon。

get-speech-synthesis-task

以下程式碼範例顯示如何使用 get-speech-synthesis-task。

AWS CLI

取得語音合成任務的相關資訊

下列get-speech-synthesis-task範例會擷取指定語音合成任務的相關資訊。

```
aws polly get-speech-synthesis-task \
    --task-id 70b61c0f-57ce-4715-a247-cae8729dcce9
```

輸出：

```
{
    "SynthesisTask": {
        "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",
        "TaskStatus": "completed",
        "OutputUri": "https://s3.us-west-2.amazonaws.com/amzn-s3-demo-
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
        "CreationTime": 1603911042.689,
        "RequestCharacters": 1311,
        "OutputFormat": "mp3",
        "TextType": "text",
        "VoiceId": "Joanna"
    }
}
```

如需詳細資訊，請參閱《Amazon Polly 開發人員指南》中的建立長音訊檔案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSpeechSynthesisTask。

list-lexicons

以下程式碼範例顯示如何使用 list-lexicons。

AWS CLI

列出語彙

下列list-lexicons範例列出您的發音語彙。

```
aws polly list-lexicons
```

輸出：

```
{
    "Lexicons": [
        {
            "Name": "w3c",
            "Attributes": {
                "Alphabet": "ipa",
                "LanguageCode": "en-US",
                "LastModified": 1603908910.99,
                "LexiconArn": "arn:aws:polly:us-east-2:123456789012:lexicon/w3c",
                "LexemesCount": 1,
                "Size": 492
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Polly 開發人員指南》中的使用 ListLexicons 操作。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListLexicons。

list-speech-synthesis-tasks

以下程式碼範例顯示如何使用 list-speech-synthesis-tasks。

AWS CLI

列出您的語音合成任務

下列list-speech-synthesis-tasks範例列出您的語音合成任務。

```
aws polly list-speech-synthesis-tasks
```

輸出：

```
{
    "SynthesisTasks": [
        {
            "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",
            "TaskStatus": "completed",
            "OutputUri": "https://s3.us-west-2.amazonaws.com/amzn-s3-demo-
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
            "CreationTime": 1603911042.689,
            "RequestCharacters": 1311,
            "OutputFormat": "mp3",
            "TextType": "text",
            "VoiceId": "Joanna"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Polly 開發人員指南》中的建立長音訊檔案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSpeechSynthesisTasks。

put-lexicon

以下程式碼範例顯示如何使用 put-lexicon。

AWS CLI

存放語彙

下列put-lexicon範例存放指定的發音語彙。example.pls 檔案會指定 W3C PLS 相容語彙。

```
aws polly put-lexicon \
    --name w3c \
    --content file://example.pls
```

example.pls 的內容

```
{
    <?xml version="1.0" encoding="UTF-8"?>
    <lexicon version="1.0"
        xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
```

```
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
            http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
        alphabet="ipa"
        xml:lang="en-US">
        <lexeme>
            <grapheme>W3C</grapheme>
            <alias>World Wide Web Consortium</alias>
        </lexeme>
    </lexicon>
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Polly 開發人員指南》中的使用 PutLexicon 操作。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutLexicon。

start-speech-synthesis-task

以下程式碼範例顯示如何使用 start-speech-synthesis-task。

AWS CLI

合成文字

下列start-speech-synthesis-task範例會在 中合成文字，並將產生的 MP3 檔案text_file.txt存放在指定的儲存貯體中。

```
aws polly start-speech-synthesis-task \
    --output-format mp3 \
    --output-s3-bucket-name amzn-s3-demo-bucket \
    --text  file://text_file.txt \
    --voice-id Joanna
```

輸出：

```
{
    "SynthesisTask": {
        "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",
        "TaskStatus": "scheduled",
        "OutputUri": "https://s3.us-east-2.amazonaws.com/amzn-s3-demo-
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
```

```
        "CreationTime": 1603911042.689,
        "RequestCharacters": 1311,
        "OutputFormat": "mp3",
        "TextType": "text",
        "VoiceId": "Joanna"
    }
}
```

如需詳細資訊，請參閱《Amazon Polly 開發人員指南》中的建立長音訊檔案。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartSpeechSynthesisTask。

AWS 價格表 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS 價格表。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`describe-services`

以下程式碼範例顯示如何使用 `describe-services`。

AWS CLI

擷取服務中繼資料

此範例會擷取 Amazon EC2 服務程式碼的中繼資料。

命令：

```
aws pricing describe-services --service-code AmazonEC2 --format-version aws_v1 --
max-items 1
```

輸出：


```
{
  "Services": [
      {
          "ServiceCode": "AmazonEC2",
          "AttributeNames": [
              "volumeType",
              "maxIopsvolume",
              "instance",
              "instanceCapacity10xlarge",
              "locationType",
              "instanceFamily",
              "operatingSystem",
              "clockSpeed",
              "LeaseContractLength",
              "ecu",
              "networkPerformance",
              "instanceCapacity8xlarge",
              "group",
              "maxThroughputvolume",
              "gpuMemory",
              "ebsOptimized",
              "elasticGpuType",
              "maxVolumeSize",
              "gpu",
              "processorFeatures",
              "intelAvxAvailable",
              "instanceCapacity4xlarge",
              "servicecode",
              "groupDescription",
              "processorArchitecture",
              "physicalCores",
              "productFamily",
              "enhancedNetworkingSupported",
              "intelTurboAvailable",
              "memory",
              "dedicatedEbsThroughput",
              "vcpu",
              "OfferingClass",
              "instanceCapacityLarge",
              "capacitystatus",
              "termType",
              "storage",
```

```
                "intelAvx2Available",
                "storageMedia",
                "physicalProcessor",
                "provisioned",
                "servicename",
                "PurchaseOption",
                "instanceCapacity18xlarge",
                "instanceType",
                "tenancy",
                "usagetype",
                "normalizationSizeFactor",
                "instanceCapacity2xlarge",
                "instanceCapacity16xlarge",
                "maxIopsBurstPerformance",
                "instanceCapacity12xlarge",
                "instanceCapacity32xlarge",
                "instanceCapacityXlarge",
                "licenseModel",
                "currentGeneration",
                "preInstalledSw",
                "location",
                "instanceCapacity24xlarge",
                "instanceCapacity9xlarge",
                "instanceCapacityMedium",
                "operation"
            ]
        }
    ],
    "FormatVersion": "aws_v1"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeServices。

get-attribute-values

以下程式碼範例顯示如何使用 get-attribute-values。

AWS CLI

擷取屬性值的清單

下列get-attribute-values範例會擷取指定屬性可用的值清單。

```
aws pricing get-attribute-values \
    --service-code AmazonEC2 \
    --attribute-name volumeType \
    --max-items 2
```

輸出：

```
{
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ==",
    "AttributeValues": [
        {
            "Value": "Cold HDD"
        },
        {
            "Value": "General Purpose"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAttributeValues。

get-products

以下程式碼範例顯示如何使用 get-products。

AWS CLI

擷取產品清單

此範例會擷取符合指定條件的產品清單。

命令：

```
aws pricing get-products --filters file://filters.json --format-version aws_v1 --
max-results 1 --service-code AmazonEC2
```

filter.json：

```
        [
  {
    "Type": "TERM_MATCH",
    "Field": "ServiceCode",
```

```
    "Value": "AmazonEC2"
  },
  {
    "Type": "TERM_MATCH",
    "Field": "volumeType",
    "Value": "Provisioned IOPS"
  }
]
```

輸出：

```
{
  "FormatVersion": "aws_v1",
  "NextToken": "WGDY7ko8fQXdlaUZVdasFQ==:RVSagyIFn770XQOzdUIcO9BY6ucBG9itXAZGZF/
zioUzOsUKh6PCcPWaOyPZRiMePb986TeoKYB9l55fw/
CyoMq5ymnGmT1Vj39TljbbAlhcqnVfTmPIilx8Uy5bdDaBYy/e/2Ofw9Edzsykbs8LTBuNbiDQ
+BBds5yeI9AQkUepruKk3aEahFPxJ55kx/zk",
  "PriceList": [
      "{\"product\":{\"productFamily\":\"Storage\",\"attributes\":{\"storageMedia\":
\"SSD-backed\",\"maxThroughputvolume\":\"320 MB/sec\",\"volumeType\":\"Provisioned
 IOPS\",\"maxIopsvolume\":\"20000\",\"servicecode\":\"AmazonEC2\",\"usagetype
\":\"APS1-EBS:VolumeUsage.piops\",\"locationType\":\"AWS Region\",\"location\":
\"Asia Pacific (Singapore)\",\"servicename\":\"Amazon Elastic Compute Cloud\",
\"maxVolumeSize\":\"16 TiB\",\"operation\":\"\"},\"sku\":\"3MKHN58N7RDDVGKJ\"},
\"serviceCode\":\"AmazonEC2\",\"terms\":{\"OnDemand\":{\"3MKHN58N7RDDVGKJ.JRTCKXETXF
\":{\"priceDimensions\":{\"3MKHN58N7RDDVGKJ.JRTCKXETXF.6YS6EN2CT7\":{\"unit\":\"GB-
Mo\",\"endRange\":\"Inf\",\"description\":\"$0.138 per GB-month of Provisioned
 IOPS SSD (io1)  provisioned storage - Asia Pacific (Singapore)\",\"appliesTo
\":[],\"rateCode\":\"3MKHN58N7RDDVGKJ.JRTCKXETXF.6YS6EN2CT7\",\"beginRange\":
\"0\",\"pricePerUnit\":{\"USD\":\"0.1380000000\"}}},\"sku\":\"3MKHN58N7RDDVGKJ
\",\"effectiveDate\":\"2018-08-01T00:00:00Z\",\"offerTermCode\":\"JRTCKXETXF
\",\"termAttributes\":{}}}},\"version\":\"20180808005701\",\"publicationDate\":
\"2018-08-08T00:57:01Z\"}"
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetProducts。

AWS Private CA 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS Private CA。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-certificate-authority-audit-report`

以下程式碼範例顯示如何使用 `create-certificate-authority-audit-report`。

AWS CLI

建立憑證授權單位稽核報告

下列`create-certificate-authority-audit-report`命令會為 ARN 識別的私有 CA 建立稽核報告。

```
aws acm-pca create-certificate-authority-audit-report --certificate-
authority-arn arn:aws:acm-pca:us-east-1:accountid:certificate-
authority/12345678-1234-1234-1234-123456789012 --s3-bucket-name your-bucket-name --
audit-report-response-format JSON
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCertificateAuthorityAuditReport。

`create-certificate-authority`

以下程式碼範例顯示如何使用 `create-certificate-authority`。

AWS CLI

建立私有憑證授權單位

下列`create-certificate-authority`命令會在您的帳戶中建立私有憑證授權單位 AWS 。

```
aws acm-pca create-certificate-authority --certificate-authority-configuration
 file://C:\ca_config.txt --revocation-configuration file://C:\revoke_config.txt --
certificate-authority-type "SUBORDINATE" --idempotency-token 98256344
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCertificateAuthority。

delete-certificate-authority

以下程式碼範例顯示如何使用 delete-certificate-authority。

AWS CLI

刪除私有憑證授權單位

下列delete-certificate-authority命令會刪除 ARN 所識別的憑證授權單位。

```
aws acm-pca delete-certificate-authority --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCertificateAuthority。

describe-certificate-authority-audit-report

以下程式碼範例顯示如何使用 describe-certificate-authority-audit-report。

AWS CLI

描述憑證授權單位的稽核報告

下列describe-certificate-authority-audit-report命令會列出 ARN 所識別 CA 之指定稽核報告的相關資訊。

```
aws acm-pca describe-certificate-authority-audit-report --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/99999999-8888-7777-6666-555555555555 --audit-report-
id 11111111-2222-3333-4444-555555555555
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCertificateAuthorityAuditReport。

describe-certificate-authority

以下程式碼範例顯示如何使用 describe-certificate-authority。

AWS CLI

描述私有憑證授權單位

下列describe-certificate-authority命令會列出 ARN 所識別之私有 CA 的相關資訊。

```
aws acm-pca describe-certificate-authority --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCertificateAuthority。

get-certificate-authority-certificate

以下程式碼範例顯示如何使用 get-certificate-authority-certificate。

AWS CLI

擷取憑證授權單位 (CA) 憑證

下列get-certificate-authority-certificate命令會擷取 ARN 所指定私有 CA 的憑證和憑證鏈。

```
aws acm-pca get-certificate-authority-certificate --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012 --output text
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCertificateAuthorityCertificate。

get-certificate-authority-csr

以下程式碼範例顯示如何使用 get-certificate-authority-csr。

AWS CLI

擷取憑證授權機構的憑證簽署請求

下列get-certificate-authority-csr命令會擷取 ARN 所指定私有 CA 的 CSR。

```
aws acm-pca get-certificate-authority-csr --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012 --output text
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCertificateAuthorityCsr。

get-certificate

以下程式碼範例顯示如何使用 get-certificate。

AWS CLI

擷取發行的憑證

下列get-certificate範例會從指定的私有 CA 擷取憑證。

```
aws acm-pca get-certificate \
    --certificate-authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012 \
    --certificate-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012/
certificate/6707447683a9b7f4055627ffd55cebcc \
    --output text
```

輸出：

```
-----BEGIN CERTIFICATE-----
MIIEDzCCAvegAwIBAgIRAJuJ8f6ZVYL7gG/rS3qvrZMwDQYJKoZIhvcNAQELBQAw
cTELMAkGA1UEBhMCVVMxEzARBgNVBAgMCldhc2hpbmd0b24xEDAOBgNVBAcMB1Nl
    ....certificate body truncated for brevity....
tKCSglgZZrd4FdLw1EkGm+UVXnodwMtJEQyy3oTfZjURPIyyaqskTu/KSS7YDjK0
KQNy73D6LtmdOEbAyq10XiDxqY41lvKHJ1eZrPaBmYNABxU=
-----END CERTIFICATE---- -----BEGIN CERTIFICATE-----
MIIDrzCCApegAwIBAgIRAOskdzLvcj1eShkoyEE693AwDQYJKoZIhvcNAQELBQAw
cTELMAkGA1UEBhMCVVMxEzARBgNVBAgMCldhc2hpbmd0b24xEDAOBgNVBAcMB1Nl
    ...certificate body truncated for brevity....
kdRGB6P2hpxstDOUIwAoCbhoaWwfA4ybJznf+jOQhAziNlRdKQRR8nODWpKt7H9w
dJ5nxsTk/fniJz86Ddtp6n8s82wYdkN3cVffeK72A9aTCOU=
-----END CERTIFICATE-----
```

輸出的第一部分是憑證本身。第二部分是鏈結至根 CA 憑證的憑證鏈。請注意，當您使用 `--output text`選項時，會在兩個憑證部分之間插入TAB字元（這是縮排文字的原因）。如果您想要取得此輸出並使用其他工具剖析憑證，您可能需要移除TAB角色，才能正確處理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCertificate。

`import-certificate-authority-certificate`

以下程式碼範例顯示如何使用 `import-certificate-authority-certificate`。

AWS CLI

將憑證授權單位憑證匯入 ACM PCA

下列`import-certificate-authority-certificate`命令會將 ARN 所指定 CA 的已簽署私有 CA 憑證匯入 ACM PCA。

```
aws acm-pca import-certificate-authority-certificate --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012 --certificate file://C:\ca_cert.pem
 --certificate-chain file://C:\ca_cert_chain.pem
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ImportCertificateAuthorityCertificate。

`issue-certificate`

以下程式碼範例顯示如何使用 `issue-certificate`。

AWS CLI

發行私有憑證

下列`issue-certificate`命令使用 ARN 指定的私有 CA 來發行私有憑證。

```
aws acm-pca issue-certificate --certificate-authority-arn arn:aws:acm-pca:us-
west-2:123456789012:certificate-authority/12345678-1234-1234-1234-123456789012
 --csr file://C:\cert_1.csr --signing-algorithm "SHA256WITHRSA" --validity
 Value=365,Type="DAYS" --idempotency-token 1234
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 IssueCertificate。

list-certificate-authorities

以下程式碼範例顯示如何使用 list-certificate-authorities。

AWS CLI

列出您的私有憑證授權單位

下列list-certificate-authorities命令會列出您帳戶中所有私有 CAs的相關資訊。

```
aws acm-pca list-certificate-authorities --max-results 10
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCertificateAuthorities。

list-tags

以下程式碼範例顯示如何使用 list-tags。

AWS CLI

列出憑證授權單位的標籤

下列list-tags命令會列出與 ARN 所指定私有 CA 相關聯的標籤。

```
aws acm-pca list-tags --certificate-authority-arn arn:aws:acm-pca:us-
west-2:123456789012:certificate-authority/123455678-1234-1234-1234-123456789012 --
max-results 10
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTags。

revoke-certificate

以下程式碼範例顯示如何使用 revoke-certificate。

AWS CLI

撤銷私有憑證

下列revoke-certificate命令會從 ARN 識別的 CA 撤銷私有憑證。

```
aws acm-pca revoke-certificate --certificate-authority-arn arn:aws:acm-pca:us-
west-2:1234567890:certificate-authority/12345678-1234-1234-1234-123456789012 --
```

```
certificate-serial 67:07:44:76:83:a9:b7:f4:05:56:27:ff:d5:5c:eb:cc --revocation-
reason "KEY_COMPROMISE"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RevokeCertificate。

tag-certificate-authority

以下程式碼範例顯示如何使用 tag-certificate-authority。

AWS CLI

將標籤連接至私有憑證授權單位

下列tag-certificate-authority命令會將一或多個標籤連接至您的私有 CA。

```
aws acm-pca tag-certificate-authority --certificate-authority-
arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012 --tags Key=Admin,Value=Alice
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TagCertificateAuthority。

untag-certificate-authority

以下程式碼範例顯示如何使用 untag-certificate-authority。

AWS CLI

從私有憑證授權單位移除一或多個標籤

下列untag-certificate-authority命令會從 ARN 識別的私有 CA 移除標籤。

```
aws acm-pca untag-certificate-authority --certificate-authority-
arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012 --tags Key=Purpose,Value=Website
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UntagCertificateAuthority。

update-certificate-authority

以下程式碼範例顯示如何使用 update-certificate-authority。

AWS CLI

更新私有憑證授權單位的組態

下列update-certificate-authority命令會更新 ARN 所識別之私有 CA 的狀態和組態。

```
aws acm-pca update-certificate-authority --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-1232456789012 --revocation-configuration file://C:
\revoke_config.txt --status "DISABLED"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateCertificateAuthority。

AWS Proton 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS Proton。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`cancel-service-instance-deployment`

以下程式碼範例顯示如何使用 cancel-service-instance-deployment。

AWS CLI

取消服務執行個體部署

下列cancel-service-instance-deployment範例會取消服務執行個體部署。

```
aws proton cancel-service-instance-deployment \
    --service-instance-name "instance-one" \
    --service-name "simple-svc"
```

輸出：

```
{
    "serviceInstance": {
        "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/service-
instance/instance-one",
        "createdAt": "2021-04-02T21:29:59.962000+00:00",
        "deploymentStatus": "CANCELLING",
        "environmentName": "simple-env",
        "lastDeploymentAttemptedAt": "2021-04-02T21:45:15.406000+00:00",
        "lastDeploymentSucceededAt": "2021-04-02T21:38:00.823000+00:00",
        "name": "instance-one",
        "serviceName": "simple-svc",
        "spec": "proton: ServiceSpec\npipeline:\n
 my_sample_pipeline_optional_input: abc\n  my_sample_pipeline_required_input:
 '123'\ninstances:\n- name: my-instance\n  environment: MySimpleEnv
\n  spec:\n    my_sample_service_instance_optional_input: def\n
 my_sample_service_instance_required_input: '456'\n- name: my-other-instance\n
 environment: MySimpleEnv\n  spec:\n    my_sample_service_instance_required_input:
 '789'\n",
        "templateMajorVersion": "1",
        "templateMinorVersion": "1",
        "templateName": "svc-simple"
    }
}
```

如需詳細資訊，請參閱《 AWS Proton 管理員指南》中的更新服務執行個體，或《 AWS Proton 使用者指南》中的更新服務執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelServiceInstanceDeployment。

cancel-service-pipeline-deployment

以下程式碼範例顯示如何使用 cancel-service-pipeline-deployment。

AWS CLI

取消服務管道部署

下列cancel-service-pipeline-deployment範例會取消服務管道部署。

```
aws proton cancel-service-pipeline-deployment \
    --service-name "simple-svc"
```

輸出：

```
{
    "pipeline": {
        "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/pipeline",
        "createdAt": "2021-04-02T21:29:59.962000+00:00",
        "deploymentStatus": "CANCELLING",
        "lastDeploymentAttemptedAt": "2021-04-02T22:02:45.095000+00:00",
        "lastDeploymentSucceededAt": "2021-04-02T21:39:28.991000+00:00",
        "templateMajorVersion": "1",
        "templateMinorVersion": "1",
        "templateName": "svc-simple"
    }
}
```

如需詳細資訊，請參閱《 AWS Proton 管理員指南》中的更新服務管道，或《 AWS Proton 使用者指南》中的更新服務管道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelServicePipelineDeployment。

create-service

以下程式碼範例顯示如何使用 create-service。

AWS CLI

建立服務

下列create-service範例會使用服務管道建立 服務。

```
aws proton create-service \
    --name "MySimpleService" \
    --template-name "fargate-service" \
    --template-major-version "1" \
    --branch-name "mainline" \
    --repository-connection-arn "arn:aws:codestar-connections:region-id:account-
id:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
    --repository-id "myorg/myapp" \
    --spec file://spec.yaml
```

spec.yaml 的內容：

```
proton: ServiceSpec
```

```
pipeline:
    my_sample_pipeline_required_input: "hello"
    my_sample_pipeline_optional_input: "bye"

instances:
    - name: "acme-network-dev"
        environment: "ENV_NAME"
        spec:
            my_sample_service_instance_required_input: "hi"
            my_sample_service_instance_optional_input: "ho"
```

輸出：

```
{
    "service": {
        "arn": "arn:aws:proton:region-id:123456789012:service/MySimpleService",
        "createdAt": "2020-11-18T19:50:27.460000+00:00",
        "lastModifiedAt": "2020-11-18T19:50:27.460000+00:00",
        "name": "MySimpleService",
        "repositoryConnectionArn": "arn:aws:codestar-connections:region-
id:123456789012connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "repositoryId": "myorg/myapp",
        "status": "CREATE_IN_PROGRESS",
        "templateName": "fargate-service"
    }
}
```

如需詳細資訊，請參閱《 AWS Proton 管理員指南》中的建立服務，以及《 AWS Proton 使用者指南》中的建立服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateService。

`delete-service`

以下程式碼範例顯示如何使用 `delete-service`。

AWS CLI

刪除服務

下列`delete-service`範例會刪除服務。

```
aws proton delete-service \
    --name "simple-svc"
```

輸出：

```
{
    "service": {
        "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc",
        "branchName": "mainline",
        "createdAt": "2020-11-28T22:40:50.512000+00:00",
        "description": "Edit by updating description",
        "lastModifiedAt": "2020-11-29T00:30:39.248000+00:00",
        "name": "simple-svc",
        "repositoryConnectionArn": "arn:aws:codestar-connections:region-
id:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "repositoryId": "myorg/myapp",
        "status": "DELETE_IN_PROGRESS",
        "templateName": "fargate-service"
    }
}
```

如需詳細資訊，請參閱《 AWS Proton 管理員指南》中的刪除服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteService。

get-service-instance

以下程式碼範例顯示如何使用 get-service-instance。

AWS CLI

取得服務執行個體詳細資訊

下列get-service-instance範例會取得服務執行個體的詳細資訊。

```
aws proton get-service-instance \
    --name "instance-one" \
    --service-name "simple-svc"
```

輸出：

```
{
```

```
    "serviceInstance": {
        "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/service-
instance/instance-one",
        "createdAt": "2020-11-28T22:40:50.512000+00:00",
        "deploymentStatus": "SUCCEEDED",
        "environmentName": "simple-env",
        "lastDeploymentAttemptedAt": "2020-11-28T22:40:50.512000+00:00",
        "lastDeploymentSucceededAt": "2020-11-28T22:40:50.512000+00:00",
        "name": "instance-one",
        "serviceName": "simple-svc",
        "spec": "proton: ServiceSpec\npipeline:\n
 my_sample_pipeline_optional_input: hello world\n
 my_sample_pipeline_required_input: pipeline up\ninstances:\n- name: instance-one\n
 environment: my-simple-env\n  spec:\n    my_sample_service_instance_optional_input:
 Ola\n    my_sample_service_instance_required_input: Ciao\n",
        "templateMajorVersion": "1",
        "templateMinorVersion": "0",
        "templateName": "svc-simple"
    }
}
```

如需詳細資訊，請參閱《 AWS Proton 管理員指南》中的檢視服務資料或《 AWS Proton 使用者指南》中的檢視服務資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetServiceInstance。

get-service

以下程式碼範例顯示如何使用 get-service。

AWS CLI

取得服務詳細資訊

下列get-service範例會取得服務的詳細資訊。

```
aws proton get-service \
    --name "simple-svc"
```

輸出：

```
{
```

```
    "service": {
        "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc",
        "branchName": "mainline",
        "createdAt": "2020-11-28T22:40:50.512000+00:00",
        "lastModifiedAt": "2020-11-28T22:44:51.207000+00:00",
        "name": "simple-svc",
        "pipeline": {
            "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/
pipeline/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "createdAt": "2020-11-28T22:40:50.512000+00:00",
            "deploymentStatus": "SUCCEEDED",
            "lastDeploymentAttemptedAt": "2020-11-28T22:40:50.512000+00:00",
            "lastDeploymentSucceededAt": "2020-11-28T22:40:50.512000+00:00",
            "spec": "proton: ServiceSpec\npipeline:\n
 my_sample_pipeline_required_input: hello\n  my_sample_pipeline_optional_input:
 bye\ninstances:\n- name: instance-svc-simple\n  environment: my-simple-
env\n  spec:\n    my_sample_service_instance_required_input: hi\n
 my_sample_service_instance_optional_input: ho\n",
            "templateMajorVersion": "1",
            "templateMinorVersion": "1",
            "templateName": "svc-simple"
        },
        "repositoryConnectionArn": "arn:aws:codestar-connections:region-
id:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "repositoryId": "myorg/myapp",
        "spec": "proton: ServiceSpec\npipeline:\n
 my_sample_pipeline_required_input: hello\n  my_sample_pipeline_optional_input:
 bye\ninstances:\n- name: instance-svc-simple\n  environment: my-simple-
env\n  spec:\n    my_sample_service_instance_required_input: hi\n
 my_sample_service_instance_optional_input: ho\n",
        "status": "ACTIVE",
        "templateName": "svc-simple"
    }
}
```

如需詳細資訊，請參閱《 AWS Proton 管理員指南》中的檢視服務資料或《 AWS Proton 使用者指南》中的檢視服務資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetService。

`list-service-instances`

以下程式碼範例顯示如何使用 `list-service-instances`。

AWS CLI

範例 1：列出所有服務執行個體

下列`list-service-instances`範例列出服務執行個體。

```
aws proton list-service-instances
```

輸出：

```
{
    "serviceInstances": [
        {
            "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/
service-instance/instance-one",
            "createdAt": "2020-11-28T22:40:50.512000+00:00",
            "deploymentStatus": "SUCCEEDED",
            "environmentArn": "arn:aws:proton:region-id:123456789012:environment/
simple-env",
            "lastDeploymentAttemptedAt": "2020-11-28T22:40:50.512000+00:00",
            "lastDeploymentSucceededAt": "2020-11-28T22:40:50.512000+00:00",
            "name": "instance-one",
            "serviceName": "simple-svc",
            "templateMajorVersion": "1",
            "templateMinorVersion": "0",
            "templateName": "fargate-service"
        }
    ]
}
```

如需詳細資訊，請參閱《 AWS Proton 管理員指南》中的檢視服務執行個體資料，或《 AWS Proton 使用者指南》中的檢視服務執行個體資料。

範例 2：列出指定的服務執行個體

下列`get-service-instance`範例會取得服務執行個體。

```
aws proton get-service-instance \
    --name "instance-one" \
    --service-name "simple-svc"
```

輸出：

```
{
    "serviceInstance": {
        "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/service-
instance/instance-one",
        "createdAt": "2020-11-28T22:40:50.512000+00:00",
        "deploymentStatus": "SUCCEEDED",
        "environmentName": "simple-env",
        "lastDeploymentAttemptedAt": "2020-11-28T22:40:50.512000+00:00",
        "lastDeploymentSucceededAt": "2020-11-28T22:40:50.512000+00:00",
        "name": "instance-one",
        "serviceName": "simple-svc",
        "spec": "proton: ServiceSpec\npipeline:\n
 my_sample_pipeline_optional_input: hello world\n
 my_sample_pipeline_required_input: pipeline up\ninstances:\n- name: instance-one\n
 environment: my-simple-env\n  spec:\n    my_sample_service_instance_optional_input:
 Ola\n    my_sample_service_instance_required_input: Ciao\n",
        "templateMajorVersion": "1",
        "templateMinorVersion": "0",
        "templateName": "svc-simple"
    }
}
```

如需詳細資訊，請參閱《 AWS Proton 管理員指南》中的檢視服務執行個體資料，或《 AWS Proton 使用者指南》中的檢視服務執行個體資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListServiceInstances。

update-service-instance

以下程式碼範例顯示如何使用 update-service-instance。

AWS CLI

將服務執行個體更新為新的次要版本

下列update-service-instance範例會將服務執行個體更新至其服務範本的新次要版本，以新增名為 "my-other-instance" 的新執行個體與新的必要輸入。

```
aws proton update-service-instance \
    --service-name "simple-svc" \
    --spec "file://service-spec.yaml " \
    --template-major-version "1" \
    --template-minor-version "1" \
```

```
    --deployment-type "MINOR_VERSION" \
    --name "instance-one"
```

service-spec.yaml 的內容：

```
proton: ServiceSpec
pipeline:
    my_sample_pipeline_optional_input: "abc"
    my_sample_pipeline_required_input: "123"
instances:
    - name: "instance-one"
        environment: "simple-env"
        spec:
            my_sample_service_instance_optional_input: "def"
            my_sample_service_instance_required_input: "456"
    - name: "my-other-instance"
        environment: "simple-env"
        spec:
            my_sample_service_instance_required_input: "789"
```

輸出：

```
{
    "serviceInstance": {
        "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/service-
instance/instance-one",
        "createdAt": "2021-04-02T21:29:59.962000+00:00",
        "deploymentStatus": "IN_PROGRESS",
        "environmentName": "arn:aws:proton:region-id:123456789012:environment/
simple-env",
        "lastDeploymentAttemptedAt": "2021-04-02T21:38:00.823000+00:00",
        "lastDeploymentSucceededAt": "2021-04-02T21:29:59.962000+00:00",
        "name": "instance-one",
        "serviceName": "simple-svc",
        "templateMajorVersion": "1",
        "templateMinorVersion": "0",
        "templateName": "svc-simple"
    }
}
```

如需詳細資訊，請參閱《 AWS Proton 管理員指南》中的更新服務執行個體或《 AWS Proton 使用者指南》中的更新服務執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateServiceInstance。

update-service-pipeline

以下程式碼範例顯示如何使用 update-service-pipeline。

AWS CLI

更新服務管道

下列update-service-pipeline範例會將服務管道更新為其服務範本的新次要版本。

```
aws proton update-service-pipeline \
    --service-name "simple-svc" \
    --spec "file://service-spec.yaml" \
    --template-major-version "1" \
    --template-minor-version "1" \
    --deployment-type "MINOR_VERSION"
```

輸出：

```
{
    "pipeline": {
        "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/pipeline/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "createdAt": "2021-04-02T21:29:59.962000+00:00",
        "deploymentStatus": "IN_PROGRESS",
        "lastDeploymentAttemptedAt": "2021-04-02T21:39:28.991000+00:00",
        "lastDeploymentSucceededAt": "2021-04-02T21:29:59.962000+00:00",
        "spec": "proton: ServiceSpec\n\npipeline:\n
 my_sample_pipeline_optional_input: \"abc\"\n  my_sample_pipeline_required_input:
 \"123\"\n\ninstances:\n  - name: \"my-instance\"\n    environment: \"MySimpleEnv
\"\n    spec:\n      my_sample_service_instance_optional_input: \"def
\"\n      my_sample_service_instance_required_input: \"456\"\n  - name:
 \"my-other-instance\"\n    environment: \"MySimpleEnv\"\n    spec:\n
 my_sample_service_instance_required_input: \"789\"\n",
        "templateMajorVersion": "1",
        "templateMinorVersion": "0",
        "templateName": "svc-simple"
    }
}
```

如需詳細資訊，請參閱《 AWS Proton 管理員指南》中的更新服務管道，或《 AWS Proton 使用者指南》中的更新服務管道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateServicePipeline。

update-service

以下程式碼範例顯示如何使用 update-service。

AWS CLI

更新服務

下列update-service範例會編輯服務描述。

```
aws proton update-service \
    --name "MySimpleService" \
    --description "Edit by updating description"
```

輸出：

```
{
    "service": {
        "arn": "arn:aws:proton:region-id:123456789012:service/MySimpleService",
        "branchName": "mainline",
        "createdAt": "2021-03-12T22:39:42.318000+00:00",
        "description": "Edit by updating description",
        "lastModifiedAt": "2021-03-12T22:44:21.975000+00:00",
        "name": "MySimpleService",
        "repositoryConnectionArn": "arn:aws:codestar-connections:region-
id:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "repositoryId": "myorg/myapp",
        "status": "ACTIVE",
        "templateName": "fargate-service"
    }
}
```

如需詳細資訊，請參閱《 AWS Proton 管理員指南》中的編輯服務或《 AWS Proton 使用者指南》中的編輯服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateService。

使用 的 QLDB 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 QLDB 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`cancel-journal-kinesis-stream`

以下程式碼範例顯示如何使用 `cancel-journal-kinesis-stream`。

AWS CLI

取消日誌串流

下列`cancel-journal-kinesis-stream`範例會從總帳取消指定的日誌串流。

```
aws qldb cancel-journal-kinesis-stream \
    --ledger-name myExampleLedger \
    --stream-id 7ISCkqwe4y25YyHLzYUFAf
```

輸出：

```
{
    "StreamId": "7ISCkqwe4y25YyHLzYUFAf"
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的從 Amazon QLDB 串流日誌資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelJournalKinesisStream。

create-ledger

以下程式碼範例顯示如何使用 create-ledger。

AWS CLI

範例 1：建立具有預設屬性的總帳

下列create-ledger範例會建立名為 myExampleLedger和許可模式 的總帳STANDARD。不會指定刪除保護和 AWS KMS 金鑰的選用參數，因此它們會分別預設為 true和 AWS 擁有的 KMS 金鑰。

```
aws qldb create-ledger \
    --name myExampleLedger \
    --permissions-mode STANDARD
```

輸出：

```
{
    "State": "CREATING",
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "DeletionProtection": true,
    "CreationDateTime": 1568839243.951,
    "Name": "myExampleLedger",
    "PermissionsMode": "STANDARD"
}
```

範例 2：建立停用刪除保護的總帳、客戶受管 KMS 金鑰和指定的標籤

下列create-ledger範例會建立名為 myExampleLedger2和許可模式 的總帳STANDARD。刪除保護功能已停用，指定的客戶受管 KMS 金鑰用於靜態加密，而指定的標籤會連接到資源。

```
aws qldb create-ledger \
    --name myExampleLedger2 \
    --permissions-mode STANDARD \
    --no-deletion-protection \
    --kms-key arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111 \
    --tags IsTest=true,Domain=Test
```

輸出：

```
{
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger2",
    "DeletionProtection": false,
    "CreationDateTime": 1568839543.557,
    "State": "CREATING",
    "Name": "myExampleLedger2",
    "PermissionsMode": "STANDARD",
    "KmsKeyArn": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的 Amazon QLDB Ledgers 的基本操作。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLedger。

delete-ledger

以下程式碼範例顯示如何使用 delete-ledger。

AWS CLI

刪除總帳

下列delete-ledger範例會刪除指定的總帳。

```
aws qldb delete-ledger \
    --name myExampleLedger
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的 Amazon QLDB Ledgers 的基本操作。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLedger。

describe-journal-kinesis-stream

以下程式碼範例顯示如何使用 describe-journal-kinesis-stream。

AWS CLI

描述日誌串流

下列describe-journal-kinesis-stream範例顯示來自總帳之指定日誌串流的詳細資訊。

```
aws qldb describe-journal-kinesis-stream \
    --ledger-name myExampleLedger \
    --stream-id 7ISCkqwe4y25YyHLzYUFAf
```

輸出：

```
{
    "Stream": {
        "LedgerName": "myExampleLedger",
        "CreationTime": 1591221984.677,
        "InclusiveStartTime": 1590710400.0,
        "ExclusiveEndTime": 1590796799.0,
        "RoleArn": "arn:aws:iam::123456789012:role/my-kinesis-stream-role",
        "StreamId": "7ISCkqwe4y25YyHLzYUFAf",
        "Arn": "arn:aws:qldb:us-east-1:123456789012:stream/
myExampleLedger/7ISCkqwe4y25YyHLzYUFAf",
        "Status": "ACTIVE",
        "KinesisConfiguration": {
            "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-for-
qldb",
            "AggregationEnabled": true
        },
        "StreamName": "myExampleLedger-stream"
    }
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的從 Amazon QLDB 串流日誌資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeJournalKinesisStream。

describe-journal-s3-export

以下程式碼範例顯示如何使用 describe-journal-s3-export。

AWS CLI

描述日誌匯出任務

下列describe-journal-s3-export範例顯示分類帳中指定匯出任務的詳細資訊。

```
aws qldb describe-journal-s3-export \
    --name myExampleLedger \
    --export-id ADR2ONPKN5LINYGb4dp7yZ
```

輸出：

```
{
    "ExportDescription": {
        "S3ExportConfiguration": {
            "Bucket": "amzn-s3-demo-bucket",
            "Prefix": "ledgerexport1/",
            "EncryptionConfiguration": {
                "ObjectEncryptionType": "SSE_S3"
            }
        },
        "RoleArn": "arn:aws:iam::123456789012:role/my-s3-export-role",
        "Status": "COMPLETED",
        "ExportCreationTime": 1568847801.418,
        "InclusiveStartTime": 1568764800.0,
        "ExclusiveEndTime": 1568847599.0,
        "LedgerName": "myExampleLedger",
        "ExportId": "ADR2ONPKN5LINYGb4dp7yZ"
    }
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的在 Amazon QLDB 中匯出日誌。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeJournalS3Export。

describe-ledger

以下程式碼範例顯示如何使用 describe-ledger。

AWS CLI

描述總帳

下列describe-ledger範例顯示指定總帳的詳細資訊。

```
aws qldb describe-ledger \
```

```
    --name myExampleLedger
```

輸出：

```
{
    "CreationDateTime": 1568839243.951,
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "State": "ACTIVE",
    "Name": "myExampleLedger",
    "DeletionProtection": true,
    "PermissionsMode": "STANDARD",
    "EncryptionDescription": {
        "KmsKeyArn": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "EncryptionStatus": "ENABLED"
    }
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的 Amazon QLDB Ledgers 的基本操作。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLedger。

export-journal-to-s3

以下程式碼範例顯示如何使用 export-journal-to-s3。

AWS CLI

將日誌區塊匯出至 S3

下列export-journal-to-s3範例會在指定日期和時間範圍內，從名為 的總帳建立日誌區塊的匯出任務myExampleLedger。匯出任務會將區塊寫入指定的 Amazon S3 儲存貯體。

```
aws qldb export-journal-to-s3 \
    --name myExampleLedger \
    --inclusive-start-time 2019-09-18T00:00:00Z \
    --exclusive-end-time 2019-09-18T22:59:59Z \
    --role-arn arn:aws:iam::123456789012:role/my-s3-export-role \
    --s3-export-configuration file://my-s3-export-config.json
```

my-s3-export-config.json 的內容：

```
{
    "Bucket": "amzn-s3-demo-bucket",
    "Prefix": "ledgerexport1/",
    "EncryptionConfiguration": {
        "ObjectEncryptionType": "SSE_S3"
    }
}
```

輸出：

```
{
    "ExportId": "ADR2ONPKN5LINYGb4dp7yZ"
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的在 Amazon QLDB 中匯出日誌。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ExportJournalToS3。

get-block

以下程式碼範例顯示如何使用 get-block。

AWS CLI

範例 1：使用輸入檔案取得日誌區塊和驗證

下列get-block範例會請求區塊資料物件，以及來自指定分類帳的證明。請求適用於指定的摘要提示地址和區塊地址。

```
aws qldb get-block \
    --name vehicle-registration \
    --block-address file://myblockaddress.json \
    --digest-tip-address file://mydigesttipaddress.json
```

myblockaddress.json 的內容：

```
{
    "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:100}"
}
```

mydigesttipaddress.json 的內容：

```
{
    "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:123}"
}
```

輸出：

```
{
    "Block": {
        "IonText": "{blockAddress:{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl
\",sequenceNo:100},transactionId:\"FnQeJBAicTX0Ah32ZnVtSX
\",blockTimestamp:2019-09-16T19:37:05.360Z,blockHash:
{{NoChM92yKRuJAb/jeLd1VnYn4DHiWIf071ACfic9uHc=}},entriesHash:
{{l05LOsiKV14SDbuaYnH7uwXzUvqzIwUiRLXGbTyj/nY=}},previousBlockHash:
{{7kewBXhpdbClcZKxhVmpoMHpUGOJtWQD0iY2LPfZkYA=}},entriesHashList:
[{{eRSwnmAM7WWANWDd5iGOyK+T4tDXyzUq6HZ/0fgLHos=}},{{mHVex/
yjHAWjFPpwhBuH2GKXmKJjK2FBa9faqoUVNtg=}},
{{y5cCBr7pOAIUfsVQ1j0TqtE97b4b4oo1R0vnYyE5wWM=}},{{TvTXygML1bMe6NvEZtGkX
+KR+W/EJl4qD1mmV77KZQg=}}],transactionInfo:{statements:[{statement:
\"FROM VehicleRegistration AS r \\nWHERE r.VIN = '1N4AL11D75C109151'\
\nINSERT INTO r.Owners.SecondaryOwners\\n    VALUE { 'PersonId' :
 'CMVdR77XP8zAglmmFDGTvt' }\",startTime:2019-09-16T19:37:05.302Z,statementDigest:
{{jcgPX2vsOJ0waum4qmDYtn1pCAT9xKNIzA+2k4R+mxA=}}}],documents:
{JUJgkIcNbhS2goq8RqLuZ4:{tableName:\"VehicleRegistration\",tableId:
\"BFJKdXgzt9oF4wjMbuxy4G\",statements:[0]}}},revisions:[{blockAddress:
{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:100},hash:
{{mHVex/yjHAWjFPpwhBuH2GKXmKJjK2FBa9faqoUVNtg=}},data:{VIN:
\"1N4AL11D75C109151\",LicensePlateNumber:\"LEWISR261LL\",State:\"WA
\",PendingPenaltyTicketAmount:90.25,ValidFromDate:2017-08-21,ValidToDate:2020-05-11,Owners:
{PrimaryOwner:{PersonId:\"BFJKdXhnLRT27sXBnojNGW\"},SecondaryOwners:
[{PersonId:\"CMVdR77XP8zAglmmFDGTvt\"}]},City:\"Everett\"},metadata:{id:
\"JUJgkIcNbhS2goq8RqLuZ4\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:
\"FnQeJBAicTX0Ah32ZnVtSX\"}}]}"
    },
    "Proof": {
        "IonText": "[{{l3+EXs69K1+rehlqyWLkt+oHDlw4Zi9pCLW/t/mgTPM=}},
{{48CXG3ehPqsxCYd34EEa8Fso0ORpWWAO8010RJKf3Do=}},{{9UnwnKSQT0i3ge1JMVa
+tMIqCEDaOPTkWxmyHSn8UPQ=}},{{3nW6Vryghk+7pd6wFCtLufgPM6qXHyTNeCb1sCwcDaI=}},
{{Irb5fNhBrNEQ1VPhzlnGT/ZQPadSmgfdtMYcwkNOxoI=}},{{+3CWpYG/ytf/
vq9GidpzSx6JJiLXt1hMQWNnqOy3jfY=}},{{NPx6cRhwsiy5m9UEWS5JTJrZoUdO2jBOAAOmyZAT
+qE=}}]"
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的 Amazon QLDB 中的資料驗證。

範例 2：使用速記語法取得日誌區塊和驗證

下列get-block範例使用速記語法，向指定的總帳請求區塊資料物件和證明。請求適用於指定的摘要提示地址和區塊地址。

```
aws qldb get-block \
    --name vehicle-registration \
    --block-address 'IonText="{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:100}"'
 \
    --digest-tip-address 'IonText="{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl
\",sequenceNo:123}"'
```

輸出：

```
{
    "Block": {
        "IonText": "{blockAddress:{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl
\",sequenceNo:100},transactionId:\"FnQeJBAicTX0Ah32ZnVtSX
\",blockTimestamp:2019-09-16T19:37:05.360Z,blockHash:
{{NoChM92yKRuJAb/jeLd1VnYn4DHiWIf071ACfic9uHc=}},entriesHash:
{{l05LOsiKV14SDbuaYnH7uwXzUvqzIwUiRLXGbTyj/nY=}},previousBlockHash:
{{7kewBXhpdbClcZKxhVmpoMHpUGOJtWQD0iY2LPfZkYA=}},entriesHashList:
[{{eRSwnmAM7WWANWDd5iGOyK+T4tDXyzUq6HZ/0fgLHos=}},{{mHVex/
yjHAWjFPpwhBuH2GKXmKJjK2FBa9faqoUVNtg=}},
{{y5cCBr7pOAIUfsVQ1j0TqtE97b4b4oo1R0vnYyE5wWM=}},{{TvTXygML1bMe6NvEZtGkX
+KR+W/EJl4qD1mmV77KZQg=}}],transactionInfo:{statements:[{statement:
\"FROM VehicleRegistration AS r \\nWHERE r.VIN = '1N4AL11D75C109151'\
\nINSERT INTO r.Owners.SecondaryOwners\\n    VALUE { 'PersonId' :
 'CMVdR77XP8zAglmmFDGTvt' }\",startTime:2019-09-16T19:37:05.302Z,statementDigest:
{{jcgPX2vsOJ0waum4qmDYtn1pCAT9xKNIzA+2k4R+mxA=}}}],documents:
{JUJgkIcNbhS2goq8RqLuZ4:{tableName:\"VehicleRegistration\",tableId:
\"BFJKdXgzt9oF4wjMbuxy4G\",statements:[0]}}},revisions:[{blockAddress:
{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:100},hash:
{{mHVex/yjHAWjFPpwhBuH2GKXmKJjK2FBa9faqoUVNtg=}},data:{VIN:
\"1N4AL11D75C109151\",LicensePlateNumber:\"LEWISR261LL\",State:\"WA
\",PendingPenaltyTicketAmount:90.25,ValidFromDate:2017-08-21,ValidToDate:2020-05-11,Owners:
{PrimaryOwner:{PersonId:\"BFJKdXhnLRT27sXBnojNGW\"},SecondaryOwners:
[{PersonId:\"CMVdR77XP8zAglmmFDGTvt\"}]},City:\"Everett\"},metadata:{id:
```

```
\"JUJgkIcNbhS2goq8RqLuZ4\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:
\"FnQeJBAicTX0Ah32ZnVtSX\"}}]}"
    },
    "Proof": {
        "IonText": "[{{l3+EXs69K1+rehlqyWLkt+oHDlw4Zi9pCLW/t/mgTPM=}},
{{48CXG3ehPqsxCYd34EEa8Fso0ORpWWAO8010RJKf3Do=}},{{9UnwnKSQT0i3ge1JMVa
+tMIqCEDaOPTkWxmyHSn8UPQ=}},{{3nW6Vryghk+7pd6wFCtLufgPM6qXHyTNeCb1sCwcDaI=}},
{{Irb5fNhBrNEQ1VPhzlnGT/ZQPadSmgfdtMYcwkNOxoI=}},{{+3CWpYG/ytf/
vq9GidpzSx6JJiLXt1hMQWNnqOy3jfY=}},{{NPx6cRhwsiy5m9UEWS5JTJrZoUdO2jBOAAOmyZAT
+qE=}}]"
    }
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的 Amazon QLDB 中的資料驗證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBlock。

get-digest

以下程式碼範例顯示如何使用 get-digest。

AWS CLI

取得總帳的摘要

下列get-digest範例會在日誌中最晚遞交的區塊請求來自指定總帳的摘要。

```
aws qldb get-digest \
    --name vehicle-registration
```

輸出：

```
{
    "Digest": "6m6BMXobbJKpMhahwVthAEsN6awgnHK62Qq5McGP1Gk=",
    "DigestTipAddress": {
        "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:123}"
    }
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的 Amazon QLDB 中的資料驗證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDigest。

get-revision

以下程式碼範例顯示如何使用 get-revision。

AWS CLI

範例 1：使用輸入檔案取得文件修訂和驗證

下列get-revision範例會請求修訂資料物件，以及來自指定分類帳的證明。請求適用於指定的摘要提示地址、文件 ID 和修訂的區塊地址。

```
aws qldb get-revision \
    --name vehicle-registration \
    --block-address file://myblockaddress.json \
    --document-id JUJgkIcNbhS2goq8RqLuZ4 \
    --digest-tip-address file://mydigesttipaddress.json
```

myblockaddress.json 的內容：

```
{
    "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:100}"
}
```

mydigesttipaddress.json 的內容：

```
{
    "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:123}"
}
```

輸出：

```
{
    "Revision": {
        "IonText": "{blockAddress:{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl
\",sequenceNo:100},hash:{{mHVex/yjHAWjFPpwhBuH2GKXmKJjK2FBa9faqoUVNtg=}},data:
{VIN:\"1N4AL11D75C109151\",LicensePlateNumber:\"LEWISR261LL\",State:\"WA
\",PendingPenaltyTicketAmount:90.25,ValidFromDate:2017-08-21,ValidToDate:2020-05-11,Owners:
{PrimaryOwner:{PersonId:\"BFJKdXhnLRT27sXBnojNGW\"},SecondaryOwners:
[{PersonId:\"CMVdR77XP8zAglmmFDGTvt\"}]},City:\"Everett\"},metadata:{id:
\"JUJgkIcNbhS2goq8RqLuZ4\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:
\"FnQeJBAicTX0Ah32ZnVtSX\"}}"
```

```
    },
    "Proof": {
        "IonText": "[{{eRSwnmAM7WWANWDd5iGOyK+T4tDXyzUq6HZ/0fgLHos=}},{{VV1rdaNuf
+yJZVGlmsM6gr2T52QvBO8Lg+KgpjcnWAU=}},
{{7kewBXhpdbClcZKxhVmpoMHpUGOJtWQD0iY2LPfZkYA=}},{{l3+EXs69K1+rehlqyWLkt
+oHDlw4Zi9pCLW/t/mgTPM=}},{{48CXG3ehPqsxCYd34EEa8Fso0ORpWWAO8010RJKf3Do=}},
{{9UnwnKSQT0i3ge1JMVa+tMIqCEDaOPTkWxmyHSn8UPQ=}},{{3nW6Vryghk
+7pd6wFCtLufgPM6qXHyTNeCb1sCwcDaI=}},{{Irb5fNhBrNEQ1VPhzlnGT/
ZQPadSmgfdtMYcwkNOxoI=}},{{+3CWpYG/ytf/vq9GidpzSx6JJiLXt1hMQWNnqOy3jfY=}},
{{NPx6cRhwsiy5m9UEWS5JTJrZoUdO2jBOAAOmyZAT+qE=}}]"
    }
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的 Amazon QLDB 中的資料驗證。

範例 2：使用速記語法取得文件修訂和驗證

下列get-revision範例會使用速記語法，向指定的總帳請求修訂資料物件和證明。請求適用於指定的摘要提示地址、文件 ID 和修訂的區塊地址。

```
aws qldb get-revision \
    --name vehicle-registration \
    --block-address 'IonText="{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:100}"'
 \
    --document-id JUJgkIcNbhS2goq8RqLuZ4 \
    --digest-tip-address 'IonText="{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl
\",sequenceNo:123}"'
```

輸出：

```
{
    "Revision": {
        "IonText": "{blockAddress:{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl
\",sequenceNo:100},hash:{{mHVex/yjHAWjFPpwhBuH2GKXmKJjK2FBa9faqoUVNtg=}},data:
{VIN:\"1N4AL11D75C109151\",LicensePlateNumber:\"LEWISR261LL\",State:\"WA
\",PendingPenaltyTicketAmount:90.25,ValidFromDate:2017-08-21,ValidToDate:2020-05-11,Owners:
{PrimaryOwner:{PersonId:\"BFJKdXhnLRT27sXBnojNGW\"},SecondaryOwners:
[{PersonId:\"CMVdR77XP8zAglmmFDGTvt\"}]},City:\"Everett\"},metadata:{id:
\"JUJgkIcNbhS2goq8RqLuZ4\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:
\"FnQeJBAicTX0Ah32ZnVtSX\"}}"
    },
    "Proof": {
```

```
        "IonText": "[{{eRSwnmAM7WWANWDd5iGOyK+T4tDXyzUq6HZ/0fgLHos=}},{{VV1rdaNuf
+yJZVGlmsM6gr2T52QvBO8Lg+KgpjcnWAU=}},
{{7kewBXhpdbClcZKxhVmpoMHpUGOJtWQD0iY2LPfZkYA=}},{{l3+EXs69K1+rehlqyWLkt
+oHDlw4Zi9pCLW/t/mgTPM=}},{{48CXG3ehPqsxCYd34EEa8Fso0ORpWWAO8010RJKf3Do=}},
{{9UnwnKSQT0i3ge1JMVa+tMIqCEDaOPTkWxmyHSn8UPQ=}},{{3nW6Vryghk
+7pd6wFCtLufgPM6qXHyTNeCb1sCwcDaI=}},{{Irb5fNhBrNEQ1VPhzlnGT/
ZQPadSmgfdtMYcwkNOxoI=}},{{+3CWpYG/ytf/vq9GidpzSx6JJiLXt1hMQWNnqOy3jfY=}},
{{NPx6cRhwsiy5m9UEWS5JTJrZoUdO2jBOAAOmyZAT+qE=}}]"
    }
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的 Amazon QLDB 中的資料驗證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRevision。

list-journal-kinesis-streams-for-ledger

以下程式碼範例顯示如何使用 list-journal-kinesis-streams-for-ledger。

AWS CLI

列出總帳的日誌串流

下列list-journal-kinesis-streams-for-ledger範例列出指定總帳的日誌串流。

```
aws qldb list-journal-kinesis-streams-for-ledger \
    --ledger-name myExampleLedger
```

輸出：

```
{
    "Streams": [
        {
            "LedgerName": "myExampleLedger",
            "CreationTime": 1591221984.677,
            "InclusiveStartTime": 1590710400.0,
            "ExclusiveEndTime": 1590796799.0,
            "RoleArn": "arn:aws:iam::123456789012:role/my-kinesis-stream-role",
            "StreamId": "7ISCkqwe4y25YyHLzYUFAf",
            "Arn": "arn:aws:qldb:us-east-1:123456789012:stream/
myExampleLedger/7ISCkqwe4y25YyHLzYUFAf",
            "Status": "ACTIVE",
```

```
            "KinesisConfiguration": {
                "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-
for-qldb",
                "AggregationEnabled": true
            },
            "StreamName": "myExampleLedger-stream"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的從 Amazon QLDB 串流日誌資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListJournalKinesisStreamsForLedger。

`list-journal-s3-exports-for-ledger`

以下程式碼範例顯示如何使用 `list-journal-s3-exports-for-ledger`。

AWS CLI

列出總帳的日誌匯出任務

下列`list-journal-s3-exports-for-ledger`範例列出指定總帳的日誌匯出任務。

```
aws qldb list-journal-s3-exports-for-ledger \
    --name myExampleLedger
```

輸出：

```
{
    "JournalS3Exports": [
        {
            "LedgerName": "myExampleLedger",
            "ExclusiveEndTime": 1568847599.0,
            "ExportCreationTime": 1568847801.418,
            "S3ExportConfiguration": {
                "Bucket": "amzn-s3-demo-bucket",
                "Prefix": "ledgerexport1/",
                "EncryptionConfiguration": {
                    "ObjectEncryptionType": "SSE_S3"
                }
            },
```

```
            "ExportId": "ADR2ONPKN5LINYGb4dp7yZ",
            "RoleArn": "arn:aws:iam::123456789012:role/qldb-s3-export",
            "InclusiveStartTime": 1568764800.0,
            "Status": "IN_PROGRESS"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的在 Amazon QLDB 中匯出日誌。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListJournalS3ExportsForLedger。

list-journal-s3-exports

以下程式碼範例顯示如何使用 `list-journal-s3-exports`。

AWS CLI

列出日誌匯出任務

下列`list-journal-s3-exports`範例列出與目前 AWS 帳戶和區域相關聯的所有總帳的日誌匯出任務。

```
aws qldb list-journal-s3-exports
```

輸出：

```
{
    "JournalS3Exports": [
        {
            "Status": "IN_PROGRESS",
            "LedgerName": "myExampleLedger",
            "S3ExportConfiguration": {
                "EncryptionConfiguration": {
                    "ObjectEncryptionType": "SSE_S3"
                },
                "Bucket": "amzn-s3-demo-bucket",
                "Prefix": "ledgerexport1/"
            },
            "RoleArn": "arn:aws:iam::123456789012:role/my-s3-export-role",
            "ExportCreationTime": 1568847801.418,
            "ExportId": "ADR2ONPKN5LINYGb4dp7yZ",
```

```
            "InclusiveStartTime": 1568764800.0,
            "ExclusiveEndTime": 1568847599.0
        },
        {
            "Status": "COMPLETED",
            "LedgerName": "myExampleLedger2",
            "S3ExportConfiguration": {
                "EncryptionConfiguration": {
                    "ObjectEncryptionType": "SSE_S3"
                },
                "Bucket": "amzn-s3-demo-bucket",
                "Prefix": "ledgerexport1/"
            },
            "RoleArn": "arn:aws:iam::123456789012:role/my-s3-export-role",
            "ExportCreationTime": 1568846847.638,
            "ExportId": "2pdvW8UQrjBAiYTMehEJDI",
            "InclusiveStartTime": 1568592000.0,
            "ExclusiveEndTime": 1568764800.0
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的在 Amazon QLDB 中匯出日誌。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListJournalS3Exports。

list-ledgers

以下程式碼範例顯示如何使用 `list-ledgers`。

AWS CLI

列出可用的總帳

下列`list-ledgers`範例列出與目前 AWS 帳戶和區域相關聯的所有總帳。

```
aws qldb list-ledgers
```

輸出：

```
{
    "Ledgers": [
```

```
        {
            "State": "ACTIVE",
            "CreationDateTime": 1568839243.951,
            "Name": "myExampleLedger"
        },
        {
            "State": "ACTIVE",
            "CreationDateTime": 1568839543.557,
            "Name": "myExampleLedger2"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的 Amazon QLDB Ledgers 的基本操作。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListLedgers。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出連接到總帳的標籤

下列`list-tags-for-resource`範例列出連接至指定總帳的所有標籤。

```
aws qldb list-tags-for-resource \
    --resource-arn arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger
```

輸出：

```
{
    "Tags": {
        "IsTest": "true",
        "Domain": "Test"
    }
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的標記 Amazon QLDB 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`stream-journal-to-kinesis`

以下程式碼範例顯示如何使用 `stream-journal-to-kinesis`。

AWS CLI

範例 1：使用輸入檔案將日誌資料串流至 Kinesis Data Streams

下列`stream-journal-to-kinesis`範例會從名為 的分類帳建立指定日期和時間範圍內的日誌資料串流`myExampleLedger`。串流會將資料傳送至指定的 Amazon Kinesis 資料串流。

```
aws qldb stream-journal-to-kinesis \
    --ledger-name myExampleLedger \
    --inclusive-start-time 2020-05-29T00:00:00Z \
    --exclusive-end-time 2020-05-29T23:59:59Z \
    --role-arn arn:aws:iam::123456789012:role/my-kinesis-stream-role \
    --kinesis-configuration file://my-kinesis-config.json \
    --stream-name myExampleLedger-stream
```

`my-kinesis-config.json` 的內容：

```
{
    "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-for-qldb",
    "AggregationEnabled": true
}
```

輸出：

```
{
    "StreamId": "7ISCkqwe4y25YyHLzYUFAf"
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的從 Amazon QLDB 串流日誌資料。

範例 2：使用速記語法將日誌資料串流至 Kinesis Data Streams

下列`stream-journal-to-kinesis`範例會從名為 的分類帳建立指定日期和時間範圍內的日誌資料串流`myExampleLedger`。串流會將資料傳送至指定的 Amazon Kinesis 資料串流。

```
aws qldb stream-journal-to-kinesis \
    --ledger-name myExampleLedger \
    --inclusive-start-time 2020-05-29T00:00:00Z \
    --exclusive-end-time 2020-05-29T23:59:59Z \
    --role-arn arn:aws:iam::123456789012:role/my-kinesis-stream-role \
    --stream-name myExampleLedger-stream \
    --kinesis-configuration StreamArn=arn:aws:kinesis:us-east-1:123456789012:stream/
stream-for-qldb,AggregationEnabled=true
```

輸出：

```
{
    "StreamId": "7ISCkqwe4y25YyHLzYUFAf"
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的從 Amazon QLDB 串流日誌資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StreamJournalToKinesis。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

標記總帳

下列tag-resource範例會將一組標籤新增至指定的總帳。

```
aws qldb tag-resource \
    --resource-arn arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger \
    --tags IsTest=true,Domain=Test
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的標記 Amazon QLDB 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源移除標籤

下列untag-resource範例會從指定的總帳移除具有指定標籤索引鍵的標籤。

```
aws qldb untag-resource \
    --resource-arn arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger \
    --tag-keys IsTest Domain
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的標記 Amazon QLDB 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-ledger-permissions-mode

以下程式碼範例顯示如何使用 update-ledger-permissions-mode。

AWS CLI

範例 1：將總帳的許可模式更新為 STANDARD

下列update-ledger-permissions-mode範例會將STANDARD許可模式指派給指定的總帳。

```
aws qldb update-ledger-permissions-mode \
    --name myExampleLedger \
    --permissions-mode STANDARD
```

輸出：

```
{
    "Name": "myExampleLedger",
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "PermissionsMode": "STANDARD"
}
```

範例 2：將總帳的許可模式更新為 ALLOW_ALL

下列update-ledger-permissions-mode範例會將ALLOW_ALL許可模式指派給指定的總帳。

```
aws qldb update-ledger-permissions-mode \
    --name myExampleLedger \
    --permissions-mode ALLOW_ALL
```

輸出：

```
{
    "Name": "myExampleLedger",
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "PermissionsMode": "ALLOW_ALL"
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的 Amazon QLDB Ledgers 的基本操作。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateLedgerPermissionsMode。

update-ledger

以下程式碼範例顯示如何使用 update-ledger。

AWS CLI

範例 1：更新總帳的刪除保護屬性

下列update-ledger範例會更新指定的總帳，以停用刪除保護功能。

```
aws qldb update-ledger \
    --name myExampleLedger \
    --no-deletion-protection
```

輸出：

```
{
    "CreationDateTime": 1568839243.951,
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "DeletionProtection": false,
    "Name": "myExampleLedger",
    "State": "ACTIVE"
}
```

範例 2：將總帳的 AWS KMS 金鑰更新為客戶受管金鑰

下列update-ledger範例會更新指定的總帳，以使用客戶受管 KMS 金鑰進行靜態加密。

```
aws qldb update-ledger \
    --name myExampleLedger \
    --kms-key arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111
```

輸出：

```
{
    "CreationDateTime": 1568839243.951,
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "DeletionProtection": false,
    "Name": "myExampleLedger",
    "State": "ACTIVE",
    "EncryptionDescription": {
        "KmsKeyArn": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "EncryptionStatus": "UPDATING"
    }
}
```

範例 3：將總帳的 AWS KMS 金鑰更新為 AWS 擁有的金鑰

下列update-ledger範例會更新指定的總帳，以使用 AWS 擁有的 KMS 金鑰進行靜態加密。

```
aws qldb update-ledger \
    --name myExampleLedger \
    --kms-key AWS_OWNED_KMS_KEY
```

輸出：

```
{
    "CreationDateTime": 1568839243.951,
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "DeletionProtection": false,
    "Name": "myExampleLedger",
    "State": "ACTIVE",
    "EncryptionDescription": {
```

```
        "KmsKeyArn": "AWS_OWNED_KMS_KEY",
        "EncryptionStatus": "UPDATING"
    }
}
```

如需詳細資訊，請參閱《Amazon QLDB 開發人員指南》中的 Amazon QLDB Ledgers 的基本操作。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateLedger。

使用 的 Amazon RDS 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon RDS 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`add-option-to-option-group`

以下程式碼範例顯示如何使用 `add-option-to-option-group`。

AWS CLI

將選項新增至選項群組

下列`add-option-to-option-group`範例會將 選項新增至指定的選項群組。

```
aws rds add-option-to-option-group \
    --option-group-name myoptiongroup \
    --options OptionName=OEM,Port=5500,DBSecurityGroupMemberships=default \
    --apply-immediately
```

輸出：


```
{
    "OptionGroup": {
        "OptionGroupName": "myoptiongroup",
        "OptionGroupDescription": "Test Option Group",
        "EngineName": "oracle-ee",
        "MajorEngineVersion": "12.1",
        "Options": [
            {
                "OptionName": "Timezone",
                "OptionDescription": "Change time zone",
                "Persistent": true,
                "Permanent": false,
                "OptionSettings": [
                    {
                        "Name": "TIME_ZONE",
                        "Value": "Australia/Sydney",
                        "DefaultValue": "UTC",
                        "Description": "Specifies the timezone the user wants to
 change the system time to",
                        "ApplyType": "DYNAMIC",
                        "DataType": "STRING",
                        "AllowedValues": "Africa/Cairo,Africa/Casablanca,Africa/
Harare,Africa/Lagos,Africa/Luanda,Africa/Monrovia,Africa/Nairobi,Africa/
Tripoli,Africa/Windhoek,America/Araguaina,America/Argentina/Buenos_Aires,America/
Asuncion,America/Bogota,America/Caracas,America/Chicago,America/Chihuahua,America/
Cuiaba,America/Denver,America/Detroit,America/Fortaleza,America/Godthab,America/
Guatemala,America/Halifax,America/Lima,America/Los_Angeles,America/Manaus,America/
Matamoros,America/Mexico_City,America/Monterrey,America/Montevideo,America/
New_York,America/Phoenix,America/Santiago,America/Sao_Paulo,America/Tijuana,America/
Toronto,Asia/Amman,Asia/Ashgabat,Asia/Baghdad,Asia/Baku,Asia/Bangkok,Asia/
Beirut,Asia/Calcutta,Asia/Damascus,Asia/Dhaka,Asia/Hong_Kong,Asia/Irkutsk,Asia/
Jakarta,Asia/Jerusalem,Asia/Kabul,Asia/Karachi,Asia/Kathmandu,Asia/Kolkata,Asia/
Krasnoyarsk,Asia/Magadan,Asia/Manila,Asia/Muscat,Asia/Novosibirsk,Asia/Rangoon,Asia/
Riyadh,Asia/Seoul,Asia/Shanghai,Asia/Singapore,Asia/Taipei,Asia/Tehran,Asia/
Tokyo,Asia/Ulaanbaatar,Asia/Vladivostok,Asia/Yakutsk,Asia/Yerevan,Atlantic/
Azores,Atlantic/Cape_Verde,Australia/Adelaide,Australia/Brisbane,Australia/
Darwin,Australia/Eucla,Australia/Hobart,Australia/Lord_Howe,Australia/
Perth,Australia/Sydney,Brazil/DeNoronha,Brazil/East,Canada/Newfoundland,Canada/
Saskatchewan,Etc/GMT-3,Europe/Amsterdam,Europe/Athens,Europe/Berlin,Europe/
Dublin,Europe/Helsinki,Europe/Kaliningrad,Europe/London,Europe/Madrid,Europe/
Moscow,Europe/Paris,Europe/Prague,Europe/Rome,Europe/Sarajevo,Pacific/Apia,Pacific/
Auckland,Pacific/Chatham,Pacific/Fiji,Pacific/Guam,Pacific/Honolulu,Pacific/
```

```
Kiritimati,Pacific/Marquesas,Pacific/Samoa,Pacific/Tongatapu,Pacific/Wake,US/
Alaska,US/Central,US/East-Indiana,US/Eastern,US/Pacific,UTC",
                        "IsModifiable": true,
                        "IsCollection": false
                    }
                ],
                "DBSecurityGroupMemberships": [],
                "VpcSecurityGroupMemberships": []
            },
            {
                "OptionName": "OEM",
                "OptionDescription": "Oracle 12c EM Express",
                "Persistent": false,
                "Permanent": false,
                "Port": 5500,
                "OptionSettings": [],
                "DBSecurityGroupMemberships": [
                    {
                        "DBSecurityGroupName": "default",
                        "Status": "authorized"
                    }
                ],
                "VpcSecurityGroupMemberships": []
            }
        ],
        "AllowsVpcAndNonVpcInstanceMemberships": false,
        "OptionGroupArn": "arn:aws:rds:us-east-1:123456789012:og:myoptiongroup"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的將選項新增至選項群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddOptionToOptionGroup。

add-role-to-db-cluster

以下程式碼範例顯示如何使用 add-role-to-db-cluster。

AWS CLI

將 AWS Identity and Access Management (IAM) 角色與資料庫叢集建立關聯

下列add-role-to-db-cluster範例會將角色與資料庫叢集建立關聯。

```
aws rds add-role-to-db-cluster \
    --db-cluster-identifier mydbcluster \
    --role-arn arn:aws:iam::123456789012:role/RDSLoadFromS3
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的將 IAM 角色與 Amazon Aurora MySQL 資料庫叢集建立關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddRoleToDbCluster。

add-role-to-db-instance

以下程式碼範例顯示如何使用 add-role-to-db-instance。

AWS CLI

將 AWS Identity and Access Management (IAM) 角色與資料庫執行個體建立關聯

下列add-role-to-db-instance範例會將 角色新增至名為 的 Oracle 資料庫執行個體test-instance。

```
aws rds add-role-to-db-instance \
    --db-instance-identifier test-instance \
    --feature-name S3_INTEGRATION \
    --role-arn arn:aws:iam::111122223333:role/rds-s3-integration-role
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的 Amazon RDS Oracle 與 Amazon S3 整合的先決條件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddRoleToDbInstance。

add-source-identifier-to-subscription

以下程式碼範例顯示如何使用 add-source-identifier-to-subscription。

AWS CLI

將來源識別符新增至訂閱

下列add-source-identifier範例會將另一個來源識別符新增至現有的訂閱。

```
aws rds add-source-identifier-to-subscription \
    --subscription-name my-instance-events \
    --source-identifier test-instance-repl
```

輸出：

```
{
    "EventSubscription": {
        "SubscriptionCreationTime": "Tue Jul 31 23:22:01 UTC 2018",
        "CustSubscriptionId": "my-instance-events",
        "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-
events",
        "Enabled": false,
        "Status": "modifying",
        "EventCategoriesList": [
            "backup",
            "recovery"
        ],
        "CustomerAwsId": "123456789012",
        "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",
        "SourceType": "db-instance",
        "SourceIdsList": [
            "test-instance",
            "test-instance-repl"
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddSourceIdentifierToSubscription。

add-tags-to-resource

以下程式碼範例顯示如何使用 add-tags-to-resource。

AWS CLI

將標籤新增至資源

下列add-tags-to-resource範例會將標籤新增至 RDS 資料庫。

```
aws rds add-tags-to-resource \
    --resource-name arn:aws:rds:us-east-1:123456789012:db:database-mysql \
    --tags "[{\"Key\": \"Name\",\"Value\": \"MyDatabase\"},{\"Key\": \"Environment
\",\"Value\": \"test\"}]"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的標記 Amazon RDS 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AddTagsToResource。

apply-pending-maintenance-action

以下程式碼範例顯示如何使用 apply-pending-maintenance-action。

AWS CLI

套用待定維護動作

下列apply-pending-maintenance-action範例會套用資料庫叢集的待定維護動作。

```
aws rds apply-pending-maintenance-action \
    --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:my-db-cluster \
    --apply-action system-update \
    --opt-in-type immediate
```

輸出：

```
{
    "ResourcePendingMaintenanceActions": {
        "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:cluster:my-db-
cluster",
        "PendingMaintenanceActionDetails": [
            {
                "Action": "system-update",
                "OptInStatus": "immediate",
                "CurrentApplyDate": "2021-01-23T01:07:36.100Z",
                "Description": "Upgrade to Aurora PostgreSQL 3.3.2"
            }
        ]
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的維護資料庫執行個體和《Amazon Aurora 使用者指南》中的維護 Amazon Aurora 資料庫叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ApplyPendingMaintenanceAction。

`authorize-db-security-group-ingress`

以下程式碼範例顯示如何使用 `authorize-db-security-group-ingress`。

AWS CLI

將 AWS Identity and Access Management (IAM) 角色與資料庫執行個體建立關聯

下列`authorize-db-security-group-ingress`範例使用 CIDR IP 範圍 192.0.2.0/24 的輸入規則設定預設安全群組。

```
aws rds authorize-db-security-group-ingress \
    --db-security-group-name default \
    --cidrip 192.0.2.0/24
```

輸出：

```
{
    "DBSecurityGroup": {
        "OwnerId": "123456789012",
        "DBSecurityGroupName": "default",
        "DBSecurityGroupDescription": "default",
        "EC2SecurityGroups": [],
        "IPRanges": [
            {
                "Status": "authorizing",
                "CIDRIP": "192.0.2.0/24"
            }
        ],
        "DBSecurityGroupArn": "arn:aws:rds:us-east-1:111122223333:secgrp:default"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的從 IP 範圍授權資料庫安全群組的網路存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AuthorizeDbSecurityGroupIngress。

backtrack-db-cluster

以下程式碼範例顯示如何使用 backtrack-db-cluster。

AWS CLI

恢復 Aurora 資料庫叢集

下列backtrack-db-cluster範例會將指定的資料庫叢集 sample-cluster 恢復至 2018 年 3 月 19 日上午 10 點。

```
aws rds backtrack-db-cluster --db-cluster-identifier sample-cluster --backtrack-
to 2018-03-19T10:00:00+00:00
```

此命令會輸出 JSON 區塊，以確認 RDS 資源的變更。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BacktrackDbCluster。

cancel-export-task

以下程式碼範例顯示如何使用 cancel-export-task。

AWS CLI

取消快照匯出至 Amazon S3

下列cancel-export-task範例會取消正在將快照匯出至 Amazon S3 的匯出任務。

```
aws rds cancel-export-task \
    --export-task-identifier my-s3-export-1
```

輸出：

```
{
    "ExportTaskIdentifier": "my-s3-export-1",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:snapshot:publisher-final-
snapshot",
    "SnapshotTime": "2019-03-24T20:01:09.815Z",
```

```
    "S3Bucket": "amzn-s3-demo-bucket",
    "S3Prefix": "",
    "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/export-snap-S3-role",
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/abcd0000-7bfd-4594-af38-
aabbccddeeff",
    "Status": "CANCELING",
    "PercentProgress": 0,
    "TotalExtractedDataInGB": 0
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的取消快照匯出任務或《Amazon Aurora 使用者指南》中的取消快照匯出任務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelExportTask。

copy-db-cluster-parameter-group

以下程式碼範例顯示如何使用 copy-db-cluster-parameter-group。

AWS CLI

複製資料庫叢集參數群組

下列copy-db-cluster-parameter-group範例會複製資料庫叢集參數群組。

```
aws rds copy-db-cluster-parameter-group \
    --source-db-cluster-parameter-group-identifier mydbclusterpg \
    --target-db-cluster-parameter-group-identifier mydbclusterpgcopy \
    --target-db-cluster-parameter-group-description "Copy of mydbclusterpg parameter
 group"
```

輸出：

```
{
    "DBClusterParameterGroup": {
        "DBClusterParameterGroupName": "mydbclusterpgcopy",
        "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:mydbclusterpgcopy",
        "DBParameterGroupFamily": "aurora-mysql5.7",
        "Description": "Copy of mydbclusterpg parameter group"
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的複製資料庫叢集參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CopyDbClusterParameterGroup。

copy-db-cluster-snapshot

以下程式碼範例顯示如何使用 copy-db-cluster-snapshot。

AWS CLI

複製資料庫叢集快照

下列copy-db-cluster-snapshot範例會建立資料庫叢集快照的副本，包括其標籤。

```
aws rds copy-db-cluster-snapshot \
    --source-db-cluster-snapshot-identifier arn:aws:rds:us-
east-1:123456789012:cluster-snapshot:rds:myaurora-2019-06-04-09-16
    --target-db-cluster-snapshot-identifier myclustersnapshotcopy \
    --copy-tags
```

輸出：

```
{
    "DBClusterSnapshot": {
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1e"
        ],
        "DBClusterSnapshotIdentifier": "myclustersnapshotcopy",
        "DBClusterIdentifier": "myaurora",
        "SnapshotCreateTime": "2019-06-04T09:16:42.649Z",
        "Engine": "aurora-mysql",
        "AllocatedStorage": 0,
        "Status": "available",
        "Port": 0,
        "VpcId": "vpc-6594f31c",
        "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
        "MasterUsername": "myadmin",
        "EngineVersion": "5.7.mysql_aurora.2.04.2",
        "LicenseModel": "aurora-mysql",
        "SnapshotType": "manual",
```

```
        "PercentProgress": 100,
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:123456789012:cluster-
snapshot:myclustersnapshotcopy",
        "IAMDatabaseAuthenticationEnabled": false
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的複製快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CopyDbClusterSnapshot。

copy-db-parameter-group

以下程式碼範例顯示如何使用 copy-db-parameter-group。

AWS CLI

複製資料庫叢集參數群組

下列copy-db-parameter-group範例會複製資料庫參數群組。

```
aws rds copy-db-parameter-group \
    --source-db-parameter-group-identifier mydbpg \
    --target-db-parameter-group-identifier mydbpgcopy \
    --target-db-parameter-group-description "Copy of mydbpg parameter group"
```

輸出：

```
{
    "DBParameterGroup": {
        "DBParameterGroupName": "mydbpgcopy",
        "DBParameterGroupArn": "arn:aws:rds:us-east-1:814387698303:pg:mydbpgcopy",
        "DBParameterGroupFamily": "mysql5.7",
        "Description": "Copy of mydbpg parameter group"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的複製資料庫參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CopyDbParameterGroup。

copy-db-snapshot

以下程式碼範例顯示如何使用 copy-db-snapshot。

AWS CLI

複製資料庫快照

下列copy-db-snapshot範例會建立資料庫快照的副本。

```
aws rds copy-db-snapshot \
    --source-db-snapshot-identifier rds:database-mysql-2019-06-06-08-38
    --target-db-snapshot-identifier mydbsnapshotcopy
```

輸出：

```
{
    "DBSnapshot": {
        "VpcId": "vpc-6594f31c",
        "Status": "creating",
        "Encrypted": true,
        "SourceDBSnapshotIdentifier": "arn:aws:rds:us-
east-1:123456789012:snapshot:rds:database-mysql-2019-06-06-08-38",
        "MasterUsername": "admin",
        "Iops": 1000,
        "Port": 3306,
        "LicenseModel": "general-public-license",
        "DBSnapshotArn": "arn:aws:rds:us-
east-1:123456789012:snapshot:mydbsnapshotcopy",
        "EngineVersion": "5.6.40",
        "OptionGroupName": "default:mysql-5-6",
        "ProcessorFeatures": [],
        "Engine": "mysql",
        "StorageType": "io1",
        "DbiResourceId": "db-ZI7UJ5BLKMBYFGX7FDENCKADC4",
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
        "SnapshotType": "manual",
        "IAMDatabaseAuthenticationEnabled": false,
        "SourceRegion": "us-east-1",
        "DBInstanceIdentifier": "database-mysql",
        "InstanceCreateTime": "2019-04-30T15:45:53.663Z",
        "AvailabilityZone": "us-east-1f",
        "PercentProgress": 0,
```

```
        "AllocatedStorage": 100,
        "DBSnapshotIdentifier": "mydbsnapshotcopy"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的複製快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CopyDbSnapshot。

copy-option-group

以下程式碼範例顯示如何使用 copy-option-group。

AWS CLI

複製選項群組

下列copy-option-group範例會複製 選項群組。

```
aws rds copy-option-group \
    --source-option-group-identifier myoptiongroup \
    --target-option-group-identifier new-option-group \
    --target-option-group-description "My option group copy"
```

輸出：

```
{
    "OptionGroup": {
        "Options": [],
        "OptionGroupName": "new-option-group",
        "MajorEngineVersion": "11.2",
        "OptionGroupDescription": "My option group copy",
        "AllowsVpcAndNonVpcInstanceMemberships": true,
        "EngineName": "oracle-ee",
        "OptionGroupArn": "arn:aws:rds:us-east-1:123456789012:og:new-option-group"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的複製選項群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CopyOptionGroup。

create-blue-green-deployment

以下程式碼範例顯示如何使用 create-blue-green-deployment。

AWS CLI

範例 1：為 RDS for MySQL 資料庫執行個體建立藍/綠部署

下列create-blue-green-deployment範例會為 MySQL 資料庫執行個體建立藍/綠部署。

```
aws rds create-blue-green-deployment \
    --blue-green-deployment-name bgd-cli-test-instance \
    --source arn:aws:rds:us-east-1:123456789012:db:my-db-instance \
    --target-engine-version 8.0 \
    --target-db-parameter-group-name mysql-80-group
```

輸出：

```
{
    "BlueGreenDeployment": {
        "BlueGreenDeploymentIdentifier": "bgd-v53303651eexfake",
        "BlueGreenDeploymentName": "bgd-cli-test-instance",
        "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
        "SwitchoverDetails": [
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3"
            }
        ],
        "Tasks": [
            {
                "Name": "CREATING_READ_REPLICA_OF_SOURCE",
```

```
                "Status": "PENDING"
            },
            {
                "Name": "DB_ENGINE_VERSION_UPGRADE",
                "Status": "PENDING"
            },
            {
                "Name": "CONFIGURE_BACKUPS",
                "Status": "PENDING"
            },
            {
                "Name": "CREATING_TOPOLOGY_OF_SOURCE",
                "Status": "PENDING"
            }
        ],
        "Status": "PROVISIONING",
        "CreateTime": "2022-02-25T21:18:51.183000+00:00"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的建立藍/綠部署。

範例 2：為 Aurora MySQL 資料庫叢集建立藍/綠部署

下列create-blue-green-deployment範例會為 Aurora MySQL 資料庫叢集建立藍/綠部署。

```
aws rds create-blue-green-deployment \
    --blue-green-deployment-name my-blue-green-deployment \
    --source arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster \
    --target-engine-version 8.0 \
    --target-db-cluster-parameter-group-name ams-80-binlog-enabled \
    --target-db-parameter-group-name mysql-80-cluster-group
```

輸出：

```
{
      "BlueGreenDeployment": {
        "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
        "BlueGreenDeploymentName": "my-blue-green-deployment",
        "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
        "SwitchoverDetails": [
          {
```

```
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-
mysql-cluster",
            "Status": "PROVISIONING"
          },
          {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-
cluster-1",
            "Status": "PROVISIONING"
          },
          {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-
cluster-2",
            "Status": "PROVISIONING"
          },
          {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-
cluster-3",
            "Status": "PROVISIONING"
          },
          {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-
excluded-member-endpoint",
            "Status": "PROVISIONING"
          },
          {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-
reader-endpoint",
            "Status": "PROVISIONING"
          }
        ],
        "Tasks": [
          {
            "Name": "CREATING_READ_REPLICA_OF_SOURCE",
            "Status": "PENDING"
          },
          {
            "Name": "DB_ENGINE_VERSION_UPGRADE",
            "Status": "PENDING"
          },
          {
            "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
            "Status": "PENDING"
          },
          {
```

```
            "Name": "CREATE_CUSTOM_ENDPOINTS",
            "Status": "PENDING"
          }
        ],
        "Status": "PROVISIONING",
        "CreateTime": "2022-02-25T21:12:00.288000+00:00"
      }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的建立藍/綠部署。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateBlueGreenDeployment。

create-db-cluster-endpoint

以下程式碼範例顯示如何使用 create-db-cluster-endpoint。

AWS CLI

建立自訂資料庫叢集端點

下列create-db-cluster-endpoint範例會建立自訂資料庫叢集端點，並將其與指定的 Aurora 資料庫叢集建立關聯。

```
aws rds create-db-cluster-endpoint \
    --db-cluster-endpoint-identifier mycustomendpoint \
    --endpoint-type reader \
    --db-cluster-identifier mydbcluster \
    --static-members dbinstance1 dbinstance2
```

輸出：

```
{
    "DBClusterEndpointIdentifier": "mycustomendpoint",
    "DBClusterIdentifier": "mydbcluster",
    "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",
    "Endpoint": "mycustomendpoint.cluster-custom-cnpexample.us-
east-1.rds.amazonaws.com",
    "Status": "creating",
    "EndpointType": "CUSTOM",
    "CustomEndpointType": "READER",
    "StaticMembers": [
```

```
        "dbinstance1",
        "dbinstance2"
    ],
    "ExcludedMembers": [],
    "DBClusterEndpointArn": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:mycustomendpoint"
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的 Amazon Aurora Connection Management。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDbClusterEndpoint。

create-db-cluster-parameter-group

以下程式碼範例顯示如何使用 create-db-cluster-parameter-group。

AWS CLI

建立資料庫叢集參數群組

下列create-db-cluster-parameter-group範例會建立資料庫叢集參數群組。

```
aws rds create-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterparametergroup \
    --db-parameter-group-family aurora5.6 \
    --description "My new cluster parameter group"
```

輸出：

```
{
    "DBClusterParameterGroup": {
        "DBClusterParameterGroupName": "mydbclusterparametergroup",
        "DBParameterGroupFamily": "aurora5.6",
        "Description": "My new cluster parameter group",
        "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:mydbclusterparametergroup"
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的建立資料庫叢集參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDbClusterParameterGroup。

create-db-cluster-snapshot

以下程式碼範例顯示如何使用 create-db-cluster-snapshot。

AWS CLI

建立資料庫叢集快照

下列create-db-cluster-snapshot範例會建立資料庫叢集快照。

```
aws rds create-db-cluster-snapshot \
    --db-cluster-identifier mydbcluster \
    --db-cluster-snapshot-identifier mydbclustersnapshot
```

輸出：

```
{
    "DBClusterSnapshot": {
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1e"
        ],
        "DBClusterSnapshotIdentifier": "mydbclustersnapshot",
        "DBClusterIdentifier": "mydbcluster",
        "SnapshotCreateTime": "2019-06-18T21:21:00.469Z",
        "Engine": "aurora-mysql",
        "AllocatedStorage": 1,
        "Status": "creating",
        "Port": 0,
        "VpcId": "vpc-6594f31c",
        "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
        "MasterUsername": "myadmin",
        "EngineVersion": "5.7.mysql_aurora.2.04.2",
        "LicenseModel": "aurora-mysql",
        "SnapshotType": "manual",
        "PercentProgress": 0,
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:123456789012:cluster-
snapshot:mydbclustersnapshot",
        "IAMDatabaseAuthenticationEnabled": false
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的建立資料庫叢集快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDbClusterSnapshot。

create-db-cluster

以下程式碼範例顯示如何使用 create-db-cluster。

AWS CLI

範例 1：建立 MySQL 5.7 相容資料庫叢集

下列create-db-cluster範例會使用預設引擎版本建立 MySQL 5.7 相容資料庫叢集。將範例密碼取代secret99為安全密碼。當您使用主控台建立資料庫叢集時，Amazon RDS 會自動為您的資料庫叢集建立寫入器資料庫執行個體。不過，當您使用 AWS CLI 建立資料庫叢集時，您必須使用 create-db-instance AWS CLI 命令明確建立資料庫叢集的寫入器資料庫執行個體。

```
aws rds create-db-cluster \
    --db-cluster-identifier sample-cluster \
    --engine aurora-mysql \
    --engine-version 5.7 \
    --master-username admin \
    --master-user-password secret99 \
    --db-subnet-group-name default \
    --vpc-security-group-ids sg-0b9130572daf3dc16
```

輸出：

```
{
    "DBCluster": {
        "DBSubnetGroup": "default",
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-0b9130572daf3dc16",
                "Status": "active"
            }
        ],
        "AllocatedStorage": 1,
        "AssociatedRoles": [],
        "PreferredBackupWindow": "09:12-09:42",
```

```
        "ClusterCreateTime": "2023-02-27T23:21:33.048Z",
        "DeletionProtection": false,
        "IAMDatabaseAuthenticationEnabled": false,
        "ReadReplicaIdentifiers": [],
        "EngineMode": "provisioned",
        "Engine": "aurora-mysql",
        "StorageEncrypted": false,
        "MultiAZ": false,
        "PreferredMaintenanceWindow": "mon:04:31-mon:05:01",
        "HttpEndpointEnabled": false,
        "BackupRetentionPeriod": 1,
        "DbClusterResourceId": "cluster-ANPAJ4AE5446DAEXAMPLE",
        "DBClusterIdentifier": "sample-cluster",
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1e"
        ],
        "MasterUsername": "master",
        "EngineVersion": "5.7.mysql_aurora.2.11.1",
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",
        "DBClusterMembers": [],
        "Port": 3306,
        "Status": "creating",
        "Endpoint": "sample-cluster.cluster-cnpexample.us-east-1.rds.amazonaws.com",
        "DBClusterParameterGroup": "default.aurora-mysql5.7",
        "HostedZoneId": "Z2R2ITUGPM61AM",
        "ReaderEndpoint": "sample-cluster.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
        "CopyTagsToSnapshot": false
    }
}
```

範例 2：建立與 PostgreSQL 相容的資料庫叢集

下列create-db-cluster範例會使用預設引擎版本建立 PostgreSQL 相容資料庫叢集。將範例密碼取代secret99為安全密碼。當您使用主控台建立資料庫叢集時，Amazon RDS 會自動為您的資料庫叢集建立寫入器資料庫執行個體。不過，當您使用 AWS CLI 建立資料庫叢集時，您必須使用 create-db-instance AWS CLI 命令明確建立資料庫叢集的寫入器資料庫執行個體。

```
aws rds create-db-cluster \
    --db-cluster-identifier sample-pg-cluster \
    --engine aurora-postgresql \
```

```
    --master-username master \
    --master-user-password secret99 \
    --db-subnet-group-name default \
    --vpc-security-group-ids sg-0b9130572daf3dc16
```

輸出：

```
{
    "DBCluster": {
        "Endpoint": "sample-pg-cluster.cluster-cnpexample.us-
east-1.rds.amazonaws.com",
        "HttpEndpointEnabled": false,
        "DBClusterMembers": [],
        "EngineMode": "provisioned",
        "CopyTagsToSnapshot": false,
        "HostedZoneId": "Z2R2ITUGPM61AM",
        "IAMDatabaseAuthenticationEnabled": false,
        "AllocatedStorage": 1,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-0b9130572daf3dc16",
                "Status": "active"
            }
        ],
        "DeletionProtection": false,
        "StorageEncrypted": false,
        "BackupRetentionPeriod": 1,
        "PreferredBackupWindow": "09:56-10:26",
        "ClusterCreateTime": "2023-02-27T23:26:08.371Z",
        "DBClusterParameterGroup": "default.aurora-postgresql13",
        "EngineVersion": "13.7",
        "Engine": "aurora-postgresql",
        "Status": "creating",
        "DBClusterIdentifier": "sample-pg-cluster",
        "MultiAZ": false,
        "Port": 5432,
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-pg-
cluster",
        "AssociatedRoles": [],
        "DbClusterResourceId": "cluster-ANPAJ4AE5446DAEXAMPLE",
        "PreferredMaintenanceWindow": "wed:03:33-wed:04:03",
        "ReaderEndpoint": "sample-pg-cluster.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
```

```
        "MasterUsername": "master",
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1c"
        ],
        "ReadReplicaIdentifiers": [],
        "DBSubnetGroup": "default"
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的建立 Amazon Aurora 資料庫叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDbCluster。

create-db-instance-read-replica

以下程式碼範例顯示如何使用 create-db-instance-read-replica。

AWS CLI

建立資料庫執行個體僅供讀取複本

此範例會建立名為 之現有資料庫執行個體的僅供讀取複本test-instance。僅供讀取複本名為 test-instance-repl。

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier test-instance-repl \
    --source-db-instance-identifier test-instance
```

輸出：

```
{
    "DBInstance": {
        "IAMDatabaseAuthenticationEnabled": false,
        "MonitoringInterval": 0,
        "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance-repl",
        "ReadReplicaSourceDBInstanceIdentifier": "test-instance",
        "DBInstanceIdentifier": "test-instance-repl",
        ...some output truncated...
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDbInstanceReadReplica。

create-db-instance

以下程式碼範例顯示如何使用 create-db-instance。

AWS CLI

建立資料庫執行個體

下列create-db-instance範例使用必要的選項來啟動新的資料庫執行個體。

```
aws rds create-db-instance \
    --db-instance-identifier test-mysql-instance \
    --db-instance-class db.t3.micro \
    --engine mysql \
    --master-username admin \
    --master-user-password secret99 \
    --allocated-storage 20
```

輸出：

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-mysql-instance",
        "DBInstanceClass": "db.t3.micro",
        "Engine": "mysql",
        "DBInstanceStatus": "creating",
        "MasterUsername": "admin",
        "AllocatedStorage": 20,
        "PreferredBackupWindow": "12:55-13:25",
        "BackupRetentionPeriod": 1,
        "DBSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-12345abc",
                "Status": "active"
            }
        ],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.mysql5.7",
                "ParameterApplyStatus": "in-sync"
```

```
            }
        ],
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-2ff2ff2f",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "sun:08:07-sun:08:37",
        "PendingModifiedValues": {
            "MasterUserPassword": "****"
        },
        "MultiAZ": false,
        "EngineVersion": "5.7.22",
```

```
        "AutoMinorVersionUpgrade": true,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "general-public-license",
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "default:mysql-5-7",
                "Status": "in-sync"
            }
        ],
        "PubliclyAccessible": true,
        "StorageType": "gp2",
        "DbInstancePort": 0,
        "StorageEncrypted": false,
        "DbiResourceId": "db-5555EXAMPLE44444444EXAMPLE",
        "CACertificateIdentifier": "rds-ca-2019",
        "DomainMemberships": [],
        "CopyTagsToSnapshot": false,
        "MonitoringInterval": 0,
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:test-mysql-
instance",
        "IAMDatabaseAuthenticationEnabled": false,
        "PerformanceInsightsEnabled": false,
        "DeletionProtection": false,
        "AssociatedRoles": []
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的建立 Amazon RDS 資料庫執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDBInstance。

create-db-parameter-group

以下程式碼範例顯示如何使用 create-db-parameter-group。

AWS CLI

建立資料庫參數群組

下列create-db-parameter-group範例會建立資料庫參數群組。

```
aws rds create-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --db-parameter-group-family MySQL5.6 \
```

```
    --description "My new parameter group"
```

輸出：

```
{
    "DBParameterGroup": {
        "DBParameterGroupName": "mydbparametergroup",
        "DBParameterGroupFamily": "mysql5.6",
        "Description": "My new parameter group",
        "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:mydbparametergroup"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的建立資料庫參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDBParameterGroup。

create-db-proxy-endpoint

以下程式碼範例顯示如何使用 create-db-proxy-endpoint。

AWS CLI

為 RDS 資料庫建立資料庫代理端點

下列create-db-proxy-endpoint範例會建立資料庫代理端點。

```
aws rds create-db-proxy-endpoint \
    --db-proxy-name proxyExample \
    --db-proxy-endpoint-name "proxyep1" \
    --vpc-subnet-ids subnetgroup1 subnetgroup2
```

輸出：

```
{
"DBProxyEndpoint": {
        "DBProxyEndpointName": "proxyep1",
        "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-
endpoint:prx-endpoint-0123a01b12345c0ab",
        "DBProxyName": "proxyExample",
        "Status": "creating",
```

```
        "VpcId": "vpc-1234567",
        "VpcSecurityGroupIds": [
            "sg-1234",
            "sg-5678"
        ],
        "VpcSubnetIds": [
            "subnetgroup1",
            "subnetgroup2"
        ],
        "Endpoint": "proxyep1.endpoint.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
        "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
        "TargetRole": "READ_WRITE",
        "IsDefault": false
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的建立代理端點和《Amazon Aurora 使用者指南》中的建立代理端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDbProxyEndpoint。

create-db-proxy

以下程式碼範例顯示如何使用 create-db-proxy。

AWS CLI

為 RDS 資料庫建立資料庫代理

下列create-db-proxy範例會建立資料庫代理。

```
aws rds create-db-proxy \
    --db-proxy-name proxyExample \
    --engine-family MYSQL \
    --auth
 Description="proxydescription1",AuthScheme="SECRETS",SecretArn="arn:aws:secretsmanager:us-
west-2:123456789123:secret:secretName-1234f",IAMAuth="DISABLED",ClientPasswordAuthType="MYSQ
 \
    --role-arn arn:aws:iam::123456789123:role/ProxyRole \
    --vpc-subnet-ids subnetgroup1 subnetgroup2
```

輸出：

```
{
"DBProxy": {
        "DBProxyName": "proxyExample",
        "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-0123a01b12345c0ab",
        "EngineFamily": "MYSQL",
        "VpcId": "vpc-1234567",
        "VpcSecuritytGroupIds": [
            "sg-1234",
            "sg-5678",
            "sg-9101"
        ],
        "VpcSubnetIds": [
            "subnetgroup1",
            "subnetgroup2"
        ],
        "Auth": "[
            {
                "Description": "proxydescription1",
                "AuthScheme": "SECRETS",
                "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret:proxysecret1-Abcd1e",
                "IAMAuth": "DISABLED"
            }
        ]",
        "RoleArn": "arn:aws:iam::12345678912:role/ProxyRole",
        "Endpoint": "proxyExample.proxy-ab0cd1efghij.us-east-1.rds.amazonaws.com",
        "RequireTLS": false,
        "IdleClientTimeout": 1800,
        "DebuggingLogging": false,
        "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
        "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的建立 RDS Proxy 和《Amazon Aurora 使用者指南》中的建立 RDS Proxy。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDbProxy。

create-db-security-group

以下程式碼範例顯示如何使用 create-db-security-group。

AWS CLI

建立 Amazon RDS 資料庫安全群組

下列create-db-security-group命令會建立新的 Amazon RDS 資料庫安全群組：

```
aws rds create-db-security-group --db-security-group-name mysecgroup --db-security-
group-description "My Test Security Group"
```

在此範例中，新的資料庫安全群組已命名mysecgroup並具有描述。

輸出：

```
{
    "DBSecurityGroup": {
        "OwnerId": "123456789012",
        "DBSecurityGroupName": "mysecgroup",
        "DBSecurityGroupDescription": "My Test Security Group",
        "VpcId": "vpc-a1b2c3d4",
        "EC2SecurityGroups": [],
        "IPRanges": [],
        "DBSecurityGroupArn": "arn:aws:rds:us-west-2:123456789012:secgrp:mysecgroup"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDbSecurityGroup。

create-db-shard-group

以下程式碼範例顯示如何使用 create-db-shard-group。

AWS CLI

範例 1：建立 Aurora PostgreSQL 主要資料庫叢集

下列create-db-cluster範例會建立與 Aurora Serverless v2 和 Aurora Limitless Database 相容的 Aurora PostgreSQL SQL 主要資料庫叢集。

```
aws rds create-db-cluster \
    --db-cluster-identifier my-sv2-cluster \
    --engine aurora-postgresql \
```

```
    --engine-version 15.2-limitless \
    --storage-type aurora-iopt1 \
    --serverless-v2-scaling-configuration MinCapacity=2,MaxCapacity=16 \
    --enable-limitless-database \
    --master-username myuser \
    --master-user-password mypassword \
    --enable-cloudwatch-logs-exports postgresql
```

輸出：

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-east-2b",
            "us-east-2c",
            "us-east-2a"
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "my-sv2-cluster",
        "DBClusterParameterGroup": "default.aurora-postgresql15",
        "DBSubnetGroup": "default",
        "Status": "creating",
        "Endpoint": "my-sv2-cluster.cluster-cekycexample.us-
east-2.rds.amazonaws.com",
        "ReaderEndpoint": "my-sv2-cluster.cluster-ro-cekycexample.us-
east-2.rds.amazonaws.com",
        "MultiAZ": false,
        "Engine": "aurora-postgresql",
        "EngineVersion": "15.2-limitless",
        "Port": 5432,
        "MasterUsername": "myuser",
        "PreferredBackupWindow": "06:05-06:35",
        "PreferredMaintenanceWindow": "mon:08:25-mon:08:55",
        "ReadReplicaIdentifiers": [],
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-########",
                "Status": "active"
            }
        ],
        "HostedZoneId": "Z2XHWR1EXAMPLE",
```

```
        "StorageEncrypted": false,
        "DbClusterResourceId": "cluster-XYEDT6ML6FHIXH4Q2J1EXAMPLE",
        "DBClusterArn": "arn:aws:rds:us-east-2:123456789012:cluster:my-sv2-cluster",
        "AssociatedRoles": [],
        "IAMDatabaseAuthenticationEnabled": false,
        "ClusterCreateTime": "2024-02-19T16:24:07.771000+00:00",
        "EnabledCloudwatchLogsExports": [
            "postgresql"
        ],
        "EngineMode": "provisioned",
        "DeletionProtection": false,
        "HttpEndpointEnabled": false,
        "CopyTagsToSnapshot": false,
        "CrossAccountClone": false,
        "DomainMemberships": [],
        "TagList": [],
        "StorageType": "aurora-iopt1",
        "AutoMinorVersionUpgrade": true,
        "ServerlessV2ScalingConfiguration": {
            "MinCapacity": 2.0,
            "MaxCapacity": 16.0
        },
        "NetworkType": "IPV4",
        "IOOptimizedNextAllowedModificationTime":
 "2024-03-21T16:24:07.781000+00:00",
        "LimitlessDatabase": {
            "Status": "not-in-use",
            "MinRequiredACU": 96.0
        }
    }
}
```

範例 2：建立主要（寫入器）資料庫執行個體

下列create-db-instance範例會建立 Aurora Serverless v2 主要（寫入器）資料庫執行個體。當您使用主控台建立資料庫叢集時，Amazon RDS 會自動為您的資料庫叢集建立寫入器資料庫執行個體。不過，當您使用 AWS CLI 建立資料庫叢集時，您必須使用 create-db-instance AWS CLI 命令明確建立資料庫叢集的寫入器資料庫執行個體。

```
aws rds create-db-instance \
    --db-instance-identifier my-sv2-instance \
    --db-cluster-identifier my-sv2-cluster \
    --engine aurora-postgresql \
```

```
    --db-instance-class db.serverless
```

輸出：

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "my-sv2-instance",
        "DBInstanceClass": "db.serverless",
        "Engine": "aurora-postgresql",
        "DBInstanceStatus": "creating",
        "MasterUsername": "myuser",
        "AllocatedStorage": 1,
        "PreferredBackupWindow": "06:05-06:35",
        "BackupRetentionPeriod": 1,
        "DBSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-########",
                "Status": "active"
            }
        ],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.aurora-postgresql15",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-########",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-2c"
                    },
                    "SubnetOutpost": {},
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
```

```
                "SubnetAvailabilityZone": {
                    "Name": "us-east-2a"
                },
                "SubnetOutpost": {},
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-########",
                "SubnetAvailabilityZone": {
                    "Name": "us-east-2b"
                },
                "SubnetOutpost": {},
                "SubnetStatus": "Active"
            }
        ]
    },
    "PreferredMaintenanceWindow": "fri:09:01-fri:09:31",
    "PendingModifiedValues": {
        "PendingCloudwatchLogsExports": {
            "LogTypesToEnable": [
                "postgresql"
            ]
        }
    },
    "MultiAZ": false,
    "EngineVersion": "15.2-limitless",
    "AutoMinorVersionUpgrade": true,
    "ReadReplicaDBInstanceIdentifiers": [],
    "LicenseModel": "postgresql-license",
    "OptionGroupMemberships": [
        {
            "OptionGroupName": "default:aurora-postgresql-15",
            "Status": "in-sync"
        }
    ],
    "PubliclyAccessible": false,
    "StorageType": "aurora-iopt1",
    "DbInstancePort": 0,
    "DBClusterIdentifier": "my-sv2-cluster",
    "StorageEncrypted": false,
    "DbiResourceId": "db-BIQTE3B3K3RM7M74SK5EXAMPLE",
    "CACertificateIdentifier": "rds-ca-rsa2048-g1",
    "DomainMemberships": [],
    "CopyTagsToSnapshot": false,
```

```
        "MonitoringInterval": 0,
        "PromotionTier": 1,
        "DBInstanceArn": "arn:aws:rds:us-east-2:123456789012:db:my-sv2-instance",
        "IAMDatabaseAuthenticationEnabled": false,
        "PerformanceInsightsEnabled": false,
        "DeletionProtection": false,
        "AssociatedRoles": [],
        "TagList": [],
        "CustomerOwnedIpEnabled": false,
        "BackupTarget": "region",
        "NetworkType": "IPV4",
        "StorageThroughput": 0,
        "CertificateDetails": {
            "CAIdentifier": "rds-ca-rsa2048-g1"
        },
        "DedicatedLogVolume": false
    }
}
```

範例 3：建立資料庫碎片群組

下列create-db-shard-group範例會在 Aurora PostgreSQL 主要資料庫叢集中建立資料庫碎片群組。

```
aws rds create-db-shard-group \
    --db-shard-group-identifier my-db-shard-group \
    --db-cluster-identifier my-sv2-cluster \
    --max-acu 768
```

輸出：

```
{
    "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
    "DBShardGroupIdentifier": "my-db-shard-group",
    "DBClusterIdentifier": "my-sv2-cluster",
    "MaxACU": 768.0,
    "ComputeRedundancy": 0,
    "Status": "creating",
    "PubliclyAccessible": false,
    "Endpoint": "my-sv2-cluster.limitless-cekycexample.us-east-2.rds.amazonaws.com"
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的使用 Aurora Serverless v2。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDbShardGroup。

create-db-snapshot

以下程式碼範例顯示如何使用 create-db-snapshot。

AWS CLI

建立資料庫快照

下列create-db-snapshot範例會建立資料庫快照。

```
aws rds create-db-snapshot \
    --db-instance-identifier database-mysql \
    --db-snapshot-identifier mydbsnapshot
```

輸出：

```
{
    "DBSnapshot": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBInstanceIdentifier": "database-mysql",
        "Engine": "mysql",
        "AllocatedStorage": 100,
        "Status": "creating",
        "Port": 3306,
        "AvailabilityZone": "us-east-1b",
        "VpcId": "vpc-6594f31c",
        "InstanceCreateTime": "2019-04-30T15:45:53.663Z",
        "MasterUsername": "admin",
        "EngineVersion": "5.6.40",
        "LicenseModel": "general-public-license",
        "SnapshotType": "manual",
        "Iops": 1000,
        "OptionGroupName": "default:mysql-5-6",
        "PercentProgress": 0,
        "StorageType": "io1",
        "Encrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
        "DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mydbsnapshot",
```

```
        "IAMDatabaseAuthenticationEnabled": false,
        "ProcessorFeatures": [],
        "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的建立資料庫快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDBSnapshot。

create-db-subnet-group

以下程式碼範例顯示如何使用 create-db-subnet-group。

AWS CLI

建立資料庫子網路群組

下列create-db-subnet-group範例mysubnetgroup會使用現有的子網路建立名為 的資料庫子網路群組。

```
aws rds create-db-subnet-group \
    --db-subnet-group-name mysubnetgroup \
    --db-subnet-group-description "test DB subnet group" \
    --subnet-ids
 '["subnet-0a1dc4e1a6f123456","subnet-070dd7ecb3aaaaaaa","subnet-00f5b198bc0abcdef"]'
```

輸出：

```
{
    "DBSubnetGroup": {
        "DBSubnetGroupName": "mysubnetgroup",
        "DBSubnetGroupDescription": "test DB subnet group",
        "VpcId": "vpc-0f08e7610a1b2c3d4",
        "SubnetGroupStatus": "Complete",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-070dd7ecb3aaaaaaa",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                },
                "SubnetStatus": "Active"
```

```
            },
            {
                "SubnetIdentifier": "subnet-00f5b198bc0abcdef",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2d"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-0a1dc4e1a6f123456",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                },
                "SubnetStatus": "Active"
            }
        ],
        "DBSubnetGroupArn": "arn:aws:rds:us-
west-2:0123456789012:subgrp:mysubnetgroup"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的在 VPC 中建立資料庫執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDbSubnetGroup。

create-event-subscription

以下程式碼範例顯示如何使用 create-event-subscription。

AWS CLI

建立事件訂閱

下列create-event-subscription範例會為目前 AWS 帳戶中的資料庫執行個體建立備份和復原事件的訂閱。通知會傳送至 指定的 Amazon Simple Notification Service 主題--sns-topic-arn。

```
aws rds create-event-subscription \
    --subscription-name my-instance-events \
    --source-type db-instance \
    --event-categories '["backup","recovery"]' \
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:interesting-events
```

輸出：

```
{
    "EventSubscription": {
        "Status": "creating",
        "CustSubscriptionId": "my-instance-events",
        "SubscriptionCreationTime": "Tue Jul 31 23:22:01 UTC 2018",
        "EventCategoriesList": [
            "backup",
            "recovery"
        ],
        "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",
        "CustomerAwsId": "123456789012",
        "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-
events",
        "SourceType": "db-instance",
        "Enabled": true
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateEventSubscription。

create-global-cluster

以下程式碼範例顯示如何使用 create-global-cluster。

AWS CLI

建立全域資料庫叢集

下列create-global-cluster範例會建立新的 Aurora MySQL 相容全域資料庫叢集。

```
aws rds create-global-cluster \
    --global-cluster-identifier myglobalcluster \
    --engine aurora-mysql
```

輸出：

```
{
    "GlobalCluster": {
        "GlobalClusterIdentifier": "myglobalcluster",
        "GlobalClusterResourceId": "cluster-f0e523bfe07aabb",
```

```
        "GlobalClusterArn": "arn:aws:rds::123456789012:global-
cluster:myglobalcluster",
        "Status": "available",
        "Engine": "aurora-mysql",
        "EngineVersion": "5.7.mysql_aurora.2.07.2",
        "StorageEncrypted": false,
        "DeletionProtection": false,
        "GlobalClusterMembers": []
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的建立 Aurora 全域資料庫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateGlobalCluster。

create-option-group

以下程式碼範例顯示如何使用 create-option-group。

AWS CLI

建立 Amazon RDS 選項群組

下列create-option-group命令會為Oracle Enterprise Edition版本 建立新的 Amazon RDS 選項群組，11.2`, is named ``MyOptionGroup並包含描述。

```
aws rds create-option-group \
    --option-group-name MyOptionGroup \
    --engine-name oracle-ee \
    --major-engine-version 11.2 \
    --option-group-description "Oracle Database Manager Database Control"
```

輸出：

```
{
    "OptionGroup": {
        "OptionGroupName": "myoptiongroup",
        "OptionGroupDescription": "Oracle Database Manager Database Control",
        "EngineName": "oracle-ee",
        "MajorEngineVersion": "11.2",
        "Options": [],
        "AllowsVpcAndNonVpcInstanceMemberships": true,
        "OptionGroupArn": "arn:aws:rds:us-west-2:123456789012:og:myoptiongroup"
```

```
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateOptionGroup。

delete-blue-green-deployment

以下程式碼範例顯示如何使用 delete-blue-green-deployment。

AWS CLI

範例 1：刪除 RDS for MySQL 資料庫執行個體綠色環境中的資源

下列delete-blue-green-deployment範例會刪除 RDS for MySQL 資料庫執行個體在綠色環境中的資源。

```
aws rds delete-blue-green-deployment \
    --blue-green-deployment-identifier bgd-v53303651eexfake \
    --delete-target
```

輸出：

```
{
    "BlueGreenDeployment": {
        "BlueGreenDeploymentIdentifier": "bgd-v53303651eexfake",
        "BlueGreenDeploymentName": "bgd-cli-test-instance",
        "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
        "Target": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-green-
rkfbpe",
        "SwitchoverDetails": [
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-green-rkfbpe",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1-green-j382ha",
```

```
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2-green-ejv4ao",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3-green-vlpz3t",
                "Status": "AVAILABLE"
            }
        ],
        "Tasks": [
            {
                "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                "Status": "COMPLETED"
            },
            {
                "Name": "DB_ENGINE_VERSION_UPGRADE",
                "Status": "COMPLETED"
            },
            {
                "Name": "CONFIGURE_BACKUPS",
                "Status": "COMPLETED"
            },
            {
                "Name": "CREATING_TOPOLOGY_OF_SOURCE",
                "Status": "COMPLETED"
            }
        ],
        "Status": "DELETING",
        "CreateTime": "2022-02-25T21:18:51.183000+00:00",
        "DeleteTime": "2022-02-25T22:25:31.331000+00:00"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的刪除藍/綠部署。

範例 2：刪除 Aurora MySQL 資料庫叢集綠色環境中的資源

下列delete-blue-green-deployment範例會刪除 Aurora MySQL 資料庫叢集綠色環境中的資源。

```
aws rds delete-blue-green-deployment \
    --blue-green-deployment-identifier bgd-wi89nwzglccsfake \
    --delete-target
```

輸出：

```
{
    "BlueGreenDeployment": {
        "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
        "BlueGreenDeploymentName": "my-blue-green-deployment",
        "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
        "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster-green-3rnukl",
        "SwitchoverDetails": [
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster-green-3rnukl",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-1",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-1-green-gpmaxf",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-2",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-2-green-j2oajq",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-3",
```

```
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-3-green-mkxies",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-green-4sqjrq",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-green-gwwzlg",
                "Status": "AVAILABLE"
            }
        ],
        "Tasks": [
            {
                "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                "Status": "COMPLETED"
            },
            {
                "Name": "DB_ENGINE_VERSION_UPGRADE",
                "Status": "COMPLETED"
            },
            {
                "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
                "Status": "COMPLETED"
            },
            {
                "Name": "CREATE_CUSTOM_ENDPOINTS",
                "Status": "COMPLETED"
            }
        ],
        "Status": "DELETING",
        "CreateTime": "2022-02-25T21:12:00.288000+00:00",
        "DeleteTime": "2022-02-25T22:29:11.336000+00:00"
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的刪除藍/綠部署。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBlueGreenDeployment。

delete-db-cluster-endpoint

以下程式碼範例顯示如何使用 delete-db-cluster-endpoint。

AWS CLI

刪除自訂資料庫叢集端點

下列delete-db-cluster-endpoint範例會刪除指定的自訂資料庫叢集端點。

```
aws rds delete-db-cluster-endpoint \
    --db-cluster-endpoint-identifier mycustomendpoint
```

輸出：

```
{
    "DBClusterEndpointIdentifier": "mycustomendpoint",
    "DBClusterIdentifier": "mydbcluster",
    "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",
    "Endpoint": "mycustomendpoint.cluster-custom-cnpexample.us-
east-1.rds.amazonaws.com",
    "Status": "deleting",
    "EndpointType": "CUSTOM",
    "CustomEndpointType": "READER",
    "StaticMembers": [
        "dbinstance1",
        "dbinstance2",
        "dbinstance3"
    ],
    "ExcludedMembers": [],
    "DBClusterEndpointArn": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:mycustomendpoint"
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的 Amazon Aurora Connection Management。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDbClusterEndpoint。

delete-db-cluster-parameter-group

以下程式碼範例顯示如何使用 delete-db-cluster-parameter-group。

AWS CLI

刪除資料庫叢集參數群組

下列delete-db-cluster-parameter-group範例會刪除指定的資料庫叢集參數群組。

```
aws rds delete-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterparametergroup
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的使用資料庫參數群組和資料庫叢集參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDbClusterParameterGroup。

delete-db-cluster-snapshot

以下程式碼範例顯示如何使用 delete-db-cluster-snapshot。

AWS CLI

刪除資料庫叢集快照

下列delete-db-cluster-snapshot範例會刪除指定的資料庫叢集快照。

```
aws rds delete-db-cluster-snapshot \
    --db-cluster-snapshot-identifier mydbclustersnapshot
```

輸出：

```
{
    "DBClusterSnapshot": {
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1e"
        ],
```

```
        "DBClusterSnapshotIdentifier": "mydbclustersnapshot",
        "DBClusterIdentifier": "mydbcluster",
        "SnapshotCreateTime": "2019-06-18T21:21:00.469Z",
        "Engine": "aurora-mysql",
        "AllocatedStorage": 0,
        "Status": "available",
        "Port": 0,
        "VpcId": "vpc-6594f31c",
        "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
        "MasterUsername": "myadmin",
        "EngineVersion": "5.7.mysql_aurora.2.04.2",
        "LicenseModel": "aurora-mysql",
        "SnapshotType": "manual",
        "PercentProgress": 100,
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:123456789012:cluster-
snapshot:mydbclustersnapshot",
        "IAMDatabaseAuthenticationEnabled": false
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的刪除快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDbClusterSnapshot。

delete-db-cluster

以下程式碼範例顯示如何使用 delete-db-cluster。

AWS CLI

範例 1：刪除資料庫叢集中的資料庫執行個體

下列delete-db-instance範例會刪除資料庫叢集中的最終資料庫執行個體。如果資料庫叢集包含未處於刪除狀態的資料庫執行個體，則無法刪除該資料庫叢集。刪除資料庫叢集中的資料庫執行個體時，您無法拍攝最終快照。

```
aws rds delete-db-instance \
    --db-instance-identifier database-3
```

輸出：

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "database-3",
        "DBInstanceClass": "db.r4.large",
        "Engine": "aurora-postgresql",
        "DBInstanceStatus": "deleting",

    ...output omitted...

    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的刪除 Aurora 資料庫叢集中的資料庫執行個體。

範例 2：刪除資料庫叢集

下列delete-db-cluster範例會刪除名為 的資料庫叢集，mycluster並拍攝名為 的最終快照mycluster-final-snapshot。拍攝快照時，可使用資料庫叢集的狀態。若要遵循刪除的進度，請使用 CLI describe-db-clusters 命令。

```
aws rds delete-db-cluster \
    --db-cluster-identifier mycluster \
    --no-skip-final-snapshot \
    --final-db-snapshot-identifier mycluster-final-snapshot
```

輸出：

```
{
    "DBCluster": {
        "AllocatedStorage": 20,
        "AvailabilityZones": [
            "eu-central-1b",
            "eu-central-1c",
            "eu-central-1a"
        ],
        "BackupRetentionPeriod": 7,
        "DBClusterIdentifier": "mycluster",
        "DBClusterParameterGroup": "default.aurora-postgresql10",
        "DBSubnetGroup": "default-vpc-aa11bb22",
        "Status": "available",
```

```
    ...output omitted...

    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的具有單一資料庫執行個體的 Aurora 叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDbCluster。

delete-db-instance-automated-backup

以下程式碼範例顯示如何使用 delete-db-instance-automated-backup。

AWS CLI

從區域刪除複寫的自動備份

下列delete-db-instance-automated-backup範例會刪除具有指定 Amazon Resource Name (ARN) 的自動備份。

```
aws rds delete-db-instance-automated-backup \
    --db-instance-automated-backups-arn "arn:aws:rds:us-west-2:123456789012:auto-backup:ab-jkib2gfq5rv7replzadausbrktni2bn4example"
```

輸出：

```
{
    "DBInstanceAutomatedBackup": {
        "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db",
        "DbiResourceId": "db-JKIB2GFQ5RV7REPLZA4EXAMPLE",
        "Region": "us-east-1",
        "DBInstanceIdentifier": "new-orcl-db",
        "RestoreWindow": {},
        "AllocatedStorage": 20,
        "Status": "deleting",
        "Port": 1521,
        "AvailabilityZone": "us-east-1b",
        "VpcId": "vpc-########",
        "InstanceCreateTime": "2020-12-04T15:28:31Z",
        "MasterUsername": "admin",
```

```
        "Engine": "oracle-se2",
        "EngineVersion": "12.1.0.2.v21",
        "LicenseModel": "bring-your-own-license",
        "OptionGroupName": "default:oracle-se2-12-1",
        "Encrypted": false,
        "StorageType": "gp2",
        "IAMDatabaseAuthenticationEnabled": false,
        "BackupRetentionPeriod": 7,
        "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-
backup:ab-jkib2gfq5rv7replzadausbrktni2bn4example"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的刪除複寫備份。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDbInstanceAutomatedBackup。

`delete-db-instance`

以下程式碼範例顯示如何使用 `delete-db-instance`。

AWS CLI

刪除資料庫執行個體

下列`delete-db-instance`範例會在建立名為 的最終資料庫快照後刪除指定的資料庫執行個體`test-instance-final-snap`。

```
aws rds delete-db-instance \
    --db-instance-identifier test-instance \
    --final-db-snapshot-identifier test-instance-final-snap
```

輸出：

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-instance",
        "DBInstanceStatus": "deleting",
        ...some output truncated...
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDBInstance。

delete-db-parameter-group

以下程式碼範例顯示如何使用 delete-db-parameter-group。

AWS CLI

刪除資料庫參數群組

下列command範例會刪除資料庫參數群組。

```
aws rds delete-db-parameter-group \
    --db-parameter-group-name mydbparametergroup
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的使用資料庫參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDBParameterGroup。

delete-db-proxy-endpoint

以下程式碼範例顯示如何使用 delete-db-proxy-endpoint。

AWS CLI

刪除 RDS 資料庫的資料庫代理端點

下列delete-db-proxy-endpoint範例會刪除目標資料庫的資料庫代理端點。

```
aws rds delete-db-proxy-endpoint \
    --db-proxy-endpoint-name proxyEP1
```

輸出：

```
{
"DBProxyEndpoint":
    {
        "DBProxyEndpointName": "proxyEP1",
```

```
        "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-
endpoint:prx-endpoint-0123a01b12345c0ab",
        "DBProxyName": "proxyExample",
        "Status": "deleting",
        "VpcId": "vpc-1234567",
        "VpcSecurityGroupIds": [
            "sg-1234",
            "sg-5678"
        ],
        "VpcSubnetIds": [
            "subnetgroup1",
            "subnetgroup2"
        ],
        "Endpoint": "proxyEP1.endpoint.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
        "CreatedDate": "2023-04-13T01:49:38.568000+00:00",
        "TargetRole": "READ_ONLY",
        "IsDefault": false
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的刪除代理端點和《Amazon Aurora 使用者指南》中的刪除代理端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDbProxyEndpoint。

delete-db-proxy

以下程式碼範例顯示如何使用 delete-db-proxy。

AWS CLI

刪除 RDS 資料庫的資料庫代理

下列delete-db-proxy範例會刪除資料庫代理。

```
aws rds delete-db-proxy \
    --db-proxy-name proxyExample
```

輸出：

```
{
```

```
        "DBProxy":
        {
            "DBProxyName": "proxyExample",
            "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-0123a01b12345c0ab",
            "Status": "deleting",
            "EngineFamily": "PostgreSQL",
            "VpcId": "vpc-1234567",
            "VpcSecurityGroupIds": [
                "sg-1234",
                "sg-5678"
            ],
            "VpcSubnetIds": [
                "subnetgroup1",
                "subnetgroup2"
            ],
            "Auth": "[
                {
                "Description": "proxydescription`"
                "AuthScheme": "SECRETS",
                "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret:proxysecret1-Abcd1e",
                "IAMAuth": "DISABLED"
                } ],
            "RoleArn": "arn:aws:iam::12345678912:role/ProxyPostgreSQLRole",
            "Endpoint": "proxyExample.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
            "RequireTLS": false,
            "IdleClientTimeout": 1800,
            "DebuggingLogging": false,
        "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
        "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的刪除 RDS Proxy 和《Amazon Aurora 使用者指南》中的刪除 RDS Proxy。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDbProxy。

delete-db-security-group

以下程式碼範例顯示如何使用 delete-db-security-group。

AWS CLI

刪除資料庫安全群組

下列delete-db-security-group範例會刪除名為 的資料庫安全群組mysecuritygroup。

```
aws rds delete-db-security-group \
    --db-security-group-name mysecuritygroup
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的使用資料庫安全群組 (EC2-Classic 平台)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDbSecurityGroup。

delete-db-shard-group

以下程式碼範例顯示如何使用 delete-db-shard-group。

AWS CLI

範例 1：刪除資料庫碎片群組失敗

下列delete-db-shard-group範例顯示嘗試在刪除所有資料庫和結構描述之前刪除資料庫碎片群組時發生的錯誤。

```
aws rds delete-db-shard-group \
    --db-shard-group-identifier limitless-test-shard-grp
```

輸出：

```
An error occurred (InvalidDBShardGroupState) when calling the DeleteDBShardGroup
 operation: Unable to delete the DB shard group limitless-test-db-shard-group.
Delete all of your Limitless Database databases and schemas, then try again.
```

範例 2：成功刪除資料庫碎片群組

下列delete-db-shard-group範例會在刪除所有資料庫和結構描述後刪除資料庫碎片群組，包括public結構描述。

```
aws rds delete-db-shard-group \
```

```
    --db-shard-group-identifier limitless-test-shard-grp
```

輸出：

```
{
    "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",
    "DBShardGroupIdentifier": "limitless-test-shard-grp",
    "DBClusterIdentifier": "limitless-test-cluster",
    "MaxACU": 768.0,
    "ComputeRedundancy": 0,
    "Status": "deleting",
    "PubliclyAccessible": true,
    "Endpoint": "limitless-test-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的刪除 Aurora 資料庫叢集和資料庫執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDbShardGroup。

delete-db-snapshot

以下程式碼範例顯示如何使用 delete-db-snapshot。

AWS CLI

刪除資料庫快照

下列delete-db-snapshot範例會刪除指定的資料庫快照。

```
aws rds delete-db-snapshot \
    --db-snapshot-identifier mydbsnapshot
```

輸出：

```
{
    "DBSnapshot": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBInstanceIdentifier": "database-mysql",
        "SnapshotCreateTime": "2019-06-18T22:08:40.702Z",
```

```
        "Engine": "mysql",
        "AllocatedStorage": 100,
        "Status": "deleted",
        "Port": 3306,
        "AvailabilityZone": "us-east-1b",
        "VpcId": "vpc-6594f31c",
        "InstanceCreateTime": "2019-04-30T15:45:53.663Z",
        "MasterUsername": "admin",
        "EngineVersion": "5.6.40",
        "LicenseModel": "general-public-license",
        "SnapshotType": "manual",
        "Iops": 1000,
        "OptionGroupName": "default:mysql-5-6",
        "PercentProgress": 100,
        "StorageType": "io1",
        "Encrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
        "DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mydbsnapshot",
        "IAMDatabaseAuthenticationEnabled": false,
        "ProcessorFeatures": [],
        "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的刪除快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDbSnapshot。

delete-db-subnet-group

以下程式碼範例顯示如何使用 delete-db-subnet-group。

AWS CLI

刪除資料庫子網路群組

下列delete-db-subnet-group範例會刪除名為 的資料庫子網路群組mysubnetgroup。

```
aws rds delete-db-subnet-group --db-subnet-group-name mysubnetgroup
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的在 VPC 中使用資料庫執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDbSubnetGroup。

delete-event-subscription

以下程式碼範例顯示如何使用 delete-event-subscription。

AWS CLI

刪除事件訂閱

下列delete-event-subscription範例會刪除指定的事件訂閱。

```
aws rds delete-event-subscription --subscription-name my-instance-events
```

輸出：

```
{
    "EventSubscription": {
        "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-
events",
        "CustomerAwsId": "123456789012",
        "Enabled": false,
        "SourceIdsList": [
            "test-instance"
        ],
        "SourceType": "db-instance",
        "EventCategoriesList": [
            "backup",
            "recovery"
        ],
        "SubscriptionCreationTime": "2018-07-31 23:22:01.893",
        "CustSubscriptionId": "my-instance-events",
        "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",
        "Status": "deleting"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteEventSubscription。

delete-global-cluster

以下程式碼範例顯示如何使用 delete-global-cluster。

AWS CLI

刪除全域資料庫叢集

下列delete-global-cluster範例會刪除與 Aurora MySQL 相容的全域資料庫叢集。輸出會顯示您要刪除的叢集，但後續describe-global-clusters命令不會列出該資料庫叢集。

```
aws rds delete-global-cluster \
    --global-cluster-identifier myglobalcluster
```

輸出：

```
{
    "GlobalCluster": {
        "GlobalClusterIdentifier": "myglobalcluster",
        "GlobalClusterResourceId": "cluster-f0e523bfe07aabb",
        "GlobalClusterArn": "arn:aws:rds::123456789012:global-
cluster:myglobalcluster",
        "Status": "available",
        "Engine": "aurora-mysql",
        "EngineVersion": "5.7.mysql_aurora.2.07.2",
        "StorageEncrypted": false,
        "DeletionProtection": false,
        "GlobalClusterMembers": []
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的刪除 Aurora 全域資料庫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteGlobalCluster。

delete-option-group

以下程式碼範例顯示如何使用 delete-option-group。

AWS CLI

刪除選項群組

下列delete-option-group範例會刪除指定的選項群組。

```
aws rds delete-option-group \
```

```
    --option-group-name myoptiongroup
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的刪除選項群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteOptionGroup。

deregister-db-proxy-targets

以下程式碼範例顯示如何使用 deregister-db-proxy-targets。

AWS CLI

從資料庫目標群組取消註冊資料庫代理目標

下列deregister-db-proxy-targets範例會移除代理proxyExample與其目標之間的關聯。

```
aws rds deregister-db-proxy-targets \
    --db-proxy-name proxyExample \
    --db-instance-identifiers database-1
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的刪除 RDS Proxy 和《Amazon Aurora 使用者指南》中的刪除 RDS Proxy。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterDbProxyTargets。

describe-account-attributes

以下程式碼範例顯示如何使用 describe-account-attributes。

AWS CLI

描述帳戶屬性

下列describe-account-attributes範例會擷取目前 AWS 帳戶的屬性。

```
aws rds describe-account-attributes
```

輸出：

```
{
    "AccountQuotas": [
        {
            "Max": 40,
            "Used": 4,
            "AccountQuotaName": "DBInstances"
        },
        {
            "Max": 40,
            "Used": 0,
            "AccountQuotaName": "ReservedDBInstances"
        },
        {
            "Max": 100000,
            "Used": 40,
            "AccountQuotaName": "AllocatedStorage"
        },
        {
            "Max": 25,
            "Used": 0,
            "AccountQuotaName": "DBSecurityGroups"
        },
        {
            "Max": 20,
            "Used": 0,
            "AccountQuotaName": "AuthorizationsPerDBSecurityGroup"
        },
        {
            "Max": 50,
            "Used": 1,
            "AccountQuotaName": "DBParameterGroups"
        },
        {
            "Max": 100,
            "Used": 3,
            "AccountQuotaName": "ManualSnapshots"
        },
        {
            "Max": 20,
            "Used": 0,
            "AccountQuotaName": "EventSubscriptions"
        },
```

```
        {
            "Max": 50,
            "Used": 1,
            "AccountQuotaName": "DBSubnetGroups"
        },
        {
            "Max": 20,
            "Used": 1,
            "AccountQuotaName": "OptionGroups"
        },
        {
            "Max": 20,
            "Used": 6,
            "AccountQuotaName": "SubnetsPerDBSubnetGroup"
        },
        {
            "Max": 5,
            "Used": 0,
            "AccountQuotaName": "ReadReplicasPerMaster"
        },
        {
            "Max": 40,
            "Used": 1,
            "AccountQuotaName": "DBClusters"
        },
        {
            "Max": 50,
            "Used": 0,
            "AccountQuotaName": "DBClusterParameterGroups"
        },
        {
            "Max": 5,
            "Used": 0,
            "AccountQuotaName": "DBClusterRoles"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAccountAttributes。

describe-blue-green-deployments

以下程式碼範例顯示如何使用 describe-blue-green-deployments。

AWS CLI

範例 1：描述建立完成後 RDS 資料庫執行個體的藍/綠部署

下列describe-blue-green-deployment範例會在建立完成後擷取藍/綠部署的詳細資訊。

```
aws rds describe-blue-green-deployments \
    --blue-green-deployment-identifier bgd-v53303651eexfake
```

輸出：

```
{
    "BlueGreenDeployments": [
        {
            "BlueGreenDeploymentIdentifier": "bgd-v53303651eexfake",
            "BlueGreenDeploymentName": "bgd-cli-test-instance",
            "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
            "Target": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-green-
rkfbpe",
            "SwitchoverDetails": [
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-green-rkfbpe",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1-green-j382ha",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2-green-ejv4ao",
                    "Status": "AVAILABLE"
                },
                {
```

```
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3-green-vlpz3t",
                    "Status": "AVAILABLE"
                }
            ],
            "Tasks": [
                {
                    "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "DB_ENGINE_VERSION_UPGRADE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CONFIGURE_BACKUPS",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CREATING_TOPOLOGY_OF_SOURCE",
                    "Status": "COMPLETED"
                }
            ],
            "Status": "AVAILABLE",
            "CreateTime": "2022-02-25T21:18:51.183000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的檢視藍/綠部署。

範例 2：描述 Aurora MySQL 資料庫叢集的藍/綠部署

下列`describe-blue-green-deployment`範例會擷取藍/綠部署的詳細資訊。

```
aws rds describe-blue-green-deployments \
    --blue-green-deployment-identifier bgd-wi89nwzglccsfake
```

輸出：

```
{
```

```
    "BlueGreenDeployments": [
        {
            "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
            "BlueGreenDeploymentName": "my-blue-green-deployment",
            "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
            "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster-green-3rnukl",
            "SwitchoverDetails": [
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster-green-3rnukl",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-1-green-gpmaxf",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-2",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-2-green-j2oajq",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3-green-mkxies",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-green-4sqjrq",
                    "Status": "AVAILABLE"
                },
```

```
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-green-gwwzlg",
                    "Status": "AVAILABLE"
                }
            ],
            "Tasks": [
                {
                    "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "DB_ENGINE_VERSION_UPGRADE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CREATE_CUSTOM_ENDPOINTS",
                    "Status": "COMPLETED"
                }
            ],
            "Status": "AVAILABLE",
            "CreateTime": "2022-02-25T21:12:00.288000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的檢視藍/綠部署。

範例 3：描述切換後 Aurora MySQL 叢集的藍/綠部署

下列`describe-blue-green-deployment`範例會在綠色環境提升為生產環境之後，擷取藍/綠部署的詳細資訊。

```
aws rds describe-blue-green-deployments \
    --blue-green-deployment-identifier bgd-wi89nwzglccsfake
```

輸出：

```
{
    "BlueGreenDeployments": [
        {
            "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
            "BlueGreenDeploymentName": "my-blue-green-deployment",
            "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster-old1",
            "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
            "SwitchoverDetails": [
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster",
                    "Status": "SWITCHOVER_COMPLETED"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-1-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-1",
                    "Status": "SWITCHOVER_COMPLETED"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-2-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-2",
                    "Status": "SWITCHOVER_COMPLETED"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3",
                    "Status": "SWITCHOVER_COMPLETED"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
```

```
                    "Status": "SWITCHOVER_COMPLETED"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
                    "Status": "SWITCHOVER_COMPLETED"
                }
            ],
            "Tasks": [
                {
                    "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "DB_ENGINE_VERSION_UPGRADE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CREATE_CUSTOM_ENDPOINTS",
                    "Status": "COMPLETED"
                }
            ],
            "Status": "SWITCHOVER_COMPLETED",
            "CreateTime": "2022-02-25T22:38:49.522000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的檢視藍/綠部署。

範例 4：描述合併的藍/綠部署

下列describe-blue-green-deployment範例會擷取合併藍/綠部署的詳細資訊。

```
aws rds describe-blue-green-deployments
```

輸出：

```
{
    "BlueGreenDeployments": [
        {
            "BlueGreenDeploymentIdentifier": "bgd-wi89nwzgfakelccs",
            "BlueGreenDeploymentName": "my-blue-green-deployment",
            "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
            "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster-green-3rnukl",
            "SwitchoverDetails": [
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster-green-3rnukl",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-1-green-gpmaxf",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-2",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-2-green-j2oajq",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3-green-mkxies",
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-green-4sqjrq",
```

```
                    "Status": "AVAILABLE"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-green-gwwzlg",
                    "Status": "AVAILABLE"
                }
            ],
            "Tasks": [
                {
                    "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "DB_ENGINE_VERSION_UPGRADE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CREATE_CUSTOM_ENDPOINTS",
                    "Status": "COMPLETED"
                }
            ],
            "Status": "AVAILABLE",
            "CreateTime": "2022-02-25T21:12:00.288000+00:00"
        },
        {
            "BlueGreenDeploymentIdentifier": "bgd-v5330365fake1eex",
            "BlueGreenDeploymentName": "bgd-cli-test-instance",
            "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-old1",
            "Target": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
            "SwitchoverDetails": [
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance",
                    "Status": "SWITCHOVER_COMPLETED"
                },
```

```
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1",
                    "Status": "SWITCHOVER_COMPLETED"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2",
                    "Status": "SWITCHOVER_COMPLETED"
                },
                {
                    "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3-old1",
                    "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3",
                    "Status": "SWITCHOVER_COMPLETED"
                }
            ],
            "Tasks": [
                {
                    "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "DB_ENGINE_VERSION_UPGRADE",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CONFIGURE_BACKUPS",
                    "Status": "COMPLETED"
                },
                {
                    "Name": "CREATING_TOPOLOGY_OF_SOURCE",
                    "Status": "COMPLETED"
                }
            ],
            "Status": "SWITCHOVER_COMPLETED",
            "CreateTime": "2022-02-25T22:33:22.225000+00:00"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的檢視藍/綠部署和《Amazon Aurora 使用者指南》中的檢視藍/綠部署。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeBlueGreenDeployments。

describe-certificates

以下程式碼範例顯示如何使用 describe-certificates。

AWS CLI

描述憑證

下列describe-certificates範例會擷取與使用者預設區域相關聯之憑證的詳細資訊。

```
aws rds describe-certificates
```

輸出：

```
{
    "Certificates": [
        {
            "CertificateIdentifier": "rds-ca-ecc384-g1",
            "CertificateType": "CA",
            "Thumbprint": "2ee3dcc06e50192559b13929e73484354f23387d",
            "ValidFrom": "2021-05-24T22:06:59+00:00",
            "ValidTill": "2121-05-24T23:06:59+00:00",
            "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-ecc384-g1",
            "CustomerOverride": false
        },
        {
            "CertificateIdentifier": "rds-ca-rsa4096-g1",
            "CertificateType": "CA",
            "Thumbprint": "19da4f2af579a8ae1f6a0fa77aa5befd874b4cab",
            "ValidFrom": "2021-05-24T22:03:20+00:00",
            "ValidTill": "2121-05-24T23:03:20+00:00",
            "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-rsa4096-g1",
            "CustomerOverride": false
        },
        {
            "CertificateIdentifier": "rds-ca-rsa2048-g1",
```

```
            "CertificateType": "CA",
            "Thumbprint": "7c40cb42714b6fdb2b296f9bbd0e8bb364436a76",
            "ValidFrom": "2021-05-24T21:59:00+00:00",
            "ValidTill": "2061-05-24T22:59:00+00:00",
            "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-rsa2048-g1",
            "CustomerOverride": true,
            "CustomerOverrideValidTill": "2061-05-24T22:59:00+00:00"
        },
        {
            "CertificateIdentifier": "rds-ca-2019",
            "CertificateType": "CA",
            "Thumbprint": "d40ddb29e3750dffa671c3140bbf5f478d1c8096",
            "ValidFrom": "2019-08-22T17:08:50+00:00",
            "ValidTill": "2024-08-22T17:08:50+00:00",
            "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-2019",
            "CustomerOverride": false
        }
    ],
    "DefaultCertificateForNewLaunches": "rds-ca-rsa2048-g1"
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的使用 SSL/TLS 加密與資料庫執行個體的連線，以及《Amazon Aurora 使用者指南》中的使用 SSL/TLS 加密與資料庫叢集的連線。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCertificates。

describe-db-cluster-backtracks

以下程式碼範例顯示如何使用 describe-db-cluster-backtracks。

AWS CLI

描述資料庫叢集的恢復

下列describe-db-cluster-backtracks範例會擷取指定資料庫叢集的詳細資訊。

```
aws rds describe-db-cluster-backtracks \
    --db-cluster-identifier mydbcluster
```

輸出：

```
{
    "DBClusterBacktracks": [
```

```
        {
            "DBClusterIdentifier": "mydbcluster",
            "BacktrackIdentifier": "2f5f5294-0dd2-44c9-9f50-EXAMPLE",
            "BacktrackTo": "2021-02-12T04:59:22Z",
            "BacktrackedFrom": "2021-02-12T14:37:31.640Z",
            "BacktrackRequestCreationTime": "2021-02-12T14:36:18.819Z",
            "Status": "COMPLETED"
        },
        {
            "DBClusterIdentifier": "mydbcluster",
            "BacktrackIdentifier": "3c7a6421-af2a-4ea3-ae95-EXAMPLE",
            "BacktrackTo": "2021-02-11T22:53:46Z",
            "BacktrackedFrom": "2021-02-12T00:09:27.006Z",
            "BacktrackRequestCreationTime": "2021-02-12T00:07:53.487Z",
            "Status": "COMPLETED"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的恢復 Aurora 資料庫叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbClusterBacktracks。

describe-db-cluster-endpoints

以下程式碼範例顯示如何使用 describe-db-cluster-endpoints。

AWS CLI

範例 1：描述資料庫叢集端點

下列describe-db-cluster-endpoints範例會擷取資料庫叢集端點的詳細資訊。最常見的 Aurora 叢集類型有兩個端點。一個端點的類型為 WRITER。您可以將此端點用於所有 SQL 陳述式。另一個端點的類型為 READER。您只能將此端點用於 SELECT 和其他唯讀 SQL 陳述式。

```
aws rds describe-db-cluster-endpoints
```

輸出：

```
{
    "DBClusterEndpoints": [
        {
```

```
            "DBClusterIdentifier": "my-database-1",
            "Endpoint": "my-database-1.cluster-cnpexample.us-
east-1.rds.amazonaws.com",
            "Status": "creating",
            "EndpointType": "WRITER"
        },
        {
            "DBClusterIdentifier": "my-database-1",
            "Endpoint": "my-database-1.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
            "Status": "creating",
            "EndpointType": "READER"
        },
        {
            "DBClusterIdentifier": "mydbcluster",
            "Endpoint": "mydbcluster.cluster-cnpexamle.us-east-1.rds.amazonaws.com",
            "Status": "available",
            "EndpointType": "WRITER"
        },
        {
            "DBClusterIdentifier": "mydbcluster",
            "Endpoint": "mydbcluster.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
            "Status": "available",
            "EndpointType": "READER"
        }
    ]
}
```

範例 2：描述單一資料庫叢集的資料庫叢集端點

下列describe-db-cluster-endpoints範例會擷取單一指定資料庫叢集之資料庫叢集端點的詳細資訊。Aurora Serverless 叢集只有具有 類型的單一端點WRITER。

```
aws rds describe-db-cluster-endpoints \
    --db-cluster-identifier serverless-cluster
```

輸出：

```
{
    "DBClusterEndpoints": [
        {
            "Status": "available",
```

```
            "Endpoint": "serverless-cluster.cluster-cnpexample.us-
east-1.rds.amazonaws.com",
            "DBClusterIdentifier": "serverless-cluster",
            "EndpointType": "WRITER"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的 Amazon Aurora Connection Management。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbClusterEndpoints。

describe-db-cluster-parameter-groups

以下程式碼範例顯示如何使用 describe-db-cluster-parameter-groups。

AWS CLI

描述資料庫叢集參數群組

下列describe-db-cluster-parameter-groups範例會擷取資料庫叢集參數群組的詳細資訊。

```
aws rds describe-db-cluster-parameter-groups
```

輸出：

```
{
    "DBClusterParameterGroups": [
        {
            "DBClusterParameterGroupName": "default.aurora-mysql5.7",
            "DBParameterGroupFamily": "aurora-mysql5.7",
            "Description": "Default cluster parameter group for aurora-mysql5.7",
            "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:default.aurora-mysql5.7"
        },
        {
            "DBClusterParameterGroupName": "default.aurora-postgresql9.6",
            "DBParameterGroupFamily": "aurora-postgresql9.6",
            "Description": "Default cluster parameter group for aurora-
postgresql9.6",
```

```
            "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:default.aurora-postgresql9.6"
        },
        {
            "DBClusterParameterGroupName": "default.aurora5.6",
            "DBParameterGroupFamily": "aurora5.6",
            "Description": "Default cluster parameter group for aurora5.6",
            "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:default.aurora5.6"
        },
        {
            "DBClusterParameterGroupName": "mydbclusterpg",
            "DBParameterGroupFamily": "aurora-mysql5.7",
            "Description": "My DB cluster parameter group",
            "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:mydbclusterpg"
        },
        {
            "DBClusterParameterGroupName": "mydbclusterpgcopy",
            "DBParameterGroupFamily": "aurora-mysql5.7",
            "Description": "Copy of mydbclusterpg parameter group",
            "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:mydbclusterpgcopy"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的使用資料庫參數群組和資料庫叢集參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbClusterParameterGroups。

describe-db-cluster-parameters

以下程式碼範例顯示如何使用 describe-db-cluster-parameters。

AWS CLI

範例 1：描述資料庫叢集參數群組中的參數

下列describe-db-cluster-parameters範例會擷取資料庫叢集參數群組中參數的詳細資訊。

```
aws rds describe-db-cluster-parameters \
```

```
    --db-cluster-parameter-group-name mydbclusterpg
```

輸出：

```
{
    "Parameters": [
        {
            "ParameterName": "allow-suspicious-udfs",
            "Description": "Controls whether user-defined functions that have only
 an xxx symbol for the main function can be loaded",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
            "ApplyMethod": "pending-reboot",
            "SupportedEngineModes": [
                "provisioned"
            ]
        },
        {
            "ParameterName": "aurora_lab_mode",
            "ParameterValue": "0",
            "Description": "Enables new features in the Aurora engine.",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot",
            "SupportedEngineModes": [
                "provisioned"
            ]
        },
        ...some output truncated...
    ]
}
```

範例 2：僅列出資料庫叢集參數群組中的參數名稱

下列describe-db-cluster-parameters範例只會擷取資料庫叢集參數群組中參數的名稱。

```
aws rds describe-db-cluster-parameters \
```

```
    --db-cluster-parameter-group-name default.aurora-mysql5.7 \
    --query 'Parameters[].{ParameterName:ParameterName}'
```

輸出：

```
[
    {
        "ParameterName": "allow-suspicious-udfs"
    },
    {
        "ParameterName": "aurora_binlog_read_buffer_size"
    },
    {
        "ParameterName": "aurora_binlog_replication_max_yield_seconds"
    },
    {
        "ParameterName": "aurora_binlog_use_large_read_buffer"
    },
    {
        "ParameterName": "aurora_lab_mode"
    },

    ...some output truncated...
    }
]
```

範例 3：僅描述資料庫叢集參數群組中的可修改參數

下列describe-db-cluster-parameters範例只會擷取您可以在資料庫叢集參數群組中修改的參數名稱。

```
aws rds describe-db-cluster-parameters \
    --db-cluster-parameter-group-name default.aurora-mysql5.7 \
    --query 'Parameters[].{ParameterName:ParameterName,IsModifiable:IsModifiable} |
 [?IsModifiable == `true`]'
```

輸出：

```
[
    {
        "ParameterName": "aurora_binlog_read_buffer_size",
        "IsModifiable": true
```

```
    },
    {
        "ParameterName": "aurora_binlog_replication_max_yield_seconds",
        "IsModifiable": true
    },
    {
        "ParameterName": "aurora_binlog_use_large_read_buffer",
        "IsModifiable": true
    },
    {
        "ParameterName": "aurora_lab_mode",
        "IsModifiable": true
    },

    ...some output truncated...
    }
]
```

範例 4：僅描述資料庫叢集參數群組中的可修改布林值參數

下列describe-db-cluster-parameters範例只會擷取您可以在資料庫叢集參數群組中修改且具有布林值資料類型的參數名稱。

```
aws rds describe-db-cluster-parameters \
    --db-cluster-parameter-group-name default.aurora-mysql5.7 \
    --query 'Parameters[].
{ParameterName:ParameterName,DataType:DataType,IsModifiable:IsModifiable} | [?
DataType == `boolean`] | [?IsModifiable == `true`]'
```

輸出：

```
[
    {
        "DataType": "boolean",
        "ParameterName": "aurora_binlog_use_large_read_buffer",
        "IsModifiable": true
    },
    {
        "DataType": "boolean",
        "ParameterName": "aurora_lab_mode",
        "IsModifiable": true
    },
    {
```

```
        "DataType": "boolean",
        "ParameterName": "autocommit",
        "IsModifiable": true
    },
    {
        "DataType": "boolean",
        "ParameterName": "automatic_sp_privileges",
        "IsModifiable": true
    },
    ...some output truncated...
    }
]
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的使用資料庫參數群組和資料庫叢集參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbClusterParameters。

describe-db-cluster-snapshot-attributes

以下程式碼範例顯示如何使用 describe-db-cluster-snapshot-attributes。

AWS CLI

描述資料庫叢集快照的屬性名稱和值

下列describe-db-cluster-snapshot-attributes範例會擷取指定資料庫叢集快照之屬性名稱和值的詳細資訊。

```
aws rds describe-db-cluster-snapshot-attributes \
    --db-cluster-snapshot-identifier myclustersnapshot
```

輸出：

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotIdentifier": "myclustersnapshot",
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123456789012"
                ]
```

```
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的共用資料庫叢集快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbClusterSnapshotAttributes。

describe-db-cluster-snapshots

以下程式碼範例顯示如何使用 describe-db-cluster-snapshots。

AWS CLI

描述資料庫叢集的資料庫叢集快照

下列describe-db-cluster-snapshots範例會擷取指定資料庫叢集之資料庫叢集快照的詳細資訊。

```
aws rds describe-db-cluster-snapshots \
    --db-cluster-identifier mydbcluster
```

輸出：

```
{
    "DBClusterSnapshots": [
        {
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1e"
            ],
            "DBClusterSnapshotIdentifier": "myclustersnapshotcopy",
            "DBClusterIdentifier": "mydbcluster",
            "SnapshotCreateTime": "2019-06-04T09:16:42.649Z",
            "Engine": "aurora-mysql",
            "AllocatedStorage": 0,
            "Status": "available",
            "Port": 0,
            "VpcId": "vpc-6594f31c",
            "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
```

```
            "MasterUsername": "myadmin",
            "EngineVersion": "5.7.mysql_aurora.2.04.2",
            "LicenseModel": "aurora-mysql",
            "SnapshotType": "manual",
            "PercentProgress": 100,
            "StorageEncrypted": true,
            "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
AKIAIOSFODNN7EXAMPLE",
            "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:814387698303:cluster-
snapshot:myclustersnapshotcopy",
            "IAMDatabaseAuthenticationEnabled": false
        },
        {
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1e"
            ],
            "DBClusterSnapshotIdentifier": "rds:mydbcluster-2019-06-20-09-16",
            "DBClusterIdentifier": "mydbcluster",
            "SnapshotCreateTime": "2019-06-20T09:16:26.569Z",
            "Engine": "aurora-mysql",
            "AllocatedStorage": 0,
            "Status": "available",
            "Port": 0,
            "VpcId": "vpc-6594f31c",
            "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
            "MasterUsername": "myadmin",
            "EngineVersion": "5.7.mysql_aurora.2.04.2",
            "LicenseModel": "aurora-mysql",
            "SnapshotType": "automated",
            "PercentProgress": 100,
            "StorageEncrypted": true,
            "KmsKeyId": "arn:aws:kms:us-east-1:814387698303:key/
AKIAIOSFODNN7EXAMPLE",
            "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:123456789012:cluster-
snapshot:rds:mydbcluster-2019-06-20-09-16",
            "IAMDatabaseAuthenticationEnabled": false
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的建立資料庫叢集快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbClusterSnapshots。

describe-db-clusters

以下程式碼範例顯示如何使用 describe-db-clusters。

AWS CLI

範例 1：描述資料庫叢集

下列describe-db-clusters範例會擷取指定資料庫叢集的詳細資訊。

```
aws rds describe-db-clusters \
    --db-cluster-identifier mydbcluster
```

輸出：

```
{
    "DBClusters": [
        {
            "AllocatedStorage": 1,
            "AvailabilityZones": [
                "us-east-1a",
                "us-east-1b",
                "us-east-1e"
            ],
            "BackupRetentionPeriod": 1,
            "DatabaseName": "mydbcluster",
            "DBClusterIdentifier": "mydbcluster",
            "DBClusterParameterGroup": "default.aurora-mysql5.7",
            "DBSubnetGroup": "default",
            "Status": "available",
            "EarliestRestorableTime": "2019-06-19T09:16:28.210Z",
            "Endpoint": "mydbcluster.cluster-cnpexample.us-
east-1.rds.amazonaws.com",
            "ReaderEndpoint": "mydbcluster.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
            "MultiAZ": true,
            "Engine": "aurora-mysql",
            "EngineVersion": "5.7.mysql_aurora.2.04.2",
            "LatestRestorableTime": "2019-06-20T22:38:14.908Z",
            "Port": 3306,
            "MasterUsername": "myadmin",
```

```
            "PreferredBackupWindow": "09:09-09:39",
            "PreferredMaintenanceWindow": "sat:04:09-sat:04:39",
            "ReadReplicaIdentifiers": [],
            "DBClusterMembers": [
                {
                    "DBInstanceIdentifier": "dbinstance3",
                    "IsClusterWriter": false,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                },
                {
                    "DBInstanceIdentifier": "dbinstance1",
                    "IsClusterWriter": false,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                },
                {
                    "DBInstanceIdentifier": "dbinstance2",
                    "IsClusterWriter": false,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                },
                {
                    "DBInstanceIdentifier": "mydbcluster",
                    "IsClusterWriter": false,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                },
                {
                    "DBInstanceIdentifier": "mydbcluster-us-east-1b",
                    "IsClusterWriter": false,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                },
                {
                    "DBInstanceIdentifier": "mydbcluster",
                    "IsClusterWriter": true,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                }
            ],
            "VpcSecurityGroups": [
                {
                    "VpcSecurityGroupId": "sg-0b9130572daf3dc16",
```

```
                    "Status": "active"
                }
            ],
            "HostedZoneId": "Z2R2ITUGPM61AM",
            "StorageEncrypted": true,
            "KmsKeyId": "arn:aws:kms:us-east-1:814387698303:key/
AKIAIOSFODNN7EXAMPLE",
            "DbClusterResourceId": "cluster-AKIAIOSFODNN7EXAMPLE",
            "DBClusterArn": "arn:aws:rds:us-
east-1:123456789012:cluster:mydbcluster",
            "AssociatedRoles": [],
            "IAMDatabaseAuthenticationEnabled": false,
            "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
            "EngineMode": "provisioned",
            "DeletionProtection": false,
            "HttpEndpointEnabled": false
        }
    ]
}
```

範例 2：列出所有資料庫叢集的特定屬性

下列describe-db-clusters範例只會擷取目前 AWS 區域中所有資料庫叢集的 DBClusterIdentifierEndpoint、 和 ReaderEndpoint 屬性。

```
aws rds describe-db-clusters \
    --query 'DBClusters[].
{DBClusterIdentifier:DBClusterIdentifier,Endpoint:Endpoint,ReaderEndpoint:ReaderEndpoint}'
```

輸出：

```
[
    {
        "Endpoint": "cluster-57-2020-05-01-2270.cluster-cnpexample.us-
east-1.rds.amazonaws.com",
        "ReaderEndpoint": "cluster-57-2020-05-01-2270.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
        "DBClusterIdentifier": "cluster-57-2020-05-01-2270"
    },
    {
        "Endpoint": "cluster-57-2020-05-01-4615.cluster-cnpexample.us-
east-1.rds.amazonaws.com",
```

```
        "ReaderEndpoint": "cluster-57-2020-05-01-4615.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
        "DBClusterIdentifier": "cluster-57-2020-05-01-4615"
    },
    {
        "Endpoint": "pg2-cluster.cluster-cnpexample.us-east-1.rds.amazonaws.com",
        "ReaderEndpoint": "pg2-cluster.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
        "DBClusterIdentifier": "pg2-cluster"
    },
    ...output omitted...
    }
]
```

範例 3：列出具有特定屬性的資料庫叢集

下列describe-db-clusters範例只會擷取使用資料庫引擎之資料庫叢集的 aurora-postgresql DBClusterIdentifier和 Engine 屬性。

```
aws rds describe-db-clusters \
    --query 'DBClusters[].{DBClusterIdentifier:DBClusterIdentifier,Engine:Engine} |
 [?Engine == `aurora-postgresql`]'
```

輸出：

```
[
    {
        "Engine": "aurora-postgresql",
        "DBClusterIdentifier": "pg2-cluster"
    }
]
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的 Amazon Aurora 資料庫叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbClusters。

describe-db-engine-versions

以下程式碼範例顯示如何使用 describe-db-engine-versions。

AWS CLI

描述 MySQL 資料庫引擎的資料庫引擎版本

下列describe-db-engine-versions範例顯示指定資料庫引擎之每個資料庫引擎版本的詳細資訊。

```
aws rds describe-db-engine-versions \
    --engine mysql
```

輸出：

```
{
    "DBEngineVersions": [
        {
            "Engine": "mysql",
            "EngineVersion": "5.5.46",
            "DBParameterGroupFamily": "mysql5.5",
            "DBEngineDescription": "MySQL Community Edition",
            "DBEngineVersionDescription": "MySQL 5.5.46",
            "ValidUpgradeTarget": [
                {
                    "Engine": "mysql",
                    "EngineVersion": "5.5.53",
                    "Description": "MySQL 5.5.53",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                {
                    "Engine": "mysql",
                    "EngineVersion": "5.5.54",
                    "Description": "MySQL 5.5.54",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                {
                    "Engine": "mysql",
                    "EngineVersion": "5.5.57",
                    "Description": "MySQL 5.5.57",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                ...some output truncated...
            ]
        }
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的什麼是 Amazon Relational Database Service (Amazon RDS)？。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDBEngineVersions。

describe-db-instance-automated-backups

以下程式碼範例顯示如何使用 describe-db-instance-automated-backups。

AWS CLI

描述資料庫執行個體的自動備份

下列describe-db-instance-automated-backups範例顯示指定資料庫執行個體自動備份的詳細資訊。詳細資訊包括其他 AWS 區域中的複寫自動備份。

```
aws rds describe-db-instance-automated-backups \
    --db-instance-identifier new-orcl-db
```

輸出：

```
{
    "DBInstanceAutomatedBackups": [
        {
            "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db",
            "DbiResourceId": "db-JKIB2GFQ5RV7REPLZA4EXAMPLE",
            "Region": "us-east-1",
            "DBInstanceIdentifier": "new-orcl-db",
            "RestoreWindow": {
                "EarliestTime": "2020-12-07T21:05:20.939Z",
                "LatestTime": "2020-12-07T21:05:20.939Z"
            },
            "AllocatedStorage": 20,
            "Status": "replicating",
            "Port": 1521,
            "InstanceCreateTime": "2020-12-04T15:28:31Z",
            "MasterUsername": "admin",
            "Engine": "oracle-se2",
            "EngineVersion": "12.1.0.2.v21",
            "LicenseModel": "bring-your-own-license",
            "OptionGroupName": "default:oracle-se2-12-1",
            "Encrypted": false,
```

```
            "StorageType": "gp2",
            "IAMDatabaseAuthenticationEnabled": false,
            "BackupRetentionPeriod": 14,
            "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-
west-2:123456789012:auto-backup:ab-jkib2gfq5rv7replzadausbrktni2bn4example"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的尋找有關複寫備份的資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbInstanceAutomatedBackups。

describe-db-instances

以下程式碼範例顯示如何使用 describe-db-instances。

AWS CLI

描述資料庫執行個體

下列describe-db-instances範例會擷取指定資料庫執行個體的詳細資訊。

```
aws rds describe-db-instances \
    --db-instance-identifier mydbinstancecf
```

輸出：

```
{
    "DBInstances": [
        {
            "DBInstanceIdentifier": "mydbinstancecf",
            "DBInstanceClass": "db.t3.small",
            "Engine": "mysql",
            "DBInstanceStatus": "available",
            "MasterUsername": "masterawsuser",
            "Endpoint": {
                "Address": "mydbinstancecf.abcexample.us-east-1.rds.amazonaws.com",
                "Port": 3306,
                "HostedZoneId": "Z2R2ITUGPM61AM"
            },
```

```
            ...some output truncated...
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDBInstances。

describe-db-log-files

以下程式碼範例顯示如何使用 describe-db-log-files。

AWS CLI

描述資料庫執行個體的日誌檔案

下列describe-db-log-files範例會擷取指定資料庫執行個體之日誌檔案的詳細資訊。

```
aws rds describe-db-log-files -\
    -db-instance-identifier test-instance
```

輸出：

```
{
    "DescribeDBLogFiles": [
        {
            "Size": 0,
            "LastWritten": 1533060000000,
            "LogFileName": "error/mysql-error-running.log"
        },
        {
            "Size": 2683,
            "LastWritten": 1532994300000,
            "LogFileName": "error/mysql-error-running.log.0"
        },
        {
            "Size": 107,
            "LastWritten": 1533057300000,
            "LogFileName": "error/mysql-error-running.log.18"
        },
        {
            "Size": 13105,
            "LastWritten": 1532991000000,
```

```
            "LogFileName": "error/mysql-error-running.log.23"
        },
        {
            "Size": 0,
            "LastWritten": 1533061200000,
            "LogFileName": "error/mysql-error.log"
        },
        {
            "Size": 3519,
            "LastWritten": 1532989252000,
            "LogFileName": "mysqlUpgrade"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbLogFiles。

describe-db-parameter-groups

以下程式碼範例顯示如何使用 describe-db-parameter-groups。

AWS CLI

描述資料庫參數群組

下列describe-db-parameter-groups範例會擷取資料庫參數群組的詳細資訊。

```
aws rds describe-db-parameter-groups
```

輸出：

```
{
    "DBParameterGroups": [
        {
            "DBParameterGroupName": "default.aurora-mysql5.7",
            "DBParameterGroupFamily": "aurora-mysql5.7",
            "Description": "Default parameter group for aurora-mysql5.7",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora-mysql5.7"
        },
        {
            "DBParameterGroupName": "default.aurora-postgresql9.6",
```

```
            "DBParameterGroupFamily": "aurora-postgresql9.6",
            "Description": "Default parameter group for aurora-postgresql9.6",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora-postgresql9.6"
        },
        {
            "DBParameterGroupName": "default.aurora5.6",
            "DBParameterGroupFamily": "aurora5.6",
            "Description": "Default parameter group for aurora5.6",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora5.6"
        },
        {
            "DBParameterGroupName": "default.mariadb10.1",
            "DBParameterGroupFamily": "mariadb10.1",
            "Description": "Default parameter group for mariadb10.1",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.mariadb10.1"
        },
        ...some output truncated...
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的使用資料庫參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDBParameterGroups。

describe-db-parameters

以下程式碼範例顯示如何使用 describe-db-parameters。

AWS CLI

描述資料庫參數群組中的參數

下列describe-db-parameters範例會擷取指定資料庫參數群組的詳細資訊。

```
aws rds describe-db-parameters \
    --db-parameter-group-name mydbpg
```

輸出：

```
{
```

```
    "Parameters": [
        {
            "ParameterName": "allow-suspicious-udfs",
            "Description": "Controls whether user-defined functions that have only
 an xxx symbol for the main function can be loaded",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
            "ApplyMethod": "pending-reboot"
        },
        {
            "ParameterName": "auto_generate_certs",
            "Description": "Controls whether the server autogenerates SSL key and
 certificate files in the data directory, if they do not already exist.",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
            "ApplyMethod": "pending-reboot"
        },
        ...some output truncated...
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的使用資料庫參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDBParameters。

describe-db-proxies

以下程式碼範例顯示如何使用 describe-db-proxies。

AWS CLI

描述 RDS 資料庫的資料庫代理

下列describe-db-proxies範例會傳回資料庫代理的相關資訊。

```
aws rds describe-db-proxies
```

輸出：

```
{
    "DBProxies": [
        {
            "DBProxyName": "proxyExample1",
            "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-0123a01b12345c0ab",
            "Status": "available",
            "EngineFamily": "PostgreSQL",
            "VpcId": "vpc-1234567",
            "VpcSecurityGroupIds": [
                "sg-1234"
            ],
            "VpcSubnetIds": [
                "subnetgroup1",
                "subnetgroup2"
            ],
            "Auth": "[
                {
                    "Description": "proxydescription1"
                    "AuthScheme": "SECRETS",
                    "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret:secretName-1234f",
                    "IAMAuth": "DISABLED"
                }
            ]",
            "RoleArn": "arn:aws:iam::12345678912??:role/ProxyPostgreSQLRole",
            "Endpoint": "proxyExample1.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
            "RequireTLS": false,
            "IdleClientTimeout": 1800,
            "DebuggingLogging": false,
            "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
            "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
        },
        {
            "DBProxyName": "proxyExample2",
            "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-1234a12b23456c1ab",
            "Status": "available",
            "EngineFamily": "PostgreSQL",
            "VpcId": "sg-1234567",
            "VpcSecurityGroupIds": [
```

```
                "sg-1234"
            ],
            "VpcSubnetIds": [
                "subnetgroup1",
                "subnetgroup2"
            ],
            "Auth": "[
                {
                    "Description": "proxydescription2"
                    "AuthScheme": "SECRETS",
                    "SecretArn": "aarn:aws:secretsmanager:us-
west-2:123456789123:secret:secretName-1234f",
                    "IAMAuth": "DISABLED"
                }
            ]",
            "RoleArn": "arn:aws:iam::12345678912:role/ProxyPostgreSQLRole",
            "Endpoint": "proxyExample2.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
            "RequireTLS": false,
            "IdleClientTimeout": 1800,
            "DebuggingLogging": false,
            "CreatedDate": "2022-01-05T16:19:33.452000+00:00",
            "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的檢視 RDS Proxy 和《Amazon Aurora 使用者指南》中的檢視 RDS Proxy。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbProxies。

describe-db-proxy-endpoints

以下程式碼範例顯示如何使用 describe-db-proxy-endpoints。

AWS CLI

描述資料庫代理端點

下列describe-db-proxy-endpoints範例會傳回資料庫代理端點的相關資訊。

```
aws rds describe-db-proxy-endpoints
```

輸出：

```
{
    "DBProxyEndpoints": [
        {
            "DBProxyEndpointName": "proxyEndpoint1",
            "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-
endpoint:prx-endpoint-0123a01b12345c0ab",
            "DBProxyName": "proxyExample",
            "Status": "available",
            "VpcId": "vpc-1234567",
            "VpcSecurityGroupIds": [
                "sg-1234"
            ],
            "VpcSubnetIds": [
                "subnetgroup1",
                "subnetgroup2"
            ],
            "Endpoint": "proxyEndpoint1.endpoint.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
            "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
            "TargetRole": "READ_WRITE",
            "IsDefault": false
        },
        {
            "DBProxyEndpointName": "proxyEndpoint2",
            "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-
endpoint:prx-endpoint-4567a01b12345c0ab",
            "DBProxyName": "proxyExample2",
            "Status": "available",
            "VpcId": "vpc1234567",
            "VpcSecurityGroupIds": [
                "sg-5678"
            ],
            "VpcSubnetIds": [
                "subnetgroup1",
                "subnetgroup2"
            ],
            "Endpoint": "proxyEndpoint2.endpoint.proxy-cd1ef2klmnop.us-
east-1.rds.amazonaws.com",
            "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
            "TargetRole": "READ_WRITE",
            "IsDefault": false
        }
```

```
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的檢視代理端點和《Amazon Aurora 使用者指南》中的建立代理端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbProxyEndpoints。

describe-db-proxy-target-groups

以下程式碼範例顯示如何使用 describe-db-proxy-target-groups。

AWS CLI

描述資料庫代理端點

下列describe-db-proxy-target-groups範例會傳回資料庫代理目標群組的相關資訊。

```
aws rds describe-db-proxy-target-groups \
    --db-proxy-name proxyExample
```

輸出：

```
{
"TargetGroups":
    {
        "DBProxyName": "proxyExample",
        "TargetGroupName": "default",
        "TargetGroupArn": "arn:aws:rds:us-east-1:123456789012:target-group:prx-
tg-0123a01b12345c0ab",
        "IsDefault": true,
        "Status": "available",
        "ConnectionPoolConfig": {
            "MaxConnectionsPercent": 100,
            "MaxIdleConnectionsPercent": 50,
            "ConnectionBorrowTimeout": 120,
            "SessionPinningFilters": []
        },
        "CreatedDate": "2023-05-02T18:41:19.495000+00:00",
        "UpdatedDate": "2023-05-02T18:41:21.762000+00:00"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的檢視 RDS Proxy 和《Amazon Aurora 使用者指南》中的檢視 RDS Proxy。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbProxyTargetGroups。

describe-db-proxy-targets

以下程式碼範例顯示如何使用 describe-db-proxy-targets。

AWS CLI

描述資料庫代理目標

下列describe-db-proxy-targets範例會傳回資料庫代理目標的相關資訊。

```
aws rds describe-db-proxy-targets \
    --db-proxy-name proxyExample
```

輸出：

```
{
    "Targets": [
        {
            "Endpoint": "database1.ab0cd1efghij.us-east-1.rds.amazonaws.com",
            "TrackedClusterId": "database1",
            "RdsResourceId": "database1-instance-1",
            "Port": 3306,
            "Type": "RDS_INSTANCE",
            "Role": "READ_WRITE",
            "TargetHealth": {
                "State": "UNAVAILABLE",
                "Reason": "PENDING_PROXY_CAPACITY",
                "Description": "DBProxy Target is waiting for proxy to scale to
 desired capacity"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的檢視 RDS 代理和《Amazon Aurora 使用者指南》中的檢視 RDS 代理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbProxyTargets。

describe-db-recommendations

以下程式碼範例顯示如何使用 describe-db-recommendations。

AWS CLI

範例 1：列出所有資料庫建議

下列describe-db-recommendations範例列出您 AWS 帳戶中的所有資料庫建議。

```
aws rds describe-db-recommendations
```

輸出：

```
{
    "DBRecommendations": [
        {
            "RecommendationId": "12ab3cde-f456-7g8h-9012-i3j45678k9lm",
            "TypeId": "config_recommendation::old_minor_version",
            "Severity": "informational",
            "ResourceArn": "arn:aws:rds:us-west-2:111122223333:db:database-1",
            "Status": "active",
            "CreatedTime": "2024-02-21T23:14:19.292000+00:00",
            "UpdatedTime": "2024-02-21T23:14:19+00:00",
            "Detection": "**[resource-name]** is not running the latest minor DB
 engine version",
            "Recommendation": "Upgrade to latest engine version",
            "Description": "Your database resources aren't running the latest minor
 DB engine version. The latest minor version contains the latest security fixes and
 other improvements.",
            "RecommendedActions": [
                {
                    "ActionId": "12ab34c5de6fg7h89i0jk1lm234n5678",
                    "Operation": "modifyDbInstance",
                    "Parameters": [
                        {
                            "Key": "EngineVersion",
                            "Value": "5.7.44"
                        },
                        {
                            "Key": "DBInstanceIdentifier",
                            "Value": "database-1"
                        }
                    ],
```

```
                    "ApplyModes": [
                        "immediately",
                        "next-maintenance-window"
                    ],
                    "Status": "ready",
                    "ContextAttributes": [
                        {
                            "Key": "Recommended value",
                            "Value": "5.7.44"
                        },
                        {
                            "Key": "Current engine version",
                            "Value": "5.7.42"
                        }
                    ]
                }
            ],
            "Category": "security",
            "Source": "RDS",
            "TypeDetection": "**[resource-count] resources** are not running the
 latest minor DB engine version",
            "TypeRecommendation": "Upgrade to latest engine version",
            "Impact": "Reduced database performance and data security at risk",
            "AdditionalInfo": "We recommend that you maintain your database with the
 latest DB engine minor version as this version includes the latest security and
 functionality fixes. The DB engine minor version upgrades contain only the changes
 which are backward-compatible with earlier minor versions of the same major version
 of the DB engine.",
            "Links": [
                {
                    "Text": "Upgrading an RDS DB instance engine version",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
USER_UpgradeDBInstance.Upgrading.html"
                },
                {
                    "Text": "Using Amazon RDS Blue/Green Deployments for database
 updates for Amazon Aurora",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/
AuroraUserGuide/blue-green-deployments.html"
                },
                {
                    "Text": "Using Amazon RDS Blue/Green Deployments for database
 updates for Amazon RDS",
```

```
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
blue-green-deployments.html"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的檢視和回應 Amazon RDS 建議，以及《Amazon Aurora 使用者指南》中的檢視和回應 Amazon RDS 建議。

範例 2：列出高嚴重性的資料庫建議

下列describe-db-recommendations範例列出您 AWS 帳戶中的高嚴重性資料庫建議。

```
aws rds describe-db-recommendations \
    --filters Name=severity,Values=high
```

輸出：

```
{
    "DBRecommendations": [
        {
            "RecommendationId": "12ab3cde-f456-7g8h-9012-i3j45678k9lm",
            "TypeId": "config_recommendation::rds_extended_support",
            "Severity": "high",
            "ResourceArn": "arn:aws:rds:us-west-2:111122223333:db:database-1",
            "Status": "active",
            "CreatedTime": "2024-02-21T23:14:19.392000+00:00",
            "UpdatedTime": "2024-02-21T23:14:19+00:00",
            "Detection": "Your databases will be auto-enrolled to RDS Extended
 Support on February 29",
            "Recommendation": "Upgrade your major version before February 29, 2024
 to avoid additional charges",
            "Description": "Your PostgreSQL 11 and MySQL 5.7 databases will be
 automatically enrolled into RDS Extended Support on February 29, 2024. To avoid
 the increase in charges due to RDS Extended Support, we recommend upgrading your
 databases to a newer major engine version before February 29, 2024.\nTo learn more
 about the RDS Extended Support pricing, refer to the pricing page.",
            "RecommendedActions": [
                {
                    "ActionId": "12ab34c5de6fg7h89i0jk1lm234n5678",
                    "Parameters": [],
```

```
                    "ApplyModes": [
                        "manual"
                    ],
                    "Status": "ready",
                    "ContextAttributes": []
                }
            ],
            "Category": "cost optimization",
            "Source": "RDS",
            "TypeDetection": "Your database will be auto-enrolled to RDS Extended
 Support on February 29",
            "TypeRecommendation": "Upgrade your major version before February 29,
 2024 to avoid additional charges",
            "Impact": "Increase in charges due to RDS Extended Support",
            "AdditionalInfo": "With Amazon RDS Extended Support, you can continue
 running your database on a major engine version past the RDS end of standard
 support date for an additional cost. This paid feature gives you more time to
 upgrade to a supported major engine version.\nDuring Extended Support, Amazon RDS
 will supply critical CVE patches and bug fixes.",
            "Links": [
                {
                    "Text": "Amazon RDS Extended Support pricing for RDS for MySQL",
                    "Url": "https://aws.amazon.com/rds/mysql/pricing/"
                },
                {
                    "Text": "Amazon RDS Extended Support for RDS for MySQL and
 PostgreSQL databases",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
extended-support.html"
                },
                {
                    "Text": "Amazon RDS Extended Support pricing for Amazon Aurora
 PostgreSQL",
                    "Url": "https://aws.amazon.com/rds/aurora/pricing/"
                },
                {
                    "Text": "Amazon RDS Extended Support for Aurora PostgreSQL
 databases",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/
AuroraUserGuide/extended-support.html"
                },
                {
                    "Text": "Amazon RDS Extended Support pricing for RDS for
 PostgreSQL",
```

```
                    "Url": "https://aws.amazon.com/rds/postgresql/pricing/"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的檢視和回應 Amazon RDS 建議，以及《Amazon Aurora 使用者指南》中的檢視和回應 Amazon RDS 建議。

範例 3：列出指定資料庫執行個體的資料庫建議

下列describe-db-recommendations範例列出指定資料庫執行個體的所有資料庫建議。

```
aws rds describe-db-recommendations \
    --filters Name=dbi-resource-id,Values=database-1
```

輸出：

```
{
    "DBRecommendations": [
        {
            "RecommendationId": "12ab3cde-f456-7g8h-9012-i3j45678k9lm",
            "TypeId": "config_recommendation::old_minor_version",
            "Severity": "informational",
            "ResourceArn": "arn:aws:rds:us-west-2:111122223333:db:database-1",
            "Status": "active",
            "CreatedTime": "2024-02-21T23:14:19.292000+00:00",
            "UpdatedTime": "2024-02-21T23:14:19+00:00",
            "Detection": "**[resource-name]** is not running the latest minor DB
 engine version",
            "Recommendation": "Upgrade to latest engine version",
            "Description": "Your database resources aren't running the latest minor
 DB engine version. The latest minor version contains the latest security fixes and
 other improvements.",
            "RecommendedActions": [
                {
                    "ActionId": "12ab34c5de6fg7h89i0jk1lm234n5678",
                    "Operation": "modifyDbInstance",
                    "Parameters": [
                        {
                            "Key": "EngineVersion",
                            "Value": "5.7.44"
```

```
                        },
                        {
                            "Key": "DBInstanceIdentifier",
                            "Value": "database-1"
                        }
                    ],
                    "ApplyModes": [
                        "immediately",
                        "next-maintenance-window"
                    ],
                    "Status": "ready",
                    "ContextAttributes": [
                        {
                            "Key": "Recommended value",
                            "Value": "5.7.44"
                        },
                        {
                            "Key": "Current engine version",
                            "Value": "5.7.42"
                        }
                    ]
                }
            ],
            "Category": "security",
            "Source": "RDS",
            "TypeDetection": "**[resource-count] resources** are not running the
 latest minor DB engine version",
            "TypeRecommendation": "Upgrade to latest engine version",
            "Impact": "Reduced database performance and data security at risk",
            "AdditionalInfo": "We recommend that you maintain your database with the
 latest DB engine minor version as this version includes the latest security and
 functionality fixes. The DB engine minor version upgrades contain only the changes
 which are backward-compatible with earlier minor versions of the same major version
 of the DB engine.",
            "Links": [
                {
                    "Text": "Upgrading an RDS DB instance engine version",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
USER_UpgradeDBInstance.Upgrading.html"
                },
                {
                    "Text": "Using Amazon RDS Blue/Green Deployments for database
 updates for Amazon Aurora",
```

```
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/
AuroraUserGuide/blue-green-deployments.html"
                },
                {
                    "Text": "Using Amazon RDS Blue/Green Deployments for database
 updates for Amazon RDS",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
blue-green-deployments.html"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的檢視和回應 Amazon RDS 建議，以及《Amazon Aurora 使用者指南》中的檢視和回應 Amazon RDS 建議。

範例 4：列出所有作用中的資料庫建議

下列describe-db-recommendations範例列出您 AWS 帳戶中所有作用中的資料庫建議。

```
aws rds describe-db-recommendations \
    --filters Name=status,Values=active
```

輸出：

```
{
    "DBRecommendations": [
        {
            "RecommendationId": "12ab3cde-f456-7g8h-9012-i3j45678k9lm",
            "TypeId": "config_recommendation::old_minor_version",
            "Severity": "informational",
            "ResourceArn": "arn:aws:rds:us-west-2:111122223333:db:database-1",
            "Status": "active",
            "CreatedTime": "2024-02-21T23:14:19.292000+00:00",
            "UpdatedTime": "2024-02-21T23:14:19+00:00",
            "Detection": "**[resource-name]** is not running the latest minor DB
 engine version",
            "Recommendation": "Upgrade to latest engine version",
            "Description": "Your database resources aren't running the latest minor
 DB engine version. The latest minor version contains the latest security fixes and
 other improvements.",
            "RecommendedActions": [
```

```
                {
                    "ActionId": "12ab34c5de6fg7h89i0jk1lm234n5678",
                    "Operation": "modifyDbInstance",
                    "Parameters": [
                        {
                            "Key": "EngineVersion",
                            "Value": "5.7.44"
                        },
                        {
                            "Key": "DBInstanceIdentifier",
                            "Value": "database-1"
                        }
                    ],
                    "ApplyModes": [
                        "immediately",
                        "next-maintenance-window"
                    ],
                    "Status": "ready",
                    "ContextAttributes": [
                        {
                            "Key": "Recommended value",
                            "Value": "5.7.44"
                        },
                        {
                            "Key": "Current engine version",
                            "Value": "5.7.42"
                        }
                    ]
                }
            ],
            "Category": "security",
            "Source": "RDS",
            "TypeDetection": "**[resource-count] resources** are not running the
latest minor DB engine version",
            "TypeRecommendation": "Upgrade to latest engine version",
            "Impact": "Reduced database performance and data security at risk",
            "AdditionalInfo": "We recommend that you maintain your database with the
latest DB engine minor version as this version includes the latest security and
functionality fixes. The DB engine minor version upgrades contain only the changes
which are backward-compatible with earlier minor versions of the same major version
of the DB engine.",
            "Links": [
                {
                    "Text": "Upgrading an RDS DB instance engine version",
```

```
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
USER_UpgradeDBInstance.Upgrading.html"
                },
                {
                    "Text": "Using Amazon RDS Blue/Green Deployments for database
 updates for Amazon Aurora",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/
AuroraUserGuide/blue-green-deployments.html"
                },
                {
                    "Text": "Using Amazon RDS Blue/Green Deployments for database
 updates for Amazon RDS",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
blue-green-deployments.html"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的檢視和回應 Amazon RDS 建議，以及《Amazon Aurora 使用者指南》中的檢視和回應 Amazon RDS 建議。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbRecommendations。

describe-db-security-groups

以下程式碼範例顯示如何使用 describe-db-security-groups。

AWS CLI

列出資料庫安全群組

下列describe-db-security-groups範例列出資料庫安全群組。

```
aws rds describe-db-security-groups
```

輸出：

```
{
    "DBSecurityGroups": [
        {
            "OwnerId": "123456789012",
```

```
            "DBSecurityGroupName": "default",
            "DBSecurityGroupDescription": "default",
            "EC2SecurityGroups": [],
            "IPRanges": [],
            "DBSecurityGroupArn": "arn:aws:rds:us-
west-1:111122223333:secgrp:default"
        },
        {
            "OwnerId": "123456789012",
            "DBSecurityGroupName": "mysecgroup",
            "DBSecurityGroupDescription": "My Test Security Group",
            "VpcId": "vpc-1234567f",
            "EC2SecurityGroups": [],
            "IPRanges": [],
            "DBSecurityGroupArn": "arn:aws:rds:us-
west-1:111122223333:secgrp:mysecgroup"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的列出可用的資料庫安全群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbSecurityGroups。

describe-db-shard-groups

以下程式碼範例顯示如何使用 describe-db-shard-groups。

AWS CLI

範例 1：描述資料庫碎片群組

下列describe-db-shard-groups範例會擷取資料庫碎片群組的詳細資訊。

```
aws rds describe-db-shard-groups
```

輸出：

```
{
    "DBShardGroups": [
        {
            "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",
            "DBShardGroupIdentifier": "limitless-test-shard-grp",
```

```
            "DBClusterIdentifier": "limitless-test-cluster",
            "MaxACU": 768.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": true,
            "Endpoint": "limitless-test-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        },
        {
            "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
            "DBShardGroupIdentifier": "my-db-shard-group",
            "DBClusterIdentifier": "my-sv2-cluster",
            "MaxACU": 768.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": false,
            "Endpoint": "my-sv2-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的 Amazon Aurora 資料庫叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbShardGroups。

describe-db-snapshot-attributes

以下程式碼範例顯示如何使用 describe-db-snapshot-attributes。

AWS CLI

描述資料庫快照的屬性名稱和值

下列describe-db-snapshot-attributes範例說明資料庫快照的屬性名稱和值。

```
aws rds describe-db-snapshot-attributes \
    --db-snapshot-identifier mydbsnapshot
```

輸出：

```
{
    "DBSnapshotAttributesResult": {
```

```
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123456789012",
                    "210987654321"
                ]
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的共用資料庫快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbSnapshotAttributes。

describe-db-snapshots

以下程式碼範例顯示如何使用 describe-db-snapshots。

AWS CLI

範例 1：描述資料庫執行個體的資料庫快照

下列describe-db-snapshots範例會擷取資料庫執行個體的資料庫快照詳細資訊。

```
aws rds describe-db-snapshots \
    --db-snapshot-identifier mydbsnapshot
```

輸出：

```
{
    "DBSnapshots": [
        {
            "DBSnapshotIdentifier": "mydbsnapshot",
            "DBInstanceIdentifier": "mysqldb",
            "SnapshotCreateTime": "2018-02-08T22:28:08.598Z",
            "Engine": "mysql",
            "AllocatedStorage": 20,
            "Status": "available",
            "Port": 3306,
            "AvailabilityZone": "us-east-1f",
```

```
            "VpcId": "vpc-6594f31c",
            "InstanceCreateTime": "2018-02-08T22:24:55.973Z",
            "MasterUsername": "mysqladmin",
            "EngineVersion": "5.6.37",
            "LicenseModel": "general-public-license",
            "SnapshotType": "manual",
            "OptionGroupName": "default:mysql-5-6",
            "PercentProgress": 100,
            "StorageType": "gp2",
            "Encrypted": false,
            "DBSnapshotArn": "arn:aws:rds:us-
east-1:123456789012:snapshot:mydbsnapshot",
            "IAMDatabaseAuthenticationEnabled": false,
            "ProcessorFeatures": [],
            "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的建立資料庫快照。

範例 2：尋找所拍攝的手動快照數量

下列describe-db-snapshots範例使用 --query選項中的 length 運算子，傳回在特定區域中已拍攝的手動快照數量 AWS 。

```
aws rds describe-db-snapshots \
    --snapshot-type manual \
    --query "length(*[].{DBSnapshots:SnapshotType})" \
    --region eu-central-1
```

輸出：

```
35
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的建立資料庫快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDBSnapshots。

describe-db-subnet-groups

以下程式碼範例顯示如何使用 describe-db-subnet-groups。

AWS CLI

描述資料庫子網路群組

下列describe-db-subnet-groups範例會擷取指定資料庫子網路群組的詳細資訊。

```
aws rds describe-db-subnet-groups
```

輸出：

```
{
    "DBSubnetGroups": [
        {
            "DBSubnetGroupName": "mydbsubnetgroup",
            "DBSubnetGroupDescription": "My DB Subnet Group",
            "VpcId": "vpc-971c12ee",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-d8c8e7f4",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-718fdc7d",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1f"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-cbc8e7e7",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-0ccde220",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
```

```
                    },
                    "SubnetStatus": "Active"
                }
            ],
            "DBSubnetGroupArn": "arn:aws:rds:us-
east-1:123456789012:subgrp:mydbsubnetgroup"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的 Amazon Virtual Private Cloud VPCs 和 Amazon RDS。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDbSubnetGroups。

describe-engine-default-cluster-parameters

以下程式碼範例顯示如何使用 describe-engine-default-cluster-parameters。

AWS CLI

描述 Aurora 資料庫引擎的預設引擎和系統參數資訊

下列describe-engine-default-cluster-parameters範例會擷取與 MySQL 5.7 相容之 Aurora 資料庫叢集的預設引擎和系統參數資訊的詳細資訊。

```
aws rds describe-engine-default-cluster-parameters \
    --db-parameter-group-family aurora-mysql5.7
```

輸出：

```
{
    "EngineDefaults": {
        "Parameters": [
            {
                "ParameterName": "aurora_load_from_s3_role",
                "Description": "IAM role ARN used to load data from AWS S3",
                "Source": "engine-default",
                "ApplyType": "dynamic",
                "DataType": "string",
                "IsModifiable": true,
                "SupportedEngineModes": [
```

```
                    "provisioned"
                ]
            },
            ...some output truncated...
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的使用資料庫參數群組和資料庫叢集參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEngineDefaultClusterParameters。

describe-engine-default-parameters

以下程式碼範例顯示如何使用 describe-engine-default-parameters。

AWS CLI

描述資料庫引擎的預設引擎和系統參數資訊

下列describe-engine-default-parameters範例會擷取 MySQL 5.7 資料庫執行個體的預設引擎和系統參數資訊的詳細資訊。

```
aws rds describe-engine-default-parameters \
    --db-parameter-group-family mysql5.7
```

輸出：

```
{
    "EngineDefaults": {
        "Parameters": [
            {
                "ParameterName": "allow-suspicious-udfs",
                "Description": "Controls whether user-defined functions that have
 only an xxx symbol for the main function can be loaded",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "boolean",
                "AllowedValues": "0,1",
```

```
                "IsModifiable": false
            },
            ...some output truncated...
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的使用資料庫參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEngineDefaultParameters。

describe-event-categories

以下程式碼範例顯示如何使用 describe-event-categories。

AWS CLI

描述事件類別

下列describe-event-categories範例會擷取所有可用事件來源的事件類別詳細資訊。

```
aws rds describe-event-categories
```

輸出：

```
{
    "EventCategoriesMapList": [
        {
            "SourceType": "db-instance",
            "EventCategories": [
                "deletion",
                "read replica",
                "failover",
                "restoration",
                "maintenance",
                "low storage",
                "configuration change",
                "backup",
                "creation",
                "availability",
                "recovery",
                "failure",
```

```
                "backtrack",
                "notification"
            ]
        },
        {
            "SourceType": "db-security-group",
            "EventCategories": [
                "configuration change",
                "failure"
            ]
        },
        {
            "SourceType": "db-parameter-group",
            "EventCategories": [
                "configuration change"
            ]
        },
        {
            "SourceType": "db-snapshot",
            "EventCategories": [
                "deletion",
                "creation",
                "restoration",
                "notification"
            ]
        },
        {
            "SourceType": "db-cluster",
            "EventCategories": [
                "failover",
                "failure",
                "notification"
            ]
        },
        {
            "SourceType": "db-cluster-snapshot",
            "EventCategories": [
                "backup"
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEventCategories。

describe-event-subscriptions

以下程式碼範例顯示如何使用 describe-event-subscriptions。

AWS CLI

描述事件訂閱

此範例說明目前 AWS 帳戶的所有 Amazon RDS 事件訂閱。

```
aws rds describe-event-subscriptions
```

輸出：

```
{
    "EventSubscriptionsList": [
        {
            "EventCategoriesList": [
                "backup",
                "recovery"
            ],
            "Enabled": true,
            "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-
instance-events",
            "Status": "creating",
            "SourceType": "db-instance",
            "CustomerAwsId": "123456789012",
            "SubscriptionCreationTime": "2018-07-31 23:22:01.893",
            "CustSubscriptionId": "my-instance-events",
            "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events"
        },
        ...some output truncated...
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEventSubscriptions。

describe-events

以下程式碼範例顯示如何使用 describe-events。

AWS CLI

描述事件

下列describe-events範例會擷取指定資料庫執行個體所發生事件的詳細資訊。

```
aws rds describe-events \
    --source-identifier test-instance \
    --source-type db-instance
```

輸出：

```
{
    "Events": [
        {
            "SourceType": "db-instance",
            "SourceIdentifier": "test-instance",
            "EventCategories": [
                "backup"
            ],
            "Message": "Backing up DB instance",
            "Date": "2018-07-31T23:09:23.983Z",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"
        },
        {
            "SourceType": "db-instance",
            "SourceIdentifier": "test-instance",
            "EventCategories": [
                "backup"
            ],
            "Message": "Finished DB Instance backup",
            "Date": "2018-07-31T23:15:13.049Z",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEvents。

describe-export-tasks

以下程式碼範例顯示如何使用 describe-export-tasks。

AWS CLI

描述快照匯出任務

下列describe-export-tasks範例會傳回快照匯出至 Amazon S3 的相關資訊。

```
aws rds describe-export-tasks
```

輸出：

```
{
    "ExportTasks": [
        {
            "ExportTaskIdentifier": "test-snapshot-export",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:snapshot:test-
snapshot",
            "SnapshotTime": "2020-03-02T18:26:28.163Z",
            "TaskStartTime": "2020-03-02T18:57:56.896Z",
            "TaskEndTime": "2020-03-02T19:10:31.985Z",
            "S3Bucket": "amzn-s3-demo-bucket",
            "S3Prefix": "",
            "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/ExportRole",
            "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
abcd0000-7fca-4128-82f2-aabbccddeeff",
            "Status": "COMPLETE",
            "PercentProgress": 100,
            "TotalExtractedDataInGB": 0
        },
        {
            "ExportTaskIdentifier": "my-s3-export",
            "SourceArn": "arn:aws:rds:us-west-2:123456789012:snapshot:db5-snapshot-
test",
            "SnapshotTime": "2020-03-27T20:48:42.023Z",
            "S3Bucket": "amzn-s3-demo-bucket",
            "S3Prefix": "",
            "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/ExportRole",
            "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
abcd0000-7fca-4128-82f2-aabbccddeeff",
            "Status": "STARTING",
            "PercentProgress": 0,
            "TotalExtractedDataInGB": 0
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的監控快照匯出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeExportTasks。

describe-global-clusters

以下程式碼範例顯示如何使用 describe-global-clusters。

AWS CLI

描述全域資料庫叢集

下列describe-global-clusters範例列出目前 AWS 區域中的 Aurora 全域資料庫叢集。

```
aws rds describe-global-clusters
```

輸出：

```
{
    "GlobalClusters": [
        {
            "GlobalClusterIdentifier": "myglobalcluster",
            "GlobalClusterResourceId": "cluster-f5982077e3b5aabb",
            "GlobalClusterArn": "arn:aws:rds::123456789012:global-
cluster:myglobalcluster",
            "Status": "available",
            "Engine": "aurora-mysql",
            "EngineVersion": "5.7.mysql_aurora.2.07.2",
            "StorageEncrypted": false,
            "DeletionProtection": false,
            "GlobalClusterMembers": []
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的管理 Aurora 全域資料庫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeGlobalClusters。

describe-option-group-options

以下程式碼範例顯示如何使用 describe-option-group-options。

AWS CLI

描述所有可用的選項

下列describe-option-group-options範例列出 Oracle Database 19c 執行個體的兩個選項。

```
aws rds describe-option-group-options \
    --engine-name oracle-ee \
    --major-engine-version 19 \
    --max-items 2
```

輸出：

```
{
    "OptionGroupOptions": [
        {
            "Name": "APEX",
            "Description": "Oracle Application Express Runtime Environment",
            "EngineName": "oracle-ee",
            "MajorEngineVersion": "19",
            "MinimumRequiredMinorEngineVersion": "0.0.0.ru-2019-07.rur-2019-07.r1",
            "PortRequired": false,
            "OptionsDependedOn": [],
            "OptionsConflictsWith": [],
            "Persistent": false,
            "Permanent": false,
            "RequiresAutoMinorEngineVersionUpgrade": false,
            "VpcOnly": false,
            "SupportsOptionVersionDowngrade": false,
            "OptionGroupOptionSettings": [],
            "OptionGroupOptionVersions": [
                {
                    "Version": "19.1.v1",
                    "IsDefault": true
                },
                {
                    "Version": "19.2.v1",
                    "IsDefault": false
                }
```

```
            ]
        },
        {
            "Name": "APEX-DEV",
            "Description": "Oracle Application Express Development Environment",
            "EngineName": "oracle-ee",
            "MajorEngineVersion": "19",
            "MinimumRequiredMinorEngineVersion": "0.0.0.ru-2019-07.rur-2019-07.r1",
            "PortRequired": false,
            "OptionsDependedOn": [
                "APEX"
            ],
            "OptionsConflictsWith": [],
            "Persistent": false,
            "Permanent": false,
            "RequiresAutoMinorEngineVersionUpgrade": false,
            "VpcOnly": false,
            "OptionGroupOptionSettings": []
        }
    ],
    "NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的列出選項群組的選項和選項設定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeOptionGroupOptions。

describe-option-groups

以下程式碼範例顯示如何使用 describe-option-groups。

AWS CLI

描述可用的選項群組

下列describe-option-groups範例列出 Oracle Database 19c 執行個體的選項群組。

```
aws rds describe-option-groups \
    --engine-name oracle-ee \
    --major-engine-version 19
```

輸出：

```
{
    "OptionGroupsList": [
        {
            "OptionGroupName": "default:oracle-ee-19",
            "OptionGroupDescription": "Default option group for oracle-ee 19",
            "EngineName": "oracle-ee",
            "MajorEngineVersion": "19",
            "Options": [],
            "AllowsVpcAndNonVpcInstanceMemberships": true,
            "OptionGroupArn": "arn:aws:rds:us-west-1:111122223333:og:default:oracle-
ee-19"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的列出選項群組的選項和選項設定。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeOptionGroups。

describe-orderable-db-instance-options

以下程式碼範例顯示如何使用 describe-orderable-db-instance-options。

AWS CLI

描述可排序的資料庫執行個體選項

下列describe-orderable-db-instance-options範例會擷取執行 MySQL 資料庫引擎之資料庫執行個體可排序選項的詳細資訊。

```
aws rds describe-orderable-db-instance-options \
    --engine mysql
```

輸出：

```
{
    "OrderableDBInstanceOptions": [
        {
            "MinStorageSize": 5,
            "ReadReplicaCapable": true,
            "MaxStorageSize": 6144,
            "AvailabilityZones": [
```

```
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
            ],
            "SupportsIops": false,
            "AvailableProcessorFeatures": [],
            "MultiAZCapable": true,
            "DBInstanceClass": "db.m1.large",
            "Vpc": true,
            "StorageType": "gp2",
            "LicenseModel": "general-public-license",
            "EngineVersion": "5.5.46",
            "SupportsStorageEncryption": false,
            "SupportsEnhancedMonitoring": true,
            "Engine": "mysql",
            "SupportsIAMDatabaseAuthentication": false,
            "SupportsPerformanceInsights": false
        }
    ]
    ...some output truncated...
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeOrderableDBInstanceOptions。

describe-pending-maintenance-actions

以下程式碼範例顯示如何使用 describe-pending-maintenance-actions。

AWS CLI

列出至少有一個待定維護動作的資源

下列describe-pending-maintenance-actions範例列出資料庫執行個體的待定維護動作。

```
aws rds describe-pending-maintenance-actions
```

輸出：

```
{
    "PendingMaintenanceActions": [
        {
            "ResourceIdentifier": "arn:aws:rds:us-
west-2:123456789012:cluster:global-db1-cl1",
            "PendingMaintenanceActionDetails": [
                {
                    "Action": "system-update",
                    "Description": "Upgrade to Aurora PostgreSQL 2.4.2"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的維護資料庫執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePendingMaintenanceActions。

describe-reserved-db-instances-offerings

以下程式碼範例顯示如何使用 describe-reserved-db-instances-offerings。

AWS CLI

描述預留資料庫執行個體方案

下列describe-reserved-db-instances-offerings範例會擷取 預留資料庫執行個體選項的詳細資訊oracle。

```
aws rds describe-reserved-db-instances-offerings \
    --product-description oracle
```

輸出：

```
{
```

```
    "ReservedDBInstancesOfferings": [
        {
            "CurrencyCode": "USD",
            "UsagePrice": 0.0,
            "ProductDescription": "oracle-se2(li)",
            "ReservedDBInstancesOfferingId": "005bdee3-9ef4-4182-aa0c-58ef7cb6c2f8",
            "MultiAZ": true,
            "DBInstanceClass": "db.m4.xlarge",
            "OfferingType": "Partial Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": 0.594,
                    "RecurringChargeFrequency": "Hourly"
                }
            ],
            "FixedPrice": 4089.0,
            "Duration": 31536000
        },
    ...some output truncated...
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReservedDbInstancesOfferings。

describe-reserved-db-instances

以下程式碼範例顯示如何使用 describe-reserved-db-instances。

AWS CLI

描述預留資料庫執行個體

下列describe-reserved-db-instances範例會擷取目前 AWS 帳戶中任何預留資料庫執行個體的詳細資訊。

```
aws rds describe-reserved-db-instances
```

輸出：

```
{
    "ReservedDBInstances": [
        {
```

```
            "ReservedDBInstanceId": "myreservedinstance",
            "ReservedDBInstancesOfferingId": "12ab34cd-59af-4b2c-a660-1abcdef23456",
            "DBInstanceClass": "db.t3.micro",
            "StartTime": "2020-06-01T13:44:21.436Z",
            "Duration": 31536000,
            "FixedPrice": 0.0,
            "UsagePrice": 0.0,
            "CurrencyCode": "USD",
            "DBInstanceCount": 1,
            "ProductDescription": "sqlserver-ex(li)",
            "OfferingType": "No Upfront",
            "MultiAZ": false,
            "State": "payment-pending",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": 0.014,
                    "RecurringChargeFrequency": "Hourly"
                }
            ],
            "ReservedDBInstanceArn": "arn:aws:rds:us-
west-2:123456789012:ri:myreservedinstance",
            "LeaseId": "a1b2c3d4-6b69-4a59-be89-5e11aa446666"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的 Amazon RDS 的預留資料庫執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReservedDbInstances。

describe-source-regions

以下程式碼範例顯示如何使用 describe-source-regions。

AWS CLI

描述來源區域

下列describe-source-regions範例會擷取所有來源 AWS 區域的詳細資訊。它還顯示，自動備份只能從美國西部（奧勒岡）複寫到目的地 AWS 區域美國東部（維吉尼亞北部）。

```
aws rds describe-source-regions \
    --region us-east-1
```

輸出：

```
{
    "SourceRegions": [
        {
            "RegionName": "af-south-1",
            "Endpoint": "https://rds.af-south-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        },
        {
            "RegionName": "ap-east-1",
            "Endpoint": "https://rds.ap-east-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        },
        {
            "RegionName": "ap-northeast-1",
            "Endpoint": "https://rds.ap-northeast-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "ap-northeast-2",
            "Endpoint": "https://rds.ap-northeast-2.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "ap-northeast-3",
            "Endpoint": "https://rds.ap-northeast-3.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        },
        {
            "RegionName": "ap-south-1",
            "Endpoint": "https://rds.ap-south-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "ap-southeast-1",
            "Endpoint": "https://rds.ap-southeast-1.amazonaws.com",
            "Status": "available",
```

```
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "ap-southeast-2",
            "Endpoint": "https://rds.ap-southeast-2.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "ap-southeast-3",
            "Endpoint": "https://rds.ap-southeast-3.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        },
        {
            "RegionName": "ca-central-1",
            "Endpoint": "https://rds.ca-central-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "eu-north-1",
            "Endpoint": "https://rds.eu-north-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "eu-south-1",
            "Endpoint": "https://rds.eu-south-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        },
        {
            "RegionName": "eu-west-1",
            "Endpoint": "https://rds.eu-west-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "eu-west-2",
            "Endpoint": "https://rds.eu-west-2.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
```

```
        {
            "RegionName": "eu-west-3",
            "Endpoint": "https://rds.eu-west-3.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "me-central-1",
             "Endpoint": "https://rds.me-central-1.amazonaws.com",
             "Status": "available",
             "SupportsDBInstanceAutomatedBackupsReplication": false
        },
        {
            "RegionName": "me-south-1",
            "Endpoint": "https://rds.me-south-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        },
        {
            "RegionName": "sa-east-1",
            "Endpoint": "https://rds.sa-east-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "us-east-2",
            "Endpoint": "https://rds.us-east-2.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "us-west-1",
            "Endpoint": "https://rds.us-west-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "us-west-2",
           "Endpoint": "https://rds.us-west-2.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的尋找有關複寫備份的資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSourceRegions。

describe-valid-db-instance-modifications

以下程式碼範例顯示如何使用 describe-valid-db-instance-modifications。

AWS CLI

描述資料庫執行個體的有效修改

下列describe-valid-db-instance-modifications範例會擷取指定資料庫執行個體之有效修改的詳細資訊。

```
aws rds describe-valid-db-instance-modifications \
    --db-instance-identifier test-instance
```

輸出：

```
{
    "ValidDBInstanceModificationsMessage": {
        "ValidProcessorFeatures": [],
        "Storage": [
            {
                "StorageSize": [
                    {
                        "Step": 1,
                        "To": 20,
                        "From": 20
                    },
                    {
                        "Step": 1,
                        "To": 6144,
                        "From": 22
                    }
                ],
                "ProvisionedIops": [
                    {
                        "Step": 1,
                        "To": 0,
```

```
                    "From": 0
                }
            ],
            "IopsToStorageRatio": [
                {
                    "To": 0.0,
                    "From": 0.0
                }
            ],
            "StorageType": "gp2"
        },
        {
            "StorageSize": [
                {
                    "Step": 1,
                    "To": 6144,
                    "From": 100
                }
            ],
            "ProvisionedIops": [
                {
                    "Step": 1,
                    "To": 40000,
                    "From": 1000
                }
            ],
            "IopsToStorageRatio": [
                {
                    "To": 50.0,
                    "From": 1.0
                }
            ],
            "StorageType": "io1"
        },
        {
            "StorageSize": [
                {
                    "Step": 1,
                    "To": 20,
                    "From": 20
                },
                {
                    "Step": 1,
                    "To": 3072,
```

```
                    "From": 22
                }
            ],
            "ProvisionedIops": [
                {
                    "Step": 1,
                    "To": 0,
                    "From": 0
                }
            ],
            "IopsToStorageRatio": [
                {
                    "To": 0.0,
                    "From": 0.0
                }
            ],
            "StorageType": "magnetic"
        }
    ]
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeValidDbInstanceModifications。

download-db-log-file-portion

以下程式碼範例顯示如何使用 download-db-log-file-portion。

AWS CLI

範例 1：下載資料庫日誌檔案的最新部分

下列download-db-log-file-portion範例只會下載日誌檔案的最新部分，並將其儲存至名為的本機檔案tail.txt。

```
aws rds download-db-log-file-portion \
    --db-instance-identifier test-instance \
    --log-file-name log.txt \
    --output text > tail.txt
```

儲存的檔案可能包含空白行。下載時，它們會出現在日誌檔案每個部分的結尾。

範例 2：下載整個資料庫日誌檔案

下列download-db-log-file-portion範例使用 --starting-token 0 參數下載整個日誌檔案，並將輸出儲存至名為 的本機檔案full.txt。

```
aws rds download-db-log-file-portion \
    --db-instance-identifier test-instance \
    --log-file-name log.txt \
    --starting-token 0 \
    --output text > full.txt
```

儲存的檔案可能包含空白行。下載時，它們會出現在日誌檔案每個部分的結尾。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DownloadDbLogFilePortion。

generate-auth-token

以下程式碼範例顯示如何使用 generate-auth-token。

AWS CLI

產生身分驗證字符

下列generate-db-auth-token範例會產生身分驗證字符，以搭配 IAM 資料庫身分驗證使用。

```
aws rds generate-db-auth-token \
    --hostname aurmysql-test.cdgmuqiadpid.us-west-2.rds.amazonaws.com \
    --port 3306 \
    --region us-east-1 \
    --username jane_doe
```

輸出：

```
aurmysql-test.cdgmuqiadpid.us-west-2.rds.amazonaws.com:3306/?
Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIESZCNJ3OEXAMPLE%2F20180731%2Fus-east-1%2Frds-db%2Faws4_request&X-
Amz-Date=20180731T235209Z&X-Amz-Expires=900&X-Amz-SignedHeaders=host&X-Amz-
Signature=5a8753ebEXAMPLEa2c724e5667797EXAMPLE9d6ec6e3f427191fa41aeEXAMPLE
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GenerateAuthToken。

generate-db-auth-token

以下程式碼範例顯示如何使用 generate-db-auth-token。

AWS CLI

產生 IAM 身分驗證字符

下列generate-db-auth-token範例會產生 IAM 身分驗證字符以連線至資料庫。

```
aws rds generate-db-auth-token \
    --hostname mydb.123456789012.us-east-1.rds.amazonaws.com \
    --port 3306 \
    --region us-east-1 \
    --username db_user
```

輸出：

```
mydb.123456789012.us-east-1.rds.amazonaws.com:3306/?
Action=connect&DBUser=db_user&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIEXAMPLE%2Fus-east-1%2Frds-db%2Faws4_request&X-Amz-
Date=20210123T011543Z&X-Amz-Expires=900&X-Amz-SignedHeaders=host&X-Amz-
Signature=88987EXAMPLE1EXAMPLE2EXAMPLE3EXAMPLE4EXAMPLE5EXAMPLE6
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的使用 IAM 身分驗證連線至資料庫執行個體，以及《Amazon Aurora 使用者指南》中的使用 IAM 身分驗證連線至資料庫叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GenerateDbAuthToken。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出 Amazon RDS 資源上的標籤

下列list-tags-for-resource範例列出資料庫執行個體上的所有標籤。

```
aws rds list-tags-for-resource \
    --resource-name arn:aws:rds:us-east-1:123456789012:db:orcl1
```

輸出：

```
{
    "TagList": [
        {
            "Key": "Environment",
            "Value": "test"
        },
        {
            "Key": "Name",
            "Value": "MyDatabase"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的標記 Amazon RDS 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

modify-certificates

以下程式碼範例顯示如何使用 modify-certificates。

AWS CLI

暫時覆寫新資料庫執行個體的系統預設 SSL/TLS 憑證

下列modify-certificates範例會暫時覆寫新資料庫執行個體的系統預設 SSL/TLS 憑證。

```
aws rds modify-certificates \
    --certificate-identifier rds-ca-2019
```

輸出：

```
{
    "Certificate": {
        "CertificateIdentifier": "rds-ca-2019",
        "CertificateType": "CA",
        "Thumbprint": "EXAMPLE123456789012",
        "ValidFrom": "2019-09-19T18:16:53Z",
        "ValidTill": "2024-08-22T17:08:50Z",
        "CertificateArn": "arn:aws:rds:us-east-1::cert:rds-ca-2019",
        "CustomerOverride": true,
```

```
        "CustomerOverrideValidTill": "2024-08-22T17:08:50Z"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的輪換您的 SSL/TLS 憑證和《Amazon Aurora 使用者指南》中的輪換您的 SSL/TLS 憑證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyCertificates。

`modify-current-db-cluster-capacity`

以下程式碼範例顯示如何使用 `modify-current-db-cluster-capacity`。

AWS CLI

擴展 Aurora Serverless 資料庫叢集的容量

下列`modify-current-db-cluster-capacity`範例會將 Aurora Serverless 資料庫叢集的容量擴展至 8。

```
aws rds modify-current-db-cluster-capacity \
    --db-cluster-identifier mydbcluster \
    --capacity 8
```

輸出：

```
{
    "DBClusterIdentifier": "mydbcluster",
    "PendingCapacity": 8,
    "CurrentCapacity": 1,
    "SecondsBeforeTimeout": 300,
    "TimeoutAction": "ForceApplyCapacityChange"
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的手動擴展 Aurora Serverless v1 資料庫叢集容量。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyCurrentDbClusterCapacity。

`modify-db-cluster-endpoint`

以下程式碼範例顯示如何使用 `modify-db-cluster-endpoint`。

AWS CLI

修改自訂資料庫叢集端點

下列modify-db-cluster-endpoint範例會修改指定的自訂資料庫叢集端點。

```
aws rds modify-db-cluster-endpoint \
    --db-cluster-endpoint-identifier mycustomendpoint \
    --static-members dbinstance1 dbinstance2 dbinstance3
```

輸出：

```
{
    "DBClusterEndpointIdentifier": "mycustomendpoint",
    "DBClusterIdentifier": "mydbcluster",
    "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",
    "Endpoint": "mycustomendpoint.cluster-custom-cnpexample.us-
east-1.rds.amazonaws.com",
    "Status": "modifying",
    "EndpointType": "CUSTOM",
    "CustomEndpointType": "READER",
    "StaticMembers": [
        "dbinstance1",
        "dbinstance2",
        "dbinstance3"
    ],
    "ExcludedMembers": [],
    "DBClusterEndpointArn": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:mycustomendpoint"
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的 Amazon Aurora Connection Management。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDbClusterEndpoint。

modify-db-cluster-parameter-group

以下程式碼範例顯示如何使用 modify-db-cluster-parameter-group。

AWS CLI

修改資料庫叢集參數群組中的參數

下列modify-db-cluster-parameter-group範例會修改資料庫叢集參數群組中的參數值。

```
aws rds modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterpg \
    --
parameters "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate"
 \

  "ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

輸出：

```
{
    "DBClusterParameterGroupName": "mydbclusterpg"
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的使用資料庫參數群組和資料庫叢集參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDbClusterParameterGroup。

modify-db-cluster-snapshot-attribute

以下程式碼範例顯示如何使用 modify-db-cluster-snapshot-attribute。

AWS CLI

修改資料庫叢集快照屬性

下列modify-db-cluster-snapshot-attribute範例會變更指定的資料庫叢集快照屬性。

```
aws rds modify-db-cluster-snapshot-attribute \
    --db-cluster-snapshot-identifier myclustersnapshot \
    --attribute-name restore \
    --values-to-add 123456789012
```

輸出：

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotIdentifier": "myclustersnapshot",
        "DBClusterSnapshotAttributes": [
```

```
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123456789012"
                ]
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的從資料庫叢集快照還原。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDbClusterSnapshotAttribute。

modify-db-cluster

以下程式碼範例顯示如何使用 modify-db-cluster。

AWS CLI

範例 1：修改資料庫叢集

下列modify-db-cluster範例會變更名為 之資料庫叢集的主要使用者密碼，cluster-2並將備份保留期設定為 14 天。--apply-immediately 參數會立即進行變更，而不是等到下一個維護時段。

```
aws rds modify-db-cluster \
    --db-cluster-identifier cluster-2 \
    --backup-retention-period 14 \
    --master-user-password newpassword99 \
    --apply-immediately
```

輸出：

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "eu-central-1b",
            "eu-central-1c",
            "eu-central-1a"
        ],
        "BackupRetentionPeriod": 14,
```

```
        "DatabaseName": "",
        "DBClusterIdentifier": "cluster-2",
        "DBClusterParameterGroup": "default.aurora5.6",
        "DBSubnetGroup": "default-vpc-2305ca49",
        "Status": "available",
        "EarliestRestorableTime": "2020-06-03T02:07:29.637Z",
        "Endpoint": "cluster-2.cluster-############.eu-central-1.rds.amazonaws.com",
        "ReaderEndpoint": "cluster-2.cluster-ro-############.eu-
central-1.rds.amazonaws.com",
        "MultiAZ": false,
        "Engine": "aurora",
        "EngineVersion": "5.6.10a",
        "LatestRestorableTime": "2020-06-04T15:11:25.748Z",
        "Port": 3306,
        "MasterUsername": "admin",
        "PreferredBackupWindow": "01:55-02:25",
        "PreferredMaintenanceWindow": "thu:21:14-thu:21:44",
        "ReadReplicaIdentifiers": [],
        "DBClusterMembers": [
            {
                "DBInstanceIdentifier": "cluster-2-instance-1",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            }
        ],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-20a5c047",
                "Status": "active"
            }
        ],
        "HostedZoneId": "Z1RLNU0EXAMPLE",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:eu-central-1:123456789012:key/
d1bd7c8f-5cdb-49ca-8a62-a1b2c3d4e5f6",
        "DbClusterResourceId": "cluster-AGJ7XI77XVIS6FUXHU1EXAMPLE",
        "DBClusterArn": "arn:aws:rds:eu-central-1:123456789012:cluster:cluster-2",
        "AssociatedRoles": [],
        "IAMDatabaseAuthenticationEnabled": false,
        "ClusterCreateTime": "2020-04-03T14:44:02.764Z",
        "EngineMode": "provisioned",
        "DeletionProtection": false,
        "HttpEndpointEnabled": false,
```

```
        "CopyTagsToSnapshot": true,
        "CrossAccountClone": false,
        "DomainMemberships": []
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的修改 Amazon Aurora 資料庫叢集。

範例 2：將 VPC 安全群組與資料庫叢集建立關聯

下列modify-db-instance範例會建立特定 VPC 安全群組的關聯，並從資料庫叢集中移除資料庫安全群組。

```
aws rds modify-db-cluster \
    --db-cluster-identifier dbName \
    --vpc-security-group-ids sg-ID
```

輸出：

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-west-2c",
            "us-west-2b",
            "us-west-2a"
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "dbName",
        "DBClusterParameterGroup": "default.aurora-mysql8.0",
        "DBSubnetGroup": "default",
        "Status": "available",
        "EarliestRestorableTime": "2024-02-15T01:12:13.966000+00:00",
        "Endpoint": "dbName.cluster-abcdefghji.us-west-2.rds.amazonaws.com",
        "ReaderEndpoint": "dbName.cluster-ro-abcdefghji.us-
west-2.rds.amazonaws.com",
        "MultiAZ": false,
        "Engine": "aurora-mysql",
        "EngineVersion": "8.0.mysql_aurora.3.04.1",
        "LatestRestorableTime": "2024-02-15T02:25:33.696000+00:00",
        "Port": 3306,
        "MasterUsername": "admin",
```

```
        "PreferredBackupWindow": "10:59-11:29",
        "PreferredMaintenanceWindow": "thu:08:54-thu:09:24",
        "ReadReplicaIdentifiers": [],
        "DBClusterMembers": [
            {
                "DBInstanceIdentifier": "dbName-instance-1",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            }
        ],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-ID",
                "Status": "active"
            }
        ],
        ...output omitted...
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的使用安全群組控制存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDbCluster。

modify-db-instance

以下程式碼範例顯示如何使用 modify-db-instance。

AWS CLI

範例 1：修改資料庫執行個體

下列modify-db-instance範例會將選項群組和參數群組與相容的 Microsoft SQL Server 資料庫執行個體建立關聯。--apply-immediately 參數會導致選項和參數群組立即建立關聯，而不是等到下一個維護時段。

```
aws rds modify-db-instance \
    --db-instance-identifier database-2 \
    --option-group-name test-se-2017 \
    --db-parameter-group-name test-sqlserver-se-2017 \
    --apply-immediately
```

輸出：

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "database-2",
        "DBInstanceClass": "db.r4.large",
        "Engine": "sqlserver-se",
        "DBInstanceStatus": "available",

        ...output omitted...

        "DBParameterGroups": [
            {
                "DBParameterGroupName": "test-sqlserver-se-2017",
                "ParameterApplyStatus": "applying"
            }
        ],
        "AvailabilityZone": "us-west-2d",

        ...output omitted...

        "MultiAZ": true,
        "EngineVersion": "14.00.3281.6.v1",
        "AutoMinorVersionUpgrade": false,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "license-included",
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "test-se-2017",
                "Status": "pending-apply"
            }
        ],
        "CharacterSetName": "SQL_Latin1_General_CP1_CI_AS",
        "SecondaryAvailabilityZone": "us-west-2c",
        "PubliclyAccessible": true,
        "StorageType": "gp2",

        ...output omitted...

        "DeletionProtection": false,
        "AssociatedRoles": [],
        "MaxAllocatedStorage": 1000
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的修改 Amazon RDS 資料庫執行個體。

範例 2：將 VPC 安全群組與資料庫執行個體建立關聯

下列modify-db-instance範例會建立特定 VPC 安全群組的關聯，並從資料庫執行個體移除資料庫安全群組：

```
aws rds modify-db-instance \
    --db-instance-identifier dbName \
    --vpc-security-group-ids sg-ID
```

輸出：

```
{
"DBInstance": {
    "DBInstanceIdentifier": "dbName",
    "DBInstanceClass": "db.t3.micro",
    "Engine": "mysql",
    "DBInstanceStatus": "available",
    "MasterUsername": "admin",
    "Endpoint": {
        "Address": "dbName.abcdefghijk.us-west-2.rds.amazonaws.com",
        "Port": 3306,
        "HostedZoneId": "ABCDEFGHIJK1234"
    },
    "AllocatedStorage": 20,
    "InstanceCreateTime": "2024-02-15T00:37:58.793000+00:00",
    "PreferredBackupWindow": "11:57-12:27",
    "BackupRetentionPeriod": 7,
    "DBSecurityGroups": [],
    "VpcSecurityGroups": [
        {
            "VpcSecurityGroupId": "sg-ID",
            "Status": "active"
        }
    ],
    ... output omitted ...
    "MultiAZ": false,
    "EngineVersion": "8.0.35",
    "AutoMinorVersionUpgrade": true,
    "ReadReplicaDBInstanceIdentifiers": [],
```

```
    "LicenseModel": "general-public-license",

    ... output ommited ...
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的使用安全群組控制存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDBInstance。

modify-db-parameter-group

以下程式碼範例顯示如何使用 modify-db-parameter-group。

AWS CLI

修改資料庫參數群組

下列modify-db-parameter-group範例會變更資料庫clr enabled參數群組中 參數的值。--apply-immediately 參數會立即修改資料庫參數群組，而不是等到下一個維護時段。

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name test-sqlserver-se-2017 \
    --parameters "ParameterName='clr
 enabled',ParameterValue=1,ApplyMethod=immediate"
```

輸出：

```
{
    "DBParameterGroupName": "test-sqlserver-se-2017"
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的修改資料庫參數群組中的參數。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDBParameterGroup。

modify-db-proxy-endpoint

以下程式碼範例顯示如何使用 modify-db-proxy-endpoint。

AWS CLI

修改 RDS 資料庫的資料庫代理端點

下列modify-db-proxy-endpoint範例修改資料庫代理端點proxyEndpoint，將讀取逾時設定為 65 秒。

```
aws rds modify-db-proxy-endpoint \
    --db-proxy-endpoint-name proxyEndpoint \
    --cli-read-timeout 65
```

輸出：

```
{
"DBProxyEndpoint":
    {
        "DBProxyEndpointName": "proxyEndpoint",
        "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-
endpoint:prx-endpoint-0123a01b12345c0ab",
        "DBProxyName": "proxyExample",
        "Status": "available",
        "VpcId": "vpc-1234567",
        "VpcSecurityGroupIds": [
            "sg-1234"
        ],
        "VpcSubnetIds": [
            "subnetgroup1",
            "subnetgroup2"
        ],
        "Endpoint": "proxyEndpoint.endpoint.proxyExample-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
        "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
        "TargetRole": "READ_WRITE",
        "IsDefault": "false"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的修改代理端點和《Amazon Aurora 使用者指南》中的修改代理端點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDbProxyEndpoint。

modify-db-proxy-target-group

以下程式碼範例顯示如何使用 modify-db-proxy-target-group。

AWS CLI

修改資料庫代理端點

下列modify-db-proxy-target-group範例修改資料庫代理目標群組，將最大連線數設定為 80%，將最大閒置連線數設定為 10%。

```
aws rds modify-db-proxy-target-group \
    --target-group-name default \
    --db-proxy-name proxyExample \
    --connection-pool-config MaxConnectionsPercent=80,MaxIdleConnectionsPercent=10
```

輸出：

```
{
"DBProxyTargetGroup":
    {
        "DBProxyName": "proxyExample",
        "TargetGroupName": "default",
        "TargetGroupArn": "arn:aws:rds:us-east-1:123456789012:target-group:prx-
tg-0123a01b12345c0ab",
        "IsDefault": true,
        "Status": "available",
        "ConnectionPoolConfig": {
            "MaxConnectionsPercent": 80,
            "MaxIdleConnectionsPercent": 10,
            "ConnectionBorrowTimeout": 120,
            "SessionPinningFilters": []
        },
        "CreatedDate": "2023-05-02T18:41:19.495000+00:00",
        "UpdatedDate": "2023-05-02T18:41:21.762000+00:00"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的修改 RDS Proxy 和《Amazon Aurora 使用者指南》中的修改 RDS Proxy。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDbProxyTargetGroup。

modify-db-proxy

以下程式碼範例顯示如何使用 modify-db-proxy。

AWS CLI

修改 RDS 資料庫的資料庫代理

下列modify-db-proxy範例會修改名為 的資料庫代理proxyExample，以要求其連線使用 SSL。

```
aws rds modify-db-proxy \
    --db-proxy-name proxyExample \
    --require-tls
```

輸出：

```
{
"DBProxy":
    {
        "DBProxyName": "proxyExample",
        "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-0123a01b12345c0ab",
        "Status": "modifying"
        "EngineFamily": "PostgreSQL",
        "VpcId": "sg-1234567",
        "VpcSecurityGroupIds": [
            "sg-1234"
        ],
        "VpcSubnetIds": [
            "subnetgroup1",
            "subnetgroup2"
        ],
        "Auth": "[
            {
                "Description": "proxydescription1",
                "AuthScheme": "SECRETS",
                "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret:proxysecret1-Abcd1e",
                "IAMAuth": "DISABLED"
            }
        ]",
        "RoleArn": "arn:aws:iam::12345678912:role/ProxyPostgreSQLRole",
        "Endpoint": "proxyExample.proxy-ab0cd1efghij.us-east-1.rds.amazonaws.com",
        "RequireTLS": true,
        "IdleClientTimeout": 1800,
        "DebuggingLogging": false,
```

```
        "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
        "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的修改 RDS Proxy 和《Amazon Aurora 使用者指南》中的建立 RDS Proxy。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDbProxy。

modify-db-shard-group

以下程式碼範例顯示如何使用 modify-db-shard-group。

AWS CLI

範例 1：修改資料庫碎片群組

下列modify-db-shard-group範例會變更資料庫碎片群組的最大容量。

```
aws rds modify-db-shard-group \
    --db-shard-group-identifier my-db-shard-group \
    --max-acu 1000
```

輸出：

```
{
    "DBShardGroups": [
        {
            "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
            "DBShardGroupIdentifier": "my-db-shard-group",
            "DBClusterIdentifier": "my-sv2-cluster",
            "MaxACU": 768.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": false,
            "Endpoint": "my-sv2-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的 Amazon Aurora 資料庫叢集。

範例 2：描述資料庫碎片群組

下列describe-db-shard-groups範例會在您執行 modify-db-shard-group命令後擷取資料庫碎片群組的詳細資訊。資料庫碎片群組的最大容量現在my-db-shard-group是 1000 個 Aurora 容量單位 (ACUs)。

```
aws rds describe-db-shard-groups
```

輸出：

```
{
    "DBShardGroups": [
        {
            "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",
            "DBShardGroupIdentifier": "limitless-test-shard-grp",
            "DBClusterIdentifier": "limitless-test-cluster",
            "MaxACU": 768.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": true,
            "Endpoint": "limitless-test-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        },
        {
            "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
            "DBShardGroupIdentifier": "my-db-shard-group",
            "DBClusterIdentifier": "my-sv2-cluster",
            "MaxACU": 1000.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": false,
            "Endpoint": "my-sv2-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的 Amazon Aurora 資料庫叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDbShardGroup。

modify-db-snapshot-attribute

以下程式碼範例顯示如何使用 modify-db-snapshot-attribute。

AWS CLI

範例 1：啟用兩個 AWS 帳戶以還原資料庫快照

下列modify-db-snapshot-attribute範例會將許可授予兩個 AWS 帳戶，其中包含識別符 111122223333和 444455556666，以還原名為 的資料庫快照mydbsnapshot。

```
aws rds modify-db-snapshot-attribute \
    --db-snapshot-identifier mydbsnapshot \
    --attribute-name restore \
    --values-to-add {"111122223333","444455556666"}
```

輸出：

```
{
    "DBSnapshotAttributesResult": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "111122223333",
                    "444455556666"
                ]
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的共用快照。

範例 2：防止 AWS 帳戶還原資料庫快照

下列modify-db-snapshot-attribute範例會移除特定 AWS 帳戶的許可，以還原名為 的資料庫快照mydbsnapshot。指定單一帳戶時，帳戶識別符不能被引號或括號包圍。

```
aws rds modify-db-snapshot-attribute \
    --db-snapshot-identifier mydbsnapshot \
    --attribute-name restore \
```

```
    --values-to-remove 444455556666
```

輸出：

```
{
    "DBSnapshotAttributesResult": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "111122223333"
                ]
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的共用快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDbSnapshotAttribute。

modify-db-snapshot-attributes

以下程式碼範例顯示如何使用 modify-db-snapshot-attributes。

AWS CLI

修改資料庫快照屬性

下列modify-db-snapshot-attribute範例允許兩個 AWS 帳戶識別符 111122223333和 444455556666還原名為 的資料庫快照mydbsnapshot。

```
aws rds modify-db-snapshot-attribute \
    --db-snapshot-identifier mydbsnapshot \
    --attribute-name restore \
    --values-to-add '["111122223333","444455556666"]'
```

輸出：

```
{
    "DBSnapshotAttributesResult": {
```

```
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "111122223333",
                    "444455556666"
                ]
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的共用快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDbSnapshotAttributes。

`modify-db-snapshot`

以下程式碼範例顯示如何使用 `modify-db-snapshot`。

AWS CLI

修改資料庫快照

下列`modify-db-snapshot`範例會將名為 的 PostgeSQL 10.6 快照升級至 `db5-snapshot-upg-test` PostgreSQL 11.7。新的資料庫引擎版本會在快照完成升級且其狀態可用後顯示。

```
aws rds modify-db-snapshot \
    --db-snapshot-identifier db5-snapshot-upg-test \
    --engine-version 11.7
```

輸出：

```
{
    "DBSnapshot": {
        "DBSnapshotIdentifier": "db5-snapshot-upg-test",
        "DBInstanceIdentifier": "database-5",
        "SnapshotCreateTime": "2020-03-27T20:49:17.092Z",
        "Engine": "postgres",
        "AllocatedStorage": 20,
        "Status": "upgrading",
        "Port": 5432,
```

```
        "AvailabilityZone": "us-west-2a",
        "VpcId": "vpc-2ff27557",
        "InstanceCreateTime": "2020-03-27T19:59:04.735Z",
        "MasterUsername": "postgres",
        "EngineVersion": "10.6",
        "LicenseModel": "postgresql-license",
        "SnapshotType": "manual",
        "OptionGroupName": "default:postgres-11",
        "PercentProgress": 100,
        "StorageType": "gp2",
        "Encrypted": false,
        "DBSnapshotArn": "arn:aws:rds:us-west-2:123456789012:snapshot:db5-snapshot-
upg-test",
        "IAMDatabaseAuthenticationEnabled": false,
        "ProcessorFeatures": [],
        "DbiResourceId": "db-GJMF75LM42IL6BTFRE4UZJ5YM4"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的升級 PostgreSQL 資料庫快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDbSnapshot。

modify-db-subnet-group

以下程式碼範例顯示如何使用 modify-db-subnet-group。

AWS CLI

修改資料庫子網路群組

下列modify-db-subnet-group範例會將 ID 為 的子網路新增至名為 subnet-08e41f9e230222222的資料庫子網路群組mysubnetgroup。若要保留子網路群組中的現有子網路，請在 --subnet-ids選項中包含其 IDs做為值。請確定資料庫子網路群組中至少有兩個不同可用區域的子網路。

```
aws rds modify-db-subnet-group \
    --db-subnet-group-name mysubnetgroup \
    --subnet-ids
 '["subnet-0a1dc4e1a6f123456","subnet-070dd7ecb3aaaaaaa","subnet-00f5b198bc0abcdef","subnet-
```

輸出：

```
{
    "DBSubnetGroup": {
        "DBSubnetGroupName": "mysubnetgroup",
        "DBSubnetGroupDescription": "test DB subnet group",
        "VpcId": "vpc-0f08e7610a1b2c3d4",
        "SubnetGroupStatus": "Complete",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-08e41f9e230222222",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-070dd7ecb3aaaaaaa",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-00f5b198bc0abcdef",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2d"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-0a1dc4e1a6f123456",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                },
                "SubnetStatus": "Active"
            }
        ],
        "DBSubnetGroupArn": "arn:aws:rds:us-
west-2:534026745191:subgrp:mysubnetgroup"
    }
}
```


如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的步驟 3：建立資料庫子網路群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyDbSubnetGroup。

modify-event-subscription

以下程式碼範例顯示如何使用 modify-event-subscription。

AWS CLI

修改事件訂閱

下列modify-event-subscription範例會停用指定的事件訂閱，使其不會再將通知發佈至指定的 Amazon Simple Notification Service 主題。

```
aws rds modify-event-subscription \
    --subscription-name my-instance-events \
    --no-enabled
```

輸出：

```
{
    "EventSubscription": {
        "EventCategoriesList": [
            "backup",
            "recovery"
        ],
        "CustomerAwsId": "123456789012",
        "SourceType": "db-instance",
        "SubscriptionCreationTime": "Tue Jul 31 23:22:01 UTC 2018",
        "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-
events",
        "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",
        "CustSubscriptionId": "my-instance-events",
        "Status": "modifying",
        "Enabled": false
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyEventSubscription。

modify-global-cluster

以下程式碼範例顯示如何使用 modify-global-cluster。

AWS CLI

修改全域資料庫叢集

下列modify-global-cluster範例會啟用 Aurora MySQL 相容全域資料庫叢集的刪除保護。

```
aws rds modify-global-cluster \
    --global-cluster-identifier myglobalcluster \
    --deletion-protection
```

輸出：

```
{
    "GlobalCluster": {
        "GlobalClusterIdentifier": "myglobalcluster",
        "GlobalClusterResourceId": "cluster-f0e523bfe07aabb",
        "GlobalClusterArn": "arn:aws:rds::123456789012:global-
cluster:myglobalcluster",
        "Status": "available",
        "Engine": "aurora-mysql",
        "EngineVersion": "5.7.mysql_aurora.2.07.2",
        "StorageEncrypted": false,
        "DeletionProtection": true,
        "GlobalClusterMembers": []
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的管理 Aurora 全域資料庫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyGlobalCluster。

promote-read-replica-db-cluster

以下程式碼範例顯示如何使用 promote-read-replica-db-cluster。

AWS CLI

提升資料庫叢集僅供讀取複本

下列promote-read-replica-db-cluster範例會將指定的僅供讀取複本提升為獨立的資料庫叢集。

```
aws rds promote-read-replica-db-cluster \
```

```
    --db-cluster-identifier mydbcluster-1
```

輸出：

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1c"
        ],
        "BackupRetentionPeriod": 1,
        "DatabaseName": "",
        "DBClusterIdentifier": "mydbcluster-1",
        ...some output truncated...
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的將僅供讀取複本提升為資料庫叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PromoteReadReplicaDbCluster。

promote-read-replica

以下程式碼範例顯示如何使用 promote-read-replica。

AWS CLI

提升僅供讀取複本

下列promote-read-replica範例會將指定的僅供讀取複本提升為獨立的資料庫執行個體。

```
aws rds promote-read-replica \
    --db-instance-identifier test-instance-repl
```

輸出：

```
{
    "DBInstance": {
        "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance-repl",
        "StorageType": "standard",
```

```
        "ReadReplicaSourceDBInstanceIdentifier": "test-instance",
        "DBInstanceStatus": "modifying",
        ...some output truncated...
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PromoteReadReplica。

purchase-reserved-db-instance

以下程式碼範例顯示如何使用 purchase-reserved-db-instance。

AWS CLI

購買預留資料庫執行個體方案

下列purchase-reserved-db-instances-offering範例會購買預留資料庫執行個體方案。reserved-db-instances-offering-id 必須是有效的方案 ID，如 describe-reserved-db-instances-offering命令所傳回。

aws rds purchase-reserved-db-instances-offering --reserved-db-instances-offering-id 438012d3-4a52-4cc7-b2e3-8dff72e0e706

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PurchaseReservedDbInstance。

purchase-reserved-db-instances-offerings

以下程式碼範例顯示如何使用 purchase-reserved-db-instances-offerings。

AWS CLI

範例 1：尋找要購買的預留資料庫執行個體

下列describe-reserved-db-instances-offerings範例列出具有 db.t2.micro 執行個體類別且持續時間為一年的可用預留 MySQL 資料庫執行個體。購買預留資料庫執行個體時需要優惠 ID。

```
aws rds describe-reserved-db-instances-offerings \
    --product-description mysql \
    --db-instance-class db.t2.micro \
    --duration 1
```

輸出：

```
{
    "ReservedDBInstancesOfferings": [
        {
            "ReservedDBInstancesOfferingId": "8ba30be1-b9ec-447f-8f23-6114e3f4c7b4",
            "DBInstanceClass": "db.t2.micro",
            "Duration": 31536000,
            "FixedPrice": 51.0,
            "UsagePrice": 0.0,
            "CurrencyCode": "USD",
            "ProductDescription": "mysql",
            "OfferingType": "Partial Upfront",
            "MultiAZ": false,
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": 0.006,
                    "RecurringChargeFrequency": "Hourly"
                }
            ]
        },
    ... some output truncated ...
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的 Amazon RDS 的預留資料庫執行個體。

範例 2：購買預留資料庫執行個體

下列purchase-reserved-db-instances-offering範例顯示如何從上一個範例購買預留資料庫執行個體方案。

aws rds purchase-reserved-db-instances-offering --reserved-db-instances-offering-id 8ba30be1-b9ec-447f-8f23-6114e3f4c7b4

輸出：

```
{
    "ReservedDBInstance": {
        "ReservedDBInstanceId": "ri-2020-06-29-16-54-57-670",
        "ReservedDBInstancesOfferingId": "8ba30be1-b9ec-447f-8f23-6114e3f4c7b4",
        "DBInstanceClass": "db.t2.micro",
        "StartTime": "2020-06-29T16:54:57.670Z",
        "Duration": 31536000,
        "FixedPrice": 51.0,
```

```
        "UsagePrice": 0.0,
        "CurrencyCode": "USD",
        "DBInstanceCount": 1,
        "ProductDescription": "mysql",
        "OfferingType": "Partial Upfront",
        "MultiAZ": false,
        "State": "payment-pending",
        "RecurringCharges": [
            {
                "RecurringChargeAmount": 0.006,
                "RecurringChargeFrequency": "Hourly"
            }
        ],
        "ReservedDBInstanceArn": "arn:aws:rds:us-
west-2:123456789012:ri:ri-2020-06-29-16-54-57-670"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的 Amazon RDS 的預留資料庫執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PurchaseReservedDbInstancesOfferings。

reboot-db-instance

以下程式碼範例顯示如何使用 reboot-db-instance。

AWS CLI

重新啟動資料庫執行個體

下列reboot-db-instance範例會開始重新啟動指定的資料庫執行個體。

```
aws rds reboot-db-instance \
    --db-instance-identifier test-mysql-instance
```

輸出：

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-mysql-instance",
        "DBInstanceClass": "db.t3.micro",
        "Engine": "mysql",
```

```
        "DBInstanceStatus": "rebooting",
        "MasterUsername": "admin",
        "Endpoint": {
            "Address": "test-mysql-instance.############.us-
west-2.rds.amazonaws.com",
            "Port": 3306,
            "HostedZoneId": "Z1PVIF0EXAMPLE"
        },

    ... output omitted...

    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的重新啟動資料庫執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RebootDBInstance。

reboot-db-shard-group

以下程式碼範例顯示如何使用 reboot-db-shard-group。

AWS CLI

範例 1：重新啟動資料庫碎片群組

下列reboot-db-shard-group範例會重新啟動資料庫碎片群組。

```
aws rds reboot-db-shard-group \
    --db-shard-group-identifier my-db-shard-group
```

輸出：

```
{
    "DBShardGroups": [
        {
            "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
            "DBShardGroupIdentifier": "my-db-shard-group",
            "DBClusterIdentifier": "my-sv2-cluster",
            "MaxACU": 1000.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": false,
```

```
            "Endpoint": "my-sv2-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的重新啟動 Amazon Aurora 資料庫叢集或 Amazon Aurora 資料庫執行個體。

範例 2：描述資料庫碎片群組

下列describe-db-shard-groups範例會在您執行 reboot-db-shard-group命令後擷取資料庫碎片群組的詳細資訊。資料庫碎片群組現在my-db-shard-group正在重新啟動。

```
aws rds describe-db-shard-groups
```

輸出：

```
{
    "DBShardGroups": [
        {
            "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",
            "DBShardGroupIdentifier": "limitless-test-shard-grp",
            "DBClusterIdentifier": "limitless-test-cluster",
            "MaxACU": 768.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": true,
            "Endpoint": "limitless-test-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        },
        {
            "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
            "DBShardGroupIdentifier": "my-db-shard-group",
            "DBClusterIdentifier": "my-sv2-cluster",
            "MaxACU": 1000.0,
            "ComputeRedundancy": 0,
            "Status": "rebooting",
            "PubliclyAccessible": false,
            "Endpoint": "my-sv2-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的重新啟動 Amazon Aurora 資料庫叢集或 Amazon Aurora 資料庫執行個體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RebootDbShardGroup。

register-db-proxy-targets

以下程式碼範例顯示如何使用 register-db-proxy-targets。

AWS CLI

向資料庫註冊資料庫代理

下列register-db-proxy-targets範例會建立資料庫與代理之間的關聯。

```
aws rds register-db-proxy-targets \
    --db-proxy-name proxyExample \
    --db-cluster-identifiers database-5
```

輸出：

```
{
    "DBProxyTargets": [
        {
            "RdsResourceId": "database-5",
            "Port": 3306,
            "Type": "TRACKED_CLUSTER",
            "TargetHealth": {
                "State": "REGISTERING"
            }
        },
        {
            "Endpoint": "database-5instance-1.ab0cd1efghij.us-
east-1.rds.amazonaws.com",
            "RdsResourceId": "database-5",
            "Port": 3306,
            "Type": "RDS_INSTANCE",
            "TargetHealth": {
                "State": "REGISTERING"
            }
        }
```

```
    ]
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的建立 RDS 代理和《Amazon Aurora 使用者指南》中的建立 RDS 代理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterDbProxyTargets。

`remove-from-global-cluster`

以下程式碼範例顯示如何使用 `remove-from-global-cluster`。

AWS CLI

從 Aurora 全域資料庫叢集分離 Aurora 次要叢集

下列`remove-from-global-cluster`範例會將 Aurora 次要叢集與 Aurora 全域資料庫叢集分離。叢集會從唯讀變更為具有讀寫功能的獨立叢集。

```
aws rds remove-from-global-cluster \
    --region us-west-2 \
    --global-cluster-identifier myglobalcluster \
    --db-cluster-identifier arn:aws:rds:us-west-2:123456789012:cluster:DB-1
```

輸出：

```
{
    "GlobalCluster": {
        "GlobalClusterIdentifier": "myglobalcluster",
        "GlobalClusterResourceId": "cluster-abc123def456gh",
        "GlobalClusterArn": "arn:aws:rds::123456789012:global-
cluster:myglobalcluster",
        "Status": "available",
        "Engine": "aurora-postgresql",
        "EngineVersion": "10.11",
        "StorageEncrypted": true,
        "DeletionProtection": false,
        "GlobalClusterMembers": [
            {
                "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:js-
global-cluster",
                "Readers": [
```

```
                    "arn:aws:rds:us-west-2:123456789012:cluster:DB-1"
                ],
                "IsWriter": true
            },
            {
                "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:DB-1",
                "Readers": [],
                "IsWriter": false,
                "GlobalWriteForwardingStatus": "disabled"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的從 Amazon Aurora 全域資料庫移除叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveFromGlobalCluster。

remove-option-from-option-group

以下程式碼範例顯示如何使用 remove-option-from-option-group。

AWS CLI

從選項群組刪除選項

下列remove-option-from-option-group範例會從 移除 OEM選項myoptiongroup。

```
aws rds remove-option-from-option-group \
    --option-group-name myoptiongroup \
    --options OEM \
    --apply-immediately
```

輸出：

```
{
    "OptionGroup": {
        "OptionGroupName": "myoptiongroup",
        "OptionGroupDescription": "Test",
        "EngineName": "oracle-ee",
        "MajorEngineVersion": "19",
        "Options": [],
```

```
        "AllowsVpcAndNonVpcInstanceMemberships": true,
        "OptionGroupArn": "arn:aws:rds:us-east-1:123456789012:og:myoptiongroup"
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的從選項群組移除選項。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveOptionFromOptionGroup。

`remove-role-from-db-cluster`

以下程式碼範例顯示如何使用 `remove-role-from-db-cluster`。

AWS CLI

取消 AWS Identity and Access Management (IAM) 角色與資料庫叢集的關聯

下列`remove-role-from-db-cluster`範例會從資料庫叢集移除角色。

```
aws rds remove-role-from-db-cluster \
    --db-cluster-identifier mydbcluster \
    --role-arn arn:aws:iam::123456789012:role/RDSLoadFromS3
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的將 IAM 角色與 Amazon Aurora MySQL 資料庫叢集建立關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveRoleFromDbCluster。

`remove-role-from-db-instance`

以下程式碼範例顯示如何使用 `remove-role-from-db-instance`。

AWS CLI

取消 AWS Identity and Access Management (IAM) 角色與資料庫執行個體的關聯

下列`remove-role-from-db-instance`範例`rds-s3-integration-role`會從名為 的 Oracle 資料庫執行個體中移除名為 的角色`test-instance`。

```
aws rds remove-role-from-db-instance \
    --db-instance-identifier test-instance \
```

```
    --feature-name S3_INTEGRATION \
    --role-arn arn:aws:iam::111122223333:role/rds-s3-integration-role
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的停用 RDS SQL Server 與 S3 整合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveRoleFromDbInstance。

remove-source-identifier-from-subscription

以下程式碼範例顯示如何使用 remove-source-identifier-from-subscription。

AWS CLI

從訂閱中移除來源識別符

下列remove-source-identifier範例會從現有訂閱中移除指定的來源識別符。

```
aws rds remove-source-identifier-from-subscription \
    --subscription-name my-instance-events \
    --source-identifier test-instance-repl
```

輸出：

```
{
    "EventSubscription": {
        "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-
events",
        "SubscriptionCreationTime": "Tue Jul 31 23:22:01 UTC 2018",
        "EventCategoriesList": [
            "backup",
            "recovery"
        ],
        "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",
        "Status": "modifying",
        "CustSubscriptionId": "my-instance-events",
        "CustomerAwsId": "123456789012",
        "SourceIdsList": [
            "test-instance"
        ],
        "SourceType": "db-instance",
        "Enabled": false
```

```
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveSourceIdentifierFromSubscription。

remove-tags-from-resource

以下程式碼範例顯示如何使用 remove-tags-from-resource。

AWS CLI

從資源移除標籤

下列remove-tags-from-resource範例會從資源移除標籤。

```
aws rds remove-tags-from-resource \
    --resource-name arn:aws:rds:us-east-1:123456789012:db:mydbinstance \
    --tag-keys Name Environment
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的標記 Amazon RDS 資源和《Amazon Aurora 使用者指南》中的標記 Amazon RDS 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 RemoveTagsFromResource。

reset-db-cluster-parameter-group

以下程式碼範例顯示如何使用 reset-db-cluster-parameter-group。

AWS CLI

範例 1：將所有參數重設為其預設值

下列reset-db-cluster-parameter-group範例會將客戶建立的資料庫叢集參數群組中的所有參數值重設為其預設值。

```
aws rds reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclpg \
    --reset-all-parameters
```

輸出：

```
{
    "DBClusterParameterGroupName": "mydbclpg"
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的使用資料庫參數群組和資料庫叢集參數群組。

範例 2：將特定參數重設為其預設值

下列reset-db-cluster-parameter-group範例會將特定參數的參數值重設為客戶建立的資料庫叢集參數群組中的預設值。

```
aws rds reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclpgy \
    --parameters "ParameterName=max_connections,ApplyMethod=immediate" \
                 "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

輸出：

```
{
    "DBClusterParameterGroupName": "mydbclpg"
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的使用資料庫參數群組和資料庫叢集參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetDbClusterParameterGroup。

reset-db-parameter-group

以下程式碼範例顯示如何使用 reset-db-parameter-group。

AWS CLI

範例 1：將所有參數重設為其預設值

下列reset-db-parameter-group範例會將客戶建立的資料庫參數群組中的所有參數值重設為其預設值。

```
aws rds reset-db-parameter-group \
    --db-parameter-group-name mypg \
    --reset-all-parameters
```

輸出：

```
{
    "DBParameterGroupName": "mypg"
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的使用資料庫參數群組和《Amazon Aurora 使用者指南》中的使用資料庫參數群組和資料庫叢集參數群組。

範例 2：將特定參數重設為其預設值

下列reset-db-parameter-group範例會將特定參數的參數值重設為客戶建立的資料庫參數群組中的預設值。

```
aws rds reset-db-parameter-group \
    --db-parameter-group-name mypg \
    --parameters "ParameterName=max_connections,ApplyMethod=immediate" \
                 "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

輸出：

```
{
    "DBParameterGroupName": "mypg"
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的使用資料庫參數群組和《Amazon Aurora 使用者指南》中的使用資料庫參數群組和資料庫叢集參數群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetDbParameterGroup。

restore-db-cluster-from-s3

以下程式碼範例顯示如何使用 restore-db-cluster-from-s3。

AWS CLI

從 Amazon S3 還原 Amazon Aurora 資料庫叢集

下列restore-db-cluster-from-s3範例會從 Amazon S3 中的 MySQL 5.7 資料庫備份檔案還原與 Amazon Aurora MySQL 5.7 版相容的資料庫叢集。 Amazon S3

```
aws rds restore-db-cluster-from-s3 \
    --db-cluster-identifier cluster-s3-restore \
    --engine aurora-mysql \
    --master-username admin \
    --master-user-password mypassword \
    --s3-bucket-name amzn-s3-demo-bucket \
    --s3-prefix test-backup \
    --s3-ingestion-role-arn arn:aws:iam::123456789012:role/service-role/TestBackup \
    --source-engine mysql \
    --source-engine-version 5.7.28
```

輸出：

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-west-2c",
            "us-west-2a",
            "us-west-2b"
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "cluster-s3-restore",
        "DBClusterParameterGroup": "default.aurora-mysql5.7",
        "DBSubnetGroup": "default",
        "Status": "creating",
        "Endpoint": "cluster-s3-restore.cluster-co3xyzabc123.us-
west-2.rds.amazonaws.com",
        "ReaderEndpoint": "cluster-s3-restore.cluster-ro-co3xyzabc123.us-
west-2.rds.amazonaws.com",
        "MultiAZ": false,
        "Engine": "aurora-mysql",
        "EngineVersion": "5.7.12",
        "Port": 3306,
        "MasterUsername": "admin",
        "PreferredBackupWindow": "11:15-11:45",
        "PreferredMaintenanceWindow": "thu:12:19-thu:12:49",
        "ReadReplicaIdentifiers": [],
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
```

```
            {
                "VpcSecurityGroupId": "sg-########",
                "Status": "active"
            }
        ],
        "HostedZoneId": "Z1PVIF0EXAMPLE",
        "StorageEncrypted": false,
        "DbClusterResourceId": "cluster-SU5THYQQHOWCXZZDGXREXAMPLE",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:cluster-s3-
restore",
        "AssociatedRoles": [],
        "IAMDatabaseAuthenticationEnabled": false,
        "ClusterCreateTime": "2020-07-27T14:22:08.095Z",
        "EngineMode": "provisioned",
        "DeletionProtection": false,
        "HttpEndpointEnabled": false,
        "CopyTagsToSnapshot": false,
        "CrossAccountClone": false,
        "DomainMemberships": []
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的使用 Amazon S3 儲存貯體從 MySQL 遷移資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreDbClusterFromS3。

`restore-db-cluster-from-snapshot`

以下程式碼範例顯示如何使用 `restore-db-cluster-from-snapshot`。

AWS CLI

從快照還原資料庫叢集

以下內容會從名為 的資料庫叢集快照`restore-db-cluster-from-snapshot`還原與 PostgreSQL 10.7 版相容的 Aurora PostgreSQL 資料庫叢集。 PostgreSQL `test-instance-snapshot`

```
aws rds restore-db-cluster-from-snapshot \
    --db-cluster-identifier newdbcluster \
    --snapshot-identifier test-instance-snapshot \
    --engine aurora-postgresql \
```

```
    --engine-version 10.7
```

輸出：

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-west-2c",
            "us-west-2a",
            "us-west-2b"
        ],
        "BackupRetentionPeriod": 7,
        "DatabaseName": "",
        "DBClusterIdentifier": "newdbcluster",
        "DBClusterParameterGroup": "default.aurora-postgresql10",
        "DBSubnetGroup": "default",
        "Status": "creating",
        "Endpoint": "newdbcluster.cluster-############.us-west-2.rds.amazonaws.com",
        "ReaderEndpoint": "newdbcluster.cluster-ro-############.us-
west-2.rds.amazonaws.com",
        "MultiAZ": false,
        "Engine": "aurora-postgresql",
        "EngineVersion": "10.7",
        "Port": 5432,
        "MasterUsername": "postgres",
        "PreferredBackupWindow": "09:33-10:03",
        "PreferredMaintenanceWindow": "sun:12:22-sun:12:52",
        "ReadReplicaIdentifiers": [],
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-########",
                "Status": "active"
            }
        ],
        "HostedZoneId": "Z1PVIF0EXAMPLE",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/287364e4-33e3-4755-a3b0-
a1b2c3d4e5f6",
        "DbClusterResourceId": "cluster-5DSB5IFQDDUVAWOUWM1EXAMPLE",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:newdbcluster",
        "AssociatedRoles": [],
```

```
        "IAMDatabaseAuthenticationEnabled": false,
        "ClusterCreateTime": "2020-06-05T15:06:58.634Z",
        "EngineMode": "provisioned",
        "DeletionProtection": false,
        "HttpEndpointEnabled": false,
        "CopyTagsToSnapshot": false,
        "CrossAccountClone": false,
        "DomainMemberships": []
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的從資料庫叢集快照還原。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreDbClusterFromSnapshot。

`restore-db-cluster-to-point-in-time`

以下程式碼範例顯示如何使用 `restore-db-cluster-to-point-in-time`。

AWS CLI

將資料庫叢集還原至指定的時間

下列`restore-db-cluster-to-point-in-time`範例會將名為 的資料庫叢集還原`database-4`至最新的可能時間。使用 `copy-on-write`作為還原類型會將新的資料庫叢集還原為來源資料庫叢集的複製。

```
aws rds restore-db-cluster-to-point-in-time \
    --source-db-cluster-identifier database-4 \
    --db-cluster-identifier sample-cluster-clone \
    --restore-type copy-on-write \
    --use-latest-restorable-time
```

輸出：

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-west-2c",
            "us-west-2a",
            "us-west-2b"
        ],
```

```
        "BackupRetentionPeriod": 7,
        "DatabaseName": "",
        "DBClusterIdentifier": "sample-cluster-clone",
        "DBClusterParameterGroup": "default.aurora-postgresql10",
        "DBSubnetGroup": "default",
        "Status": "creating",
        "Endpoint": "sample-cluster-clone.cluster-############.us-
west-2.rds.amazonaws.com",
        "ReaderEndpoint": "sample-cluster-clone.cluster-ro-############.us-
west-2.rds.amazonaws.com",
        "MultiAZ": false,
        "Engine": "aurora-postgresql",
        "EngineVersion": "10.7",
        "Port": 5432,
        "MasterUsername": "postgres",
        "PreferredBackupWindow": "09:33-10:03",
        "PreferredMaintenanceWindow": "sun:12:22-sun:12:52",
        "ReadReplicaIdentifiers": [],
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-########",
                "Status": "active"
            }
        ],
        "HostedZoneId": "Z1PVIF0EXAMPLE",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/287364e4-33e3-4755-a3b0-
a1b2c3d4e5f6",
        "DbClusterResourceId": "cluster-BIZ77GDSA2XBSTNPFW1EXAMPLE",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster-
clone",
        "AssociatedRoles": [],
        "IAMDatabaseAuthenticationEnabled": false,
        "CloneGroupId": "8d19331a-099a-45a4-b4aa-11aa22bb33cc44dd",
        "ClusterCreateTime": "2020-03-10T19:57:38.967Z",
        "EngineMode": "provisioned",
        "DeletionProtection": false,
        "HttpEndpointEnabled": false,
        "CopyTagsToSnapshot": false,
        "CrossAccountClone": false
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的將資料庫叢集還原至指定時間。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreDbClusterToPointInTime。

`restore-db-instance-from-db-snapshot`

以下程式碼範例顯示如何使用 `restore-db-instance-from-db-snapshot`。

AWS CLI

從資料庫快照還原資料庫執行個體

下列`restore-db-instance-from-db-snapshot`範例會從指定的資料庫快照，`db7-new-instance`使用`db.t3.small`資料庫執行個體類別建立名為 的新資料庫執行個體。從中擷取快照的來源資料庫執行個體會使用已取代的資料庫執行個體類別，因此您無法將其升級。

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier db7-new-instance \
    --db-snapshot-identifier db7-test-snapshot \
    --db-instance-class db.t3.small
```

輸出：

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "db7-new-instance",
        "DBInstanceClass": "db.t3.small",
        "Engine": "mysql",
        "DBInstanceStatus": "creating",

        ...output omitted...

        "PreferredMaintenanceWindow": "mon:07:37-mon:08:07",
        "PendingModifiedValues": {},
        "MultiAZ": false,
        "EngineVersion": "5.7.22",
        "AutoMinorVersionUpgrade": true,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "general-public-license",

        ...output omitted...

        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:db7-new-instance",
```

```
        "IAMDatabaseAuthenticationEnabled": false,
        "PerformanceInsightsEnabled": false,
        "DeletionProtection": false,
        "AssociatedRoles": []
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的從資料庫快照還原。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreDbInstanceFromDbSnapshot。

`restore-db-instance-from-s3`

以下程式碼範例顯示如何使用 `restore-db-instance-from-s3`。

AWS CLI

從 Amazon S3 中的備份還原資料庫執行個體

下列`restore-db-instance-from-s3`範例`restored-test-instance`會從 `my-backups` S3 儲存貯體中的現有備份建立名為 的新資料庫執行個體。

```
aws rds restore-db-instance-from-s3 \
    --db-instance-identifier restored-test-instance \
    --allocated-storage 250 --db-instance-class db.m4.large --engine mysql \
    --master-username master --master-user-password secret99 \
    --s3-bucket-name my-backups --s3-ingestion-role-
arn arn:aws:iam::123456789012:role/my-role \
    --source-engine mysql --source-engine-version 5.6.27
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreDbInstanceFromS3。

`restore-db-instance-to-point-in-time`

以下程式碼範例顯示如何使用 `restore-db-instance-to-point-in-time`。

AWS CLI

範例 1：將資料庫執行個體還原至某個時間點

下列`restore-db-instance-to-point-in-time`範例會在`restored-test-instance`指定時間還原`test-instance`至名為 的新資料庫執行個體。

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-identifier test-instance \
    --target-db-instance restored-test-instance \
    --restore-time 2018-07-30T23:45:00.000Z
```

輸出：

```
{
    "DBInstance": {
        "AllocatedStorage": 20,
        "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:restored-test-
instance",
        "DBInstanceStatus": "creating",
        "DBInstanceIdentifier": "restored-test-instance",
        ...some output omitted...
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的將資料庫執行個體還原至指定的時間。

範例 2：從複寫備份將資料庫執行個體還原至指定的時間

下列restore-db-instance-to-point-in-time範例會從複寫的自動備份將 Oracle 資料庫執行個體還原至指定的時間。

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-automated-backups-arn "arn:aws:rds:us-
west-2:123456789012:auto-backup:ab-jkib2gfq5rv7replzadausbrktni2bn4example" \
    --target-db-instance-identifier myorclinstance-from-replicated-backup \
    --restore-time 2020-12-08T18:45:00.000Z
```

輸出：

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "myorclinstance-from-replicated-backup",
        "DBInstanceClass": "db.t3.micro",
        "Engine": "oracle-se2",
        "DBInstanceStatus": "creating",
        "MasterUsername": "admin",
        "DBName": "ORCL",
```

```
        "AllocatedStorage": 20,
        "PreferredBackupWindow": "07:45-08:15",
        "BackupRetentionPeriod": 14,
        ... some output omitted ...
        "DbiResourceId": "db-KGLXG75BGVIWKQT7NQ4EXAMPLE",
        "CACertificateIdentifier": "rds-ca-2019",
        "DomainMemberships": [],
        "CopyTagsToSnapshot": false,
        "MonitoringInterval": 0,
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:myorclinstance-from-
replicated-backup",
        "IAMDatabaseAuthenticationEnabled": false,
        "PerformanceInsightsEnabled": false,
        "DeletionProtection": false,
        "AssociatedRoles": [],
        "TagList": []
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的從複寫備份還原至指定的時間。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreDbInstanceToPointInTime。

`start-activity-stream`

以下程式碼範例顯示如何使用 `start-activity-stream`。

AWS CLI

啟動資料庫活動串流

下列`start-activity-stream`範例會啟動非同步活動串流，以監控名為 my-pg-cluster 的 Aurora 叢集。

```
aws rds start-activity-stream \
    --region us-east-1 \
    --mode async \
    --kms-key-id arn:aws:kms:us-east-1:1234567890123:key/a12c345d-6ef7-890g-
h123-456i789jk0l1 \
    --resource-arn arn:aws:rds:us-east-1:1234567890123:cluster:my-pg-cluster \
    --apply-immediately
```

輸出：

```
{
    "KmsKeyId": "arn:aws:kms:us-east-1:1234567890123:key/a12c345d-6ef7-890g-
h123-456i789jk0l1",
    "KinesisStreamName": "aws-rds-das-cluster-0ABCDEFGHI1JKLM2NOPQ3R4S",
    "Status": "starting",
    "Mode": "async",
    "ApplyImmediately": true
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的啟動資料庫活動串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartActivityStream。

start-db-cluster

以下程式碼範例顯示如何使用 start-db-cluster。

AWS CLI

啟動資料庫叢集

下列start-db-cluster範例會啟動資料庫叢集及其資料庫執行個體。

```
aws rds start-db-cluster \
    --db-cluster-identifier mydbcluster
```

輸出：

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1e",
            "us-east-1b"
        ],
        "BackupRetentionPeriod": 1,
        "DatabaseName": "mydb",
        "DBClusterIdentifier": "mydbcluster",
        ...some output truncated...
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的停止和啟動 Amazon Aurora 資料庫叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartDbCluster。

start-db-instance-automated-backups-replication

以下程式碼範例顯示如何使用 start-db-instance-automated-backups-replication。

AWS CLI

啟用跨區域自動備份

下列start-db-instance-automated-backups-replication範例會將自動備份從美國東部（維吉尼亞北部）區域的資料庫執行個體複寫至美國西部（奧勒岡）。備份保留期為 14 天。

```
aws rds start-db-instance-automated-backups-replication \
    --region us-west-2 \
    --source-db-instance-arn "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db" \
    --backup-retention-period 14
```

輸出：

```
{
    "DBInstanceAutomatedBackup": {
        "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db",
        "DbiResourceId": "db-JKIB2GFQ5RV7REPLZA4EXAMPLE",
        "Region": "us-east-1",
        "DBInstanceIdentifier": "new-orcl-db",
        "RestoreWindow": {},
        "AllocatedStorage": 20,
        "Status": "pending",
        "Port": 1521,
        "InstanceCreateTime": "2020-12-04T15:28:31Z",
        "MasterUsername": "admin",
        "Engine": "oracle-se2",
        "EngineVersion": "12.1.0.2.v21",
        "LicenseModel": "bring-your-own-license",
        "OptionGroupName": "default:oracle-se2-12-1",
        "Encrypted": false,
```

```
        "StorageType": "gp2",
        "IAMDatabaseAuthenticationEnabled": false,
        "BackupRetentionPeriod": 14,
        "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-
backup:ab-jkib2gfq5rv7replzadausbrktni2bn4example"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的啟用跨區域自動備份。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartDbInstanceAutomatedBackupsReplication。

start-db-instance

以下程式碼範例顯示如何使用 start-db-instance。

AWS CLI

啟動資料庫執行個體

下列start-db-instance範例會啟動指定的資料庫執行個體。

```
aws rds start-db-instance \
    --db-instance-identifier test-instance
```

輸出：

```
{
    "DBInstance": {
        "DBInstanceStatus": "starting",
        ...some output truncated...
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartDbInstance。

start-export-task

以下程式碼範例顯示如何使用 start-export-task。

AWS CLI

將快照匯出至 Amazon S3

下列start-export-task範例會將名為 的資料庫快照匯出db5-snapshot-test至名為 的 Amazon S3 儲存貯體amzn-s3-demo-bucket。

```
aws rds start-export-task \
    --export-task-identifier my-s3-export \
    --source-arn arn:aws:rds:us-west-2:123456789012:snapshot:db5-snapshot-test \
    --s3-bucket-name amzn-s3-demo-bucket \
    --iam-role-arn arn:aws:iam::123456789012:role/service-role/ExportRole \
    --kms-key-id arn:aws:kms:us-west-2:123456789012:key/abcd0000-7fca-4128-82f2-aabbccddeeff
```

輸出：

```
{
    "ExportTaskIdentifier": "my-s3-export",
    "SourceArn": "arn:aws:rds:us-west-2:123456789012:snapshot:db5-snapshot-test",
    "SnapshotTime": "2020-03-27T20:48:42.023Z",
    "S3Bucket": "amzn-s3-demo-bucket",
    "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/ExportRole",
    "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/abcd0000-7fca-4128-82f2-aabbccddeeff",
    "Status": "STARTING",
    "PercentProgress": 0,
    "TotalExtractedDataInGB": 0
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的將快照匯出至 Amazon S3 儲存貯體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartExportTask。

stop-activity-stream

以下程式碼範例顯示如何使用 stop-activity-stream。

AWS CLI

停止資料庫活動串流

下列stop-activity-stream範例會停止名為 my-pg-cluster 之 Aurora 叢集中的活動串流。

```
aws rds stop-activity-stream \
    --region us-east-1 \
    --resource-arn arn:aws:rds:us-east-1:1234567890123:cluster:my-pg-cluster \
    --apply-immediately
```

輸出：

```
{
    "KmsKeyId": "arn:aws:kms:us-east-1:1234567890123:key/a12c345d-6ef7-890g-
h123-456i789jk0l1",
    "KinesisStreamName": "aws-rds-das-cluster-0ABCDEFGHI1JKLM2NOPQ3R4S",
    "Status": "stopping"
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的停止活動串流。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopActivityStream。

stop-db-cluster

以下程式碼範例顯示如何使用 stop-db-cluster。

AWS CLI

停止資料庫叢集

下列stop-db-cluster範例會停止資料庫叢集及其資料庫執行個體。

```
aws rds stop-db-cluster \
    --db-cluster-identifier mydbcluster
```

輸出：

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-east-1a",
```

```
            "us-east-1e",
            "us-east-1b"
        ],
        "BackupRetentionPeriod": 1,
        "DatabaseName": "mydb",
        "DBClusterIdentifier": "mydbcluster",
        ...some output truncated...
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的停止和啟動 Amazon Aurora 資料庫叢集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopDbCluster。

stop-db-instance-automated-backups-replication

以下程式碼範例顯示如何使用 stop-db-instance-automated-backups-replication。

AWS CLI

停止複寫自動備份

以下內容會stop-db-instance-automated-backups-replication結束自動備份複寫至美國西部（奧勒岡）區域。複寫的備份會根據設定的備份保留期間進行保留。

```
aws rds stop-db-instance-automated-backups-replication \
    --region us-west-2 \
    --source-db-instance-arn "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db"
```

輸出：

```
{
    "DBInstanceAutomatedBackup": {
        "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db",
        "DbiResourceId": "db-JKIB2GFQ5RV7REPLZA4EXAMPLE",
        "Region": "us-east-1",
        "DBInstanceIdentifier": "new-orcl-db",
        "RestoreWindow": {
            "EarliestTime": "2020-12-04T23:13:21.030Z",
            "LatestTime": "2020-12-07T19:59:57Z"
        },
```

```
        "AllocatedStorage": 20,
        "Status": "replicating",
        "Port": 1521,
        "InstanceCreateTime": "2020-12-04T15:28:31Z",
        "MasterUsername": "admin",
        "Engine": "oracle-se2",
        "EngineVersion": "12.1.0.2.v21",
        "LicenseModel": "bring-your-own-license",
        "OptionGroupName": "default:oracle-se2-12-1",
        "Encrypted": false,
        "StorageType": "gp2",
        "IAMDatabaseAuthenticationEnabled": false,
        "BackupRetentionPeriod": 7,
        "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-
backup:ab-jkib2gfq5rv7replzadausbrktni2bn4example"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的停止自動備份複寫。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopDbInstanceAutomatedBackupsReplication。

stop-db-instance

以下程式碼範例顯示如何使用 stop-db-instance。

AWS CLI

停止資料庫執行個體

下列stop-db-instance範例會停止指定的資料庫執行個體。

```
aws rds stop-db-instance \
    --db-instance-identifier test-instance
```

輸出：

```
{
    "DBInstance": {
        "DBInstanceStatus": "stopping",
        ...some output truncated...
```

```
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopDbInstance。

switchover-blue-green-deployment

以下程式碼範例顯示如何使用 switchover-blue-green-deployment。

AWS CLI

範例 1：切換 RDS 資料庫執行個體的藍/綠部署

下列switchover-blue-green-deployment範例會將指定的綠色環境提升為新的生產環境。

```
aws rds switchover-blue-green-deployment \
    --blue-green-deployment-identifier bgd-wi89nwzglccsfake \
    --switchover-timeout 300
```

輸出：

```
{
    "BlueGreenDeployment": {
        "BlueGreenDeploymentIdentifier": "bgd-v53303651eexfake",
        "BlueGreenDeploymentName": "bgd-cli-test-instance",
        "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
        "Target": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-green-
blhi1e",
        "SwitchoverDetails": [
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-green-blhi1e",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1-green-k5fv7u",
                "Status": "AVAILABLE"
```

```
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2-green-ggsh8m",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3-green-o2vwm0",
                "Status": "AVAILABLE"
            }
        ],
        "Tasks": [
            {
                "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                "Status": "COMPLETED"
            },
            {
                "Name": "DB_ENGINE_VERSION_UPGRADE",
                "Status": "COMPLETED"
            },
            {
                "Name": "CONFIGURE_BACKUPS",
                "Status": "COMPLETED"
            },
            {
                "Name": "CREATING_TOPOLOGY_OF_SOURCE",
                "Status": "COMPLETED"
            }
        ],
        "Status": "SWITCHOVER_IN_PROGRESS",
        "CreateTime": "2022-02-25T22:33:22.225000+00:00"
    }
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的切換藍/綠部署。

範例 2：提升 Aurora MySQL 資料庫叢集的藍/綠部署

下列`switchover-blue-green-deployment`範例會將指定的綠色環境提升為新的生產環境。

```
aws rds switchover-blue-green-deployment \
    --blue-green-deployment-identifier bgd-wi89nwzglccsfake \
    --switchover-timeout 300
```

輸出：

```
{
    "BlueGreenDeployment": {
        "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
        "BlueGreenDeploymentName": "my-blue-green-deployment",
        "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
        "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster-green-3ud8z6",
        "SwitchoverDetails": [
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster-green-3ud8z6",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-1",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-1-green-bvxc73",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-2",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-2-green-7wc4ie",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-3",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-3-green-p4xxkz",
                "Status": "AVAILABLE"
            },
```

```
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-green-np1ikl",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-green-miszlf",
                "Status": "AVAILABLE"
            }
        ],
        "Tasks": [
            {
                "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                "Status": "COMPLETED"
            },
            {
                "Name": "DB_ENGINE_VERSION_UPGRADE",
                "Status": "COMPLETED"
            },
            {
                "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
                "Status": "COMPLETED"
            },
            {
                "Name": "CREATE_CUSTOM_ENDPOINTS",
                "Status": "COMPLETED"
            }
        ],
        "Status": "SWITCHOVER_IN_PROGRESS",
        "CreateTime": "2022-02-25T22:38:49.522000+00:00"
    }
}
```

如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的切換藍/綠部署。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SwitchoverBlueGreenDeployment。

使用 的 Amazon RDS Data Service 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon RDS Data Service 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`batch-execute-statement`

以下程式碼範例顯示如何使用 `batch-execute-statement`。

AWS CLI

執行批次 SQL 陳述式

下列`batch-execute-statement`範例會在具有參數集的資料陣列上執行批次 SQL 陳述式。

```
aws rds-data batch-execute-statement \
    --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \
    --database "mydb" \
    --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \
    --sql "insert into mytable values (:id, :val)" \
    --parameter-sets "[[{\"name\": \"id\", \"value\": {\"longValue\": 1}},{\"name\":
 \"val\", \"value\": {\"stringValue\": \"ValueOne\"}}],
       [{\"name\": \"id\", \"value\": {\"longValue\": 2}},{\"name\": \"val\",
 \"value\": {\"stringValue\": \"ValueTwo\"}}],
       [{\"name\": \"id\", \"value\": {\"longValue\": 3}},{\"name\": \"val\",
 \"value\": {\"stringValue\": \"ValueThree\"}}]]"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的使用 Aurora Serverless 的資料 API。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchExecuteStatement。

begin-transaction

以下程式碼範例顯示如何使用 begin-transaction。

AWS CLI

啟動 SQL 交易

下列begin-transaction範例會啟動 SQL 交易。

```
aws rds-data begin-transaction \
    --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \
    --database "mydb" \
    --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret"
```

輸出：

```
{
    "transactionId": "ABC1234567890xyz"
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的使用 Aurora Serverless 的資料 API。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BeginTransaction。

commit-transaction

以下程式碼範例顯示如何使用 commit-transaction。

AWS CLI

遞交 SQL 交易

下列commit-transaction範例會結束指定的 SQL 交易，並遞交您所做的變更。

```
aws rds-data commit-transaction \
    --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \
    --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \
```

```
    --transaction-id "ABC1234567890xyz"
```

輸出：

```
{
    "transactionStatus": "Transaction Committed"
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的使用 Aurora Serverless 的資料 API。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的CommitTransaction。

execute-statement

以下程式碼範例顯示如何使用 execute-statement。

AWS CLI

範例 1：執行屬於交易一部分的 SQL 陳述式

下列execute-statement範例會執行屬於交易一部分的 SQL 陳述式。

```
aws rds-data execute-statement \
    --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \
    --database "mydb" \
    --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \
    --sql "update mytable set quantity=5 where id=201" \
    --transaction-id "ABC1234567890xyz"
```

輸出：

```
{
    "numberOfRecordsUpdated": 1
}
```

範例 2：使用參數執行 SQL 陳述式

下列execute-statement範例會執行具有參數的 SQL 陳述式。

```
aws rds-data execute-statement \
```

```
    --resource-arn "arn:aws:rds:us-east-1:123456789012:cluster:mydbcluster" \
    --database "mydb" \
    --secret-arn "arn:aws:secretsmanager:us-east-1:123456789012:secret:mysecret" \
    --sql "insert into mytable values (:id, :val)" \
    --parameters "[{\"name\": \"id\", \"value\": {\"longValue\": 1}},{\"name\":
 \"val\", \"value\": {\"stringValue\": \"value1\"}}]"
```

輸出：

```
{
    "numberOfRecordsUpdated": 1
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的使用 Aurora Serverless 的資料 API。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ExecuteStatement。

rollback-transaction

以下程式碼範例顯示如何使用 rollback-transaction。

AWS CLI

復原 SQL 交易

下列rollback-transaction範例會復原指定的 SQL 交易。

```
aws rds-data rollback-transaction \
    --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \
    --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \
    --transaction-id "ABC1234567890xyz"
```

輸出：

```
{
    "transactionStatus": "Rollback Complete"
}
```

如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的使用 Aurora Serverless 的資料 API。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RollbackTransaction。

使用 的 Amazon RDS Performance Insights 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon RDS Performance Insights 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-performance-analysis-report`

以下程式碼範例顯示如何使用 `create-performance-analysis-report`。

AWS CLI

建立效能分析報告

下列`create-performance-analysis-report`範例會建立效能分析報告，其中包含資料庫`1682979503`的開始時間`1682969503`和結束時間`db-abcdefg123456789`。

```
aws pi create-performance-analysis-report \
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --start-time 1682969503 \
    --end-time 1682979503
```

輸出：

```
{
    "AnalysisReportId": "report-0234d3ed98e28fb17"
}
```

如需建立效能分析報告的詳細資訊，請參閱《Amazon RDS 使用者指南》中的在績效詳情中建立效能分析報告，以及《Amazon Aurora 使用者指南》中的在績效詳情中建立效能分析報告。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePerformanceAnalysisReport。

delete-performance-analysis-report

以下程式碼範例顯示如何使用 delete-performance-analysis-report。

AWS CLI

刪除效能分析報告

下列delete-performance-analysis-report範例會刪除報告 ID 為 的效能分析報告report-0d99cc91c4422ee61。

```
aws pi delete-performance-analysis-report \
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --analysis-report-id report-0d99cc91c4422ee61
```

此命令不會產生輸出。

如需刪除效能分析報告的詳細資訊，請參閱《Amazon RDS 使用者指南》中的刪除績效詳情中的效能分析報告，以及《Amazon Aurora 使用者指南》中的刪除績效詳情中的效能分析報告。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePerformanceAnalysisReport。

describe-dimension-keys

以下程式碼範例顯示如何使用 describe-dimension-keys。

AWS CLI

範例 1：描述維度索引鍵

此範例會請求所有等待事件的名稱。資料會依事件名稱，以及這些事件在指定期間內的彙總值進行摘要。

命令：

```
aws pi describe-dimension-keys --service-type RDS --identifier db-
LKCGOBK26374TPTDFXOIWVCPPM --start-time 1527026400 --end-time 1527080400 --
metric db.load.avg --group-by '{"Group":"db.wait_event"}'
```

輸出：


```
{
    "AlignedEndTime": 1.5270804E9,
    "AlignedStartTime": 1.5270264E9,
    "Keys": [
        {
            "Dimensions": {"db.wait_event.name": "wait/synch/mutex/innodb/
aurora_lock_thread_slot_futex"},
            "Total": 0.05906906851195666
        },
        {
            "Dimensions": {"db.wait_event.name": "wait/io/aurora_redo_log_flush"},
            "Total": 0.015824722186149193
        },
        {
            "Dimensions": {"db.wait_event.name": "CPU"},
            "Total": 0.008014396230265477
        },
        {
            "Dimensions": {"db.wait_event.name": "wait/io/
aurora_respond_to_client"},
            "Total": 0.0036361612526204477
        },
        {
            "Dimensions": {"db.wait_event.name": "wait/io/table/sql/handler"},
            "Total": 0.0019108398419382965
        },
        {
            "Dimensions": {"db.wait_event.name": "wait/synch/cond/mysys/
my_thread_var::suspend"},
            "Total": 8.533847837782684E-4
        },
        {
            "Dimensions": {"db.wait_event.name": "wait/io/file/csv/data"},
            "Total": 6.864181956477376E-4
        },
        {
            "Dimensions": {"db.wait_event.name": "Unknown"},
            "Total": 3.895887056379051E-4
        },
        {
            "Dimensions": {"db.wait_event.name": "wait/synch/mutex/sql/
FILE_AS_TABLE::LOCK_shim_lists"},
```

```
            "Total": 3.710368625122906E-5
        },
        {
            "Dimensions": {"db.wait_event.name": "wait/lock/table/sql/handler"},
            "Total": 0
        }
    ]
}
```

範例 2：尋找對資料庫負載貢獻最大之陳述式的 SQL ID

以下describe-dimension-keys請求 10 個陳述式的 SQL 陳述式和 SQL ID，這些陳述式對資料庫負載的貢獻最大。

```
aws pi describe-dimension-keys \
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --start-time 2023-05-01T00:00:00Z \
    --end-time 2023-05-01T01:00:00Z \
    --metric db.load.avg \
    --group-by '{"Group": "db.sql", "Dimensions": ["db.sql.id",
 "db.sql.statement"],"Limit": 10}'
```

輸出：

```
{
    "AlignedEndTime": 1.5270804E9,
    "AlignedStartTime": 1.5270264E9,
    "Identifier": "db-abcdefg123456789",
    "MetricList": [
        {
            "Keys": [
                {
                    "Dimensions": {"db.sql.id": "AKIAIOSFODNN7EXAMPLE",
 "db.sql.statement": "SELECT * FROM customers WHERE customer_id = 123"},
                    "Total": 25.5,"Partitions": [12.3, 13.2]
                }
            ]
        }
    ]
}
```

如需績效詳情中維度的詳細資訊，請參閱《Amazon RDS 使用者指南》中的資料庫負載和《Amazon Aurora 使用者指南》中的資料庫負載。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDimensionKeys。

get-dimension-key-details

以下程式碼範例顯示如何使用 get-dimension-key-details。

AWS CLI

取得資料庫執行個體之指定維度群組的詳細資訊

下列get-dimension-key-details範例會擷取資料庫執行個體 的 SQL 陳述式全文db-10BCD2EFGHIJ3KL4M5NO6PQRS5。--group 即為 db.sql，而 --group-identifier 即為 db.sql.id。在此範例中， example-sql-id代表使用 get-resource-metrics或 describe-dimension-keys操作擷取的 SQL ID。在此範例中，維度詳細資訊可供使用。因此，績效詳情會擷取 SQL 陳述式的完整文字，而不會將其截斷。

```
aws pi get-dimension-key-details \
    --service-type RDS \
    --identifier db-10BCD2EFGHIJ3KL4M5NO6PQRS5 \
    --group db.sql \
    --group-identifier example-sql-id \
    --requested-dimensions statement
```

輸出：

```
{
    "Dimensions":[
        {
            "Value": "SELECT e.last_name, d.department_name FROM employees e,
 departments d WHERE e.department_id=d.department_id",
            "Dimension": "db.sql.statement",
            "Status": "AVAILABLE"
        },
    ...
    ]
}
```

如需績效詳情中維度的詳細資訊，請參閱《Amazon RDS 使用者指南》中的資料庫負載和《Amazon Aurora 使用者指南》中的資料庫負載。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDimensionKeyDetails。

get-performance-analysis-report

以下程式碼範例顯示如何使用 get-performance-analysis-report。

AWS CLI

取得效能分析報告

下列get-performance-analysis-report範例會取得db-abcdefg123456789報告 ID 為 之資料庫的效能分析報告report-0d99cc91c4422ee61。回應提供報告狀態、ID、時間詳細資訊和洞見。

```
aws pi get-performance-analysis-report \
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --analysis-report-id report-0d99cc91c4422ee61
```

輸出：

```
{
    "AnalysisReport": {
        "Status": "Succeeded",
        "ServiceType": "RDS",
        "Identifier": "db-abcdefg123456789",
        "StartTime": 1680583486.584,
        "AnalysisReportId": "report-0d99cc91c4422ee61",
        "EndTime": 1680587086.584,
        "CreateTime": 1680587087.139,
        "Insights": [
            ... (Condensed for space)
       ]
    }
}
```

如需效能分析報告的詳細資訊，請參閱《Amazon RDS 使用者指南》中的分析一段時間的資料庫效能，以及《Amazon Aurora 使用者指南》中的分析一段時間的資料庫效能。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPerformanceAnalysisReport。

get-resource-metadata

以下程式碼範例顯示如何使用 get-resource-metadata。

AWS CLI

取得資料庫的資源中繼資料

下列get-resource-metadata範例會取得資料庫 的資源中繼資料db-abcdefg123456789。回應顯示已啟用 SQL 摘要統計資料。

```
aws pi get-resource-metadata \
    --service-type RDS \
    --identifier db-abcdefg123456789
```

輸出：

```
{
    "Identifier": "db-abcdefg123456789",
    "Features":{
        "SQL_DIGEST_STATISTICS":{
            "Status": "ENABLED"
        }
    }
}
```

如需績效詳情 SQL 統計資料的詳細資訊，請參閱《Amazon RDS 使用者指南》中的績效詳情的 SQL 統計資料和《Amazon Aurora 使用者指南》中的績效詳情的 SQL 統計資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResourceMetadata。

get-resource-metrics

以下程式碼範例顯示如何使用 get-resource-metrics。

AWS CLI

取得資源指標

此範例會請求 db.wait_event 維度群組的資料點，以及該群組內 db.wait_event.name 維度的資料點。在回應中，相關資料點會依請求的維度 (db.wait_event.name) 分組。

命令：

```
aws pi get-resource-metrics --service-type RDS --identifier db-
LKCGOBK26374TPTDFXOIWVCPPM --start-time 1527026400 --end-time 1527080400 --period-
in-seconds 300 --metric db.load.avg --metric-queries file://metric-queries.json
```

的引數--metric-queries存放在 JSON 檔案 中metric-queries.json。以下是該檔案的內容：

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": {
            "Group":"db.wait_event"
        }
    }
]
```

輸出：

```
{
    "AlignedEndTime": 1.5270804E9,
    "AlignedStartTime": 1.5270264E9,
    "Identifier": "db-LKCGOBK26374TPTDFXOIWVCPPM",
    "MetricList": [
        {
            "Key": {
                "Metric": "db.load.avg"
            },
            "DataPoints": [
                {
                    "Timestamp": 1527026700.0,
                    "Value": 1.3533333333333333
                },
                {
                    "Timestamp": 1527027000.0,
                    "Value": 0.88
                },
                <...remaining output omitted...>
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
```

```
                "Dimensions": {
                    "db.wait_event.name": "wait/synch/mutex/innodb/
aurora_lock_thread_slot_futex"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1527026700.0,
                    "Value": 0.8566666666666667
                },
                {
                    "Timestamp": 1527027000.0,
                    "Value": 0.8633333333333333
                },
                <...remaining output omitted...>
            ],
        },
            <...remaining output omitted...>
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResourceMetrics。

list-available-resource-dimensions

以下程式碼範例顯示如何使用 list-available-resource-dimensions。

AWS CLI

列出可在資料庫執行個體上查詢指標類型的維度

下列list-available-resource-dimensions範例列出您可以查詢資料庫 的db.load指標db-abcdefg123456789。

```
aws pi list-available-resource-dimensions \
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --metrics db.load
```

輸出：

```
{
```

```
    "MetricDimensions": [
        {
            "Metric": "db.load",
            "Groups": [
                {
                    "Group": "db.user",
                    "Dimensions": [
                        {
                            "Identifier": "db.user.id"
                        },
                        {
                            "Identifier": "db.user.name"
                        }
                    ]
                },
                {
                    "Group": "db.sql_tokenized",
                    "Dimensions": [
                        {
                            "Identifier": "db.sql_tokenized.id"
                        },
                        {
                            "Identifier": "db.sql_tokenized.db_id"
                        },
                        {
                            "Identifier": "db.sql_tokenized.statement"
                        }
                    ]
                },
                ...
            ]
        }
    ]
}
```

如需績效詳情中維度的詳細資訊，請參閱《Amazon RDS 使用者指南》中的資料庫負載和《Amazon Aurora 使用者指南》中的資料庫負載。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAvailableResourceDimensions。

`list-available-resource-metrics`

以下程式碼範例顯示如何使用 `list-available-resource-metrics`。

AWS CLI

列出可在資料庫執行個體上查詢指標類型的指標

下列list-available-resource-metrics範例列出您可以查詢資料庫 的db.load指標db-abcdefg123456789。

```
aws pi list-available-resource-metrics \
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --metric-types "os" "db"
```

輸出：

```
{
    "Metrics": [
        {
            "Description": "The number of virtual CPUs for the DB instance",
            "Metric": "os.general.numVCPUs",
            "Unit": "vCPUs"
        },
        ......,
        {
            "Description": "Time spent reading data file blocks by backends in this
 instance",
            "Metric": "db.IO.read_latency",
            "Unit": "Milliseconds per block"
        },
        ......
    ]
}
```

如需績效詳情中指標的詳細資訊，請參閱《Amazon RDS 使用者指南》中的資料庫負載和《Amazon Aurora 使用者指南》中的資料庫負載。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAvailableResourceMetrics。

list-performance-analysis-reports

以下程式碼範例顯示如何使用 list-performance-analysis-reports。

AWS CLI

列出資料庫的效能分析報告

下列list-performance-analysis-reports範例列出資料庫 的效能分析報告db-abcdefg123456789。回應將列出所有報告，其中包含報告 ID，狀態和時間區間的詳細資訊。

```
aws pi list-performance-analysis-reports \
    --service-type RDS \
    --identifier db-abcdefg123456789
```

輸出：

```
{
    "AnalysisReports": [
        {
            "Status": "Succeeded",
            "EndTime": 1680587086.584,
            "CreateTime": 1680587087.139,
            "StartTime": 1680583486.584,
            "AnalysisReportId": "report-0d99cc91c4422ee61"
        },
        {
            "Status": "Succeeded",
            "EndTime": 1681491137.914,
            "CreateTime": 1681491145.973,
            "StartTime": 1681487537.914,
            "AnalysisReportId": "report-002633115cc002233"
        },
        {
            "Status": "Succeeded",
            "EndTime": 1681493499.849,
            "CreateTime": 1681493507.762,
            "StartTime": 1681489899.849,
            "AnalysisReportId": "report-043b1e006b47246f9"
        },
        {
            "Status": "InProgress",
            "EndTime": 1682979503.0,
            "CreateTime": 1682979618.994,
            "StartTime": 1682969503.0,
            "AnalysisReportId": "report-01ad15f9b88bcbd56"
        }
```

```
    ]
}
```

如需效能分析報告的詳細資訊，請參閱《Amazon RDS 使用者指南》中的分析一段時間的資料庫效能，以及《Amazon Aurora 使用者指南》中的分析一段時間的資料庫效能。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPerformanceAnalysisReports。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出效能分析報告的標籤

下列`list-tags-for-resource`範例會列出具有報告 ID 之效能分析報告的標籤`report-0d99cc91c4422ee61`。

```
aws pi list-tags-for-resource \
    --service-type RDS \
    --resource-arn arn:aws:pi:us-west-2:123456789012:perf-reports/RDS/db-abcdefg123456789/report-0d99cc91c4422ee61
```

輸出：

```
{
    "Tags": [
        {
            "Value": "test-tag",
            "Key": "name"
        }
    ]
}
```

如需有關標記效能分析報告的詳細資訊，請參閱《Amazon RDS 使用者指南》中的將標籤新增至效能分析報告，以及《Amazon Aurora 使用者指南》中的將標籤新增至效能分析報告。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤新增至效能分析報告

下列tag-resource範例會將標籤值name為 的標籤索引鍵新增至報告 ID test-tag 為 的效能分析報告report-0d99cc91c4422ee61。

```
aws pi tag-resource \
    --service-type RDS \
    --resource-arn arn:aws:pi:us-west-2:123456789012:perf-reports/RDS/db-abcdefg123456789/report-0d99cc91c4422ee61 \
    --tags Key=name,Value=test-tag
```

此命令不會產生輸出。

如需有關標記效能分析報告的詳細資訊，請參閱《Amazon RDS 使用者指南》中的將標籤新增至效能分析報告，以及《Amazon Aurora 使用者指南》中的將標籤新增至效能分析報告。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

刪除效能分析報告的標籤

下列untag-resource範例會刪除具有報告 ID 之name效能分析報告的標籤report-0d99cc91c4422ee61。

```
aws pi untag-resource \
    --service-type RDS \
    --resource-arn arn:aws:pi:us-west-2:123456789012:perf-reports/RDS/db-abcdefg123456789/report-0d99cc91c4422ee61 \
    --tag-keys name
```

此命令不會產生輸出。

如需有關標記效能分析報告的詳細資訊，請參閱《Amazon RDS 使用者指南》中的將標籤新增至效能分析報告，以及《Amazon Aurora 使用者指南》中的將標籤新增至效能分析報告。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

使用 的 Amazon Redshift 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon Redshift 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`accept-reserved-node-exchange`

以下程式碼範例顯示如何使用 `accept-reserved-node-exchange`。

AWS CLI

接受預留節點交換

下列`accept-reserved-node-exchange`範例接受 DC1 預留節點的 DC2 預留節點交換。

```
aws redshift accept-reserved-node-exchange /
    --reserved-node-id 12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE /
    --target-reserved-node-offering-id 12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE
```

輸出：

```
{
    "ExchangedReservedNode": {
        "ReservedNodeId": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",
        "ReservedNodeOfferingId": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",
        "NodeType": "dc2.large",
        "StartTime": "2019-12-06T21:17:26Z",
```

```
        "Duration": 31536000,
        "FixedPrice": 0.0,
        "UsagePrice": 0.0,
        "CurrencyCode": "USD",
        "NodeCount": 1,
        "State": "exchanging",
        "OfferingType": "All Upfront",
        "RecurringCharges": [
            {
                "RecurringChargeAmount": 0.0,
                "RecurringChargeFrequency": "Hourly"
            }
        ],
        "ReservedNodeOfferingType": "Regular"
    }
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的使用 AWS CLI 升級預留節點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptReservedNodeExchange。

authorize-cluster-security-group-ingress

以下程式碼範例顯示如何使用 authorize-cluster-security-group-ingress。

AWS CLI

授權存取 EC2 安全GroupThis範例授權存取名為 的 Amazon EC2 安全群組。命令：

```
aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name
 mysecuritygroup --ec2-security-group-name myec2securitygroup --ec2-security-group-
owner-id 123445677890
```

授權存取 CIDR rangeThis範例授權存取 CIDR 範圍。命令：

```
aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name
 mysecuritygroup --cidrip 192.168.100.100/32
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AuthorizeClusterSecurityGroupIngress。

authorize-snapshot-access

以下程式碼範例顯示如何使用 authorize-snapshot-access。

AWS CLI

授權 AWS 帳戶還原SnapshotThis範例授權 AWS 帳戶444455556666還原快照 my-snapshot-id。根據預設，輸出為 JSON 格式。命令：

```
aws redshift authorize-snapshot-access --snapshot-id my-snapshot-id --account-with-restore-access 444455556666
```

結果：

```
{
   "Snapshot": {
      "Status": "available",
      "SnapshotCreateTime": "2013-07-17T22:04:18.947Z",
      "EstimatedSecondsToCompletion": 0,
      "AvailabilityZone": "us-east-1a",
      "ClusterVersion": "1.0",
      "MasterUsername": "adminuser",
      "Encrypted": false,
      "OwnerAccount": "111122223333",
      "BackupProgressInMegabytes": 11.0,
      "ElapsedTimeInSeconds": 0,
      "DBName": "dev",
      "CurrentBackupRateInMegabytesPerSecond: 0.1534,
      "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
      "ActualIncrementalBackupSizeInMegabytes"; 11.0,
      "SnapshotType": "manual",
      "NodeType": "dw.hs1.xlarge",
      "ClusterIdentifier": "mycluster",
      "TotalBackupSizeInMegabytes": 20.0,
      "Port": 5439,
      "NumberOfNodes": 2,
      "SnapshotIdentifier": "my-snapshot-id"
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AuthorizeSnapshotAccess。

batch-delete-cluster-snapshots

以下程式碼範例顯示如何使用 batch-delete-cluster-snapshots。

AWS CLI

刪除一組叢集快照

下列batch-delete-cluster-snapshots範例會刪除一組手動叢集快照。

```
aws redshift batch-delete-cluster-snapshots \
        --
identifiers SnapshotIdentifier=mycluster-2019-11-06-14-12 SnapshotIdentifier=mycluster-2019-
```

輸出：

```
{
    "Resources": [
        "mycluster-2019-11-06-14-12",
        "mycluster-2019-11-06-14-20"
    ]
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的 Amazon Redshift 快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDeleteClusterSnapshots。

batch-modify-cluster-snapshots

以下程式碼範例顯示如何使用 batch-modify-cluster-snapshots。

AWS CLI

修改一組叢集快照

下列batch-modify-cluster-snapshots範例會修改一組叢集快照的設定。

```
aws redshift batch-modify-cluster-snapshots \
    --snapshot-identifier-list mycluster-2019-11-06-16-31 mycluster-2019-11-06-16-32
 \
    --manual-snapshot-retention-period 30
```

輸出：

```
{
    "Resources": [
        "mycluster-2019-11-06-16-31",
        "mycluster-2019-11-06-16-32"
    ],
    "Errors": [],
    "ResponseMetadata": {
        "RequestId": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",
        "HTTPStatusCode": 200,
        "HTTPHeaders": {
                "x-amzn-requestid": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE,
                "content-type": "text/xml",
                "content-length": "480",
                "date": "Sat, 07 Dec 2019 00:36:09 GMT",
                "connection": "keep-alive"
        },
        "RetryAttempts": 0
    }
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的 Amazon Redshift 快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchModifyClusterSnapshots。

cancel-resize

以下程式碼範例顯示如何使用 cancel-resize。

AWS CLI

取消叢集的大小調整

下列cancel-resize範例會取消叢集的傳統調整大小操作。

```
aws redshift cancel-resize \
    --cluster-identifier mycluster
```

輸出：

```
{
    "TargetNodeType": "dc2.large",
    "TargetNumberOfNodes": 2,
```

```
    "TargetClusterType": "multi-node",
    "Status": "CANCELLING",
    "ResizeType": "ClassicResize",
    "TargetEncryptionType": "NONE"
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的調整 Amazon Redshift 中的叢集大小。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelResize。

copy-cluster-snapshot

以下程式碼範例顯示如何使用 copy-cluster-snapshot。

AWS CLI

取得所有叢集VersionsThis範例會傳回所有叢集版本的描述。根據預設，輸出為 JSON 格式。命令：

```
aws redshift copy-cluster-snapshot --source-snapshot-identifier
 cm:examplecluster-2013-01-22-19-27-58 --target-snapshot-identifier my-saved-
snapshot-copy
```

結果：

```
{
   "Snapshot": {
      "Status": "available",
      "SnapshotCreateTime": "2013-01-22T19:27:58.931Z",
      "AvailabilityZone": "us-east-1c",
      "ClusterVersion": "1.0",
      "MasterUsername": "adminuser",
      "DBName": "dev",
      "ClusterCreateTime": "2013-01-22T19:23:59.368Z",
      "SnapshotType": "manual",
      "NodeType": "dw.hs1.xlarge",
      "ClusterIdentifier": "examplecluster",
      "Port": 5439,
      "NumberOfNodes": "2",
      "SnapshotIdentifier": "my-saved-snapshot-copy"
   },
```

```
   "ResponseMetadata": {
      "RequestId": "3b279691-64e3-11e2-bec0-17624ad140dd"
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CopyClusterSnapshot。

create-cluster-parameter-group

以下程式碼範例顯示如何使用 create-cluster-parameter-group。

AWS CLI

建立叢集參數GroupThis範例會建立新的叢集參數群組。命令：

```
aws redshift create-cluster-parameter-group --parameter-group-name
 myclusterparametergroup --parameter-group-family redshift-1.0 --description "My
 first cluster parameter group"
```

結果：

```
{
   "ClusterParameterGroup": {
      "ParameterGroupFamily": "redshift-1.0",
      "Description": "My first cluster parameter group",
      "ParameterGroupName": "myclusterparametergroup"
   },
   "ResponseMetadata": {
      "RequestId": "739448f0-64cc-11e2-8f7d-3b939af52818"
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateClusterParameterGroup。

create-cluster-security-group

以下程式碼範例顯示如何使用 create-cluster-security-group。

AWS CLI

建立叢集安全GroupThis範例會建立新的叢集安全群組。根據預設，輸出為 JSON 格式。命令：

```
aws redshift create-cluster-security-group --cluster-security-group-name
 mysecuritygroup --description "This is my cluster security group"
```

結果：

```
{
   "create_cluster_security_group_response": {
      "create_cluster_security_group_result": {
         "cluster_security_group": {
            "description": "This is my cluster security group",
            "owner_id": "300454760768",
            "cluster_security_group_name": "mysecuritygroup",
            "ec2_security_groups": \[],
            "ip_ranges": \[]
         }
      },
      "response_metadata": {
         "request_id": "5df486a0-343a-11e2-b0d8-d15d0ef48549"
      }
   }
}
```

您也可以使用 `--output text`選項，以文字格式取得相同的資訊。命令：

`--output text` option.Command：

option.Command：

```
aws redshift create-cluster-security-group --cluster-security-group-name
 mysecuritygroup --description "This is my cluster security group" --output text
```

結果：

```
This is my cluster security group   300454760768    mysecuritygroup
a0c0bfab-343a-11e2-95d2-c3dc9fe8ab57
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateClusterSecurityGroup。

`create-cluster-snapshot`

以下程式碼範例顯示如何使用 `create-cluster-snapshot`。

AWS CLI

建立叢集SnapshotThis範例會建立新的叢集快照。根據預設，輸出為 JSON 格式。命令：

```
aws redshift create-cluster-snapshot --cluster-identifier mycluster --snapshot-
identifier my-snapshot-id
```

結果：

```
{
   "Snapshot": {
      "Status": "creating",
      "SnapshotCreateTime": "2013-01-22T22:20:33.548Z",
      "AvailabilityZone": "us-east-1a",
      "ClusterVersion": "1.0",
      "MasterUsername": "adminuser",
      "DBName": "dev",
      "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
      "SnapshotType": "manual",
      "NodeType": "dw.hs1.xlarge",
      "ClusterIdentifier": "mycluster",
      "Port": 5439,
      "NumberOfNodes": "2",
      "SnapshotIdentifier": "my-snapshot-id"
   },
   "ResponseMetadata": {
      "RequestId": "f024d1a5-64e1-11e2-88c5-53eb05787dfb"
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateClusterSnapshot。

`create-cluster-subnet-group`

以下程式碼範例顯示如何使用 `create-cluster-subnet-group`。

AWS CLI

建立叢集子網路GroupThis範例會建立新的叢集子網路群組。命令：

```
aws redshift create-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
 --description "My subnet group" --subnet-ids subnet-763fdd1c
```

結果：

```
{
   "ClusterSubnetGroup": {
      "Subnets": [
         {
            "SubnetStatus": "Active",
            "SubnetIdentifier": "subnet-763fdd1c",
            "SubnetAvailabilityZone": {
               "Name": "us-east-1a"
            }
         } ],
      "VpcId": "vpc-7e3fdd14",
      "SubnetGroupStatus": "Complete",
      "Description": "My subnet group",
      "ClusterSubnetGroupName": "mysubnetgroup"
   },
   "ResponseMetadata": {
      "RequestId": "500b8ce2-698f-11e2-9790-fd67517fb6fd"
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateClusterSubnetGroup。

create-cluster

以下程式碼範例顯示如何使用 create-cluster。

AWS CLI

建立具有最小ParametersThis範例會建立具有最小參數集的叢集。根據預設，輸出為 JSON 格式。
命令：

```
aws redshift create-cluster --node-type dw.hs1.xlarge --number-of-nodes 2 --master-
username adminuser --master-user-password TopSecret1 --cluster-identifier mycluster
```

結果：

```
{
   "Cluster": {
      "NodeType": "dw.hs1.xlarge",
```

```
      "ClusterVersion": "1.0",
      "PubliclyAccessible": "true",
      "MasterUsername": "adminuser",
      "ClusterParameterGroups": [
         {
            "ParameterApplyStatus": "in-sync",
            "ParameterGroupName": "default.redshift-1.0"
         } ],
      "ClusterSecurityGroups": [
         {
            "Status": "active",
            "ClusterSecurityGroupName": "default"
         } ],
      "AllowVersionUpgrade": true,
      "VpcSecurityGroups": \[],
      "PreferredMaintenanceWindow": "sat:03:30-sat:04:00",
      "AutomatedSnapshotRetentionPeriod": 1,
      "ClusterStatus": "creating",
      "ClusterIdentifier": "mycluster",
      "DBName": "dev",
      "NumberOfNodes": 2,
      "PendingModifiedValues": {
         "MasterUserPassword": "\****"
      }
   },
   "ResponseMetadata": {
      "RequestId": "7cf4bcfc-64dd-11e2-bea9-49e0ce183f07"
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCluster。

create-event-subscription

以下程式碼範例顯示如何使用 create-event-subscription。

AWS CLI

建立事件的通知訂閱

下列create-event-subscription範例會建立事件通知訂閱。

```
aws redshift create-event-subscription \
```

```
    --subscription-name mysubscription \
    --sns-topic-arn arn:aws:sns:us-west-2:123456789012:MySNStopic \
    --source-type cluster \
    --source-ids mycluster
```

輸出：

```
{
        "EventSubscription": {
        "CustomerAwsId": "123456789012",
        "CustSubscriptionId": "mysubscription",
        "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:MySNStopic",
        "Status": "active",
        "SubscriptionCreationTime": "2019-12-09T20:05:19.365Z",
        "SourceType": "cluster",
        "SourceIdsList": [
            "mycluster"
        ],
        "EventCategoriesList": [],
        "Severity": "INFO",
        "Enabled": true,
        "Tags": []
    }
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的訂閱 Amazon Redshift 事件通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateEventSubscription。

create-hsm-client-certificate

以下程式碼範例顯示如何使用 create-hsm-client-certificate。

AWS CLI

建立 HSM 用戶端憑證

下列create-hsm-client-certificate範例會產生 HSM 用戶端憑證，讓叢集可用來連線至 HSM。

```
aws redshift create-hsm-client-certificate \
    --hsm-client-certificate-identifier myhsmclientcert
```

輸出：

```
{
    "HsmClientCertificate": {
        "HsmClientCertificateIdentifier": "myhsmclientcert",
        "HsmClientCertificatePublicKey": "-----BEGIN CERTIFICATE-----
        MIICiEXAMPLECQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
        VVMxCzAJBgNVBAgTEXAMPLEwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
        b24xFDASBgNVBAsTC0lBTSBDb25EXAMPLEIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
        BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb2EXAMPLETEwNDI1MjA0NTIxWhcN
        MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBEXAMPLEMRAwDgYD
        EXAMPLETZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
        b2xlMRIwEAEXAMPLEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
        YXpvbi5jb20wgZ8wDQYJKEXAMPLEAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
        21uUSfwfEvySWtC2XADZ4nB+BLYgVIk6EXAMPLE3G93vUEIO3IyNoH/f0wYK8m9T
        rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugEXAMPLEzZswY6786m86gpE
        Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEEXAMPLEEAtCu4
        nUhVVxYUEXAMPLEh8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
        FFBjvSfpJIlJ00zbhNYS5f6GEXAMPLEl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
        NYiytVbZPQUQ5Yaxu2jXnimvw3rEXAMPLE=-----END CERTIFICATE-----\n",
    "Tags": []
    }
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的 Amazon Redshift API 許可參考。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateHsmClientCertificate。

create-hsm-configuration

以下程式碼範例顯示如何使用 create-hsm-configuration。

AWS CLI

建立 HSM 組態

下列create-hsm-configuration範例會建立指定的 HSM 組態，其中包含叢集在硬體安全模組 (HSM) 中存放和使用資料庫加密金鑰所需的資訊。

```
aws redshift create-hsm-configuration /
    --hsm-configuration-identifier myhsmconnection
    --description "My HSM connection"
    --hsm-ip-address 192.0.2.09
```

```
    --hsm-partition-name myhsmpartition /
    --hsm-partition-password A1b2c3d4 /
    --hsm-server-public-certificate myhsmclientcert
```

輸出：

```
{
    "HsmConfiguration": {
        "HsmConfigurationIdentifier": "myhsmconnection",
        "Description": "My HSM connection",
        "HsmIpAddress": "192.0.2.09",
        "HsmPartitionName": "myhsmpartition",
        "Tags": []
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateHsmConfiguration。

create-snapshot-copy-grant

以下程式碼範例顯示如何使用 create-snapshot-copy-grant。

AWS CLI

建立快照複製授予

下列create-snapshot-copy-grant範例會建立快照複製授予，並加密目的地區域中複製的快照 AWS 。

```
aws redshift create-snapshot-copy-grant \
    --snapshot-copy-grant-name mysnapshotcopygrantname
```

輸出：

```
{
    "SnapshotCopyGrant": {
        "SnapshotCopyGrantName": "mysnapshotcopygrantname",
        "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
bPxRfih3yCo8nvbEXAMPLEKEY",
        "Tags": []
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的 Amazon Redshift 資料庫加密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSnapshotCopyGrant。

create-snapshot-schedule

以下程式碼範例顯示如何使用 create-snapshot-schedule。

AWS CLI

建立快照排程

下列create-snapshot-schedule範例會使用指定的描述和每 12 小時的速率來建立快照排程。

```
aws redshift create-snapshot-schedule \
    --schedule-definitions "rate(12 hours)" \
    --schedule-identifier mysnapshotschedule \
    --schedule-description "My schedule description"
```

輸出：

```
{
    "ScheduleDefinitions": [
        "rate(12 hours)"
    ],
    "ScheduleIdentifier": "mysnapshotschedule",
    "ScheduleDescription": "My schedule description",
    "Tags": []
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的自動快照排程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSnapshotSchedule。

create-tags

以下程式碼範例顯示如何使用 create-tags。

AWS CLI

建立叢集的標籤

下列create-tags範例會將指定的標籤鍵/值對新增至指定的叢集。

```
aws redshift create-tags \
    --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \
    --tags "Key"="mytags","Value"="tag1"
```

此命令不會產生任何輸出。

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的在 Amazon Redshift 中標記資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateTags。

delete-cluster-parameter-group

以下程式碼範例顯示如何使用 delete-cluster-parameter-group。

AWS CLI

刪除叢集參數GroupThis範例會刪除叢集參數群組。命令：

```
aws redshift delete-cluster-parameter-group --parameter-group-name
 myclusterparametergroup
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteClusterParameterGroup。

delete-cluster-security-group

以下程式碼範例顯示如何使用 delete-cluster-security-group。

AWS CLI

刪除叢集安全GroupThis範例會刪除叢集安全群組。命令：

```
aws redshift delete-cluster-security-group --cluster-security-group-name
 mysecuritygroup
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteClusterSecurityGroup。

delete-cluster-snapshot

以下程式碼範例顯示如何使用 delete-cluster-snapshot。

AWS CLI

刪除叢集SnapshotThis範例會刪除叢集快照。命令：

```
aws redshift delete-cluster-snapshot --snapshot-identifier my-snapshot-id
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteClusterSnapshot。

`delete-cluster-subnet-group`

以下程式碼範例顯示如何使用 `delete-cluster-subnet-group`。

AWS CLI

刪除叢集子網路GroupThis範例會刪除叢集子網路群組。命令：

```
aws redshift delete-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
```

結果：

```
{
   "ResponseMetadata": {
      "RequestId": "253fbffd-6993-11e2-bc3a-47431073908a"
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteClusterSubnetGroup。

`delete-cluster`

以下程式碼範例顯示如何使用 `delete-cluster`。

AWS CLI

刪除沒有最終叢集SnapshotThis範例會刪除叢集，強制刪除資料，因此不會建立最終叢集快照。命令：

```
aws redshift delete-cluster --cluster-identifier mycluster --skip-final-cluster-
snapshot
```

刪除叢集，允許最終叢集SnapshotThis範例會刪除叢集，但會指定最終叢集快照。命令：

```
aws redshift delete-cluster --cluster-identifier mycluster --final-cluster-snapshot-
identifier myfinalsnapshot
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCluster。

delete-event-subscription

以下程式碼範例顯示如何使用 delete-event-subscription。

AWS CLI

刪除事件訂閱

下列delete-event-subscription範例會刪除指定的事件通知訂閱。

```
aws redshift delete-event-subscription \
    --subscription-name mysubscription
```

此命令不會產生任何輸出。

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的訂閱 Amazon Redshift 事件通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteEventSubscription。

delete-hsm-client-certificate

以下程式碼範例顯示如何使用 delete-hsm-client-certificate。

AWS CLI

刪除 HSM 用戶端憑證

下列delete-hsm-client-certificate範例會刪除 HSM 用戶端憑證。

```
aws redshift delete-hsm-client-certificate \
    --hsm-client-certificate-identifier myhsmclientcert
```

此命令不會產生任何輸出。

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的 Amazon Redshift API 許可參考。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteHsmClientCertificate。

delete-hsm-configuration

以下程式碼範例顯示如何使用 delete-hsm-configuration。

AWS CLI

刪除 HSM 組態

下列delete-hsm-configuration範例會從目前 AWS 帳戶刪除指定的 HSM 組態。

```
aws redshift delete-hsm-configuration /
    --hsm-configuration-identifier myhsmconnection
```

此命令不會產生任何輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteHsmConfiguration。

delete-scheduled-action

以下程式碼範例顯示如何使用 delete-scheduled-action。

AWS CLI

刪除排程動作

下列delete-scheduled-action範例會刪除指定的排程動作。

```
aws redshift delete-scheduled-action \
    --scheduled-action-name myscheduledaction
```

此命令不會產生任何輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteScheduledAction。

delete-snapshot-copy-grant

以下程式碼範例顯示如何使用 delete-snapshot-copy-grant。

AWS CLI

刪除快照複製授予

下列delete-snapshot-copy-grant範例會刪除指定的快照複製授予。

```
aws redshift delete-snapshot-copy-grant \
    --snapshot-copy-grant-name mysnapshotcopygrantname
```

此命令不會產生任何輸出。

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的 Amazon Redshift 資料庫加密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSnapshotCopyGrant。

delete-snapshot-schedule

以下程式碼範例顯示如何使用 delete-snapshot-schedule。

AWS CLI

刪除快照排程

下列delete-snapshot-schedule範例會刪除指定的快照排程。您必須先取消叢集的關聯，才能刪除排程。

```
aws redshift delete-snapshot-schedule \
    --schedule-identifier mysnapshotschedule
```

此命令不會產生任何輸出。

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的自動快照排程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSnapshotSchedule。

delete-tags

以下程式碼範例顯示如何使用 delete-tags。

AWS CLI

從叢集刪除標籤

下列delete-tags範例會從指定的叢集刪除具有指定金鑰名稱的標籤。

```
aws redshift delete-tags \
    --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \
    --tag-keys "clustertagkey" "clustertagvalue"
```

此命令不會產生任何輸出。

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的在 Amazon Redshift 中標記資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTags。

describe-account-attributes

以下程式碼範例顯示如何使用 describe-account-attributes。

AWS CLI

描述 AWS 帳戶的屬性

下列describe-account-attributes範例顯示連接至呼叫 AWS 帳戶的屬性。

```
aws redshift describe-account-attributes
```

輸出：

```
{
    "AccountAttributes": [
        {
            "AttributeName": "max-defer-maintenance-duration",
            "AttributeValues": [
                {
                    "AttributeValue": "45"
                }
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAccountAttributes。

describe-cluster-db-revisions

以下程式碼範例顯示如何使用 describe-cluster-db-revisions。

AWS CLI

描述叢集的資料庫修訂

下列describe-cluster-db-revisions範例顯示指定叢集之ClusterDbRevision物件陣列的詳細資訊。

```
aws redshift describe-cluster-db-revisions \
    --cluster-identifier mycluster
```

輸出：

```
{
    "ClusterDbRevisions": [
        {
            "ClusterIdentifier": "mycluster",
            "CurrentDatabaseRevision": "11420",
            "DatabaseRevisionReleaseDate": "2019-11-22T16:43:49.597Z",
            "RevisionTargets": []
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClusterDbRevisions。

describe-cluster-parameter-groups

以下程式碼範例顯示如何使用 describe-cluster-parameter-groups。

AWS CLI

取得所有叢集參數GroupsThis範例會傳回帳戶所有叢集參數群組的描述，其中包含資料欄標頭。根據預設，輸出為 JSON 格式。命令：

```
aws redshift describe-cluster-parameter-groups
```

結果：

```
{
   "ParameterGroups": [
      {
         "ParameterGroupFamily": "redshift-1.0",
         "Description": "My first cluster parameter group",
         "ParameterGroupName": "myclusterparametergroup"
      } ],
```

```
   "ResponseMetadata": {
      "RequestId": "8ceb8f6f-64cc-11e2-bea9-49e0ce183f07"
   }
}
```

您也可以使用 `--output text`選項，以文字格式取得相同的資訊。命令：

`--output text` option.Command：

option.Command：

```
aws redshift describe-cluster-parameter-groups --output text
```

結果：

```
redshift-1.0        My first cluster parameter group        myclusterparametergroup
RESPONSEMETADATA    9e665a36-64cc-11e2-8f7d-3b939af52818
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClusterParameterGroups。

`describe-cluster-parameters`

以下程式碼範例顯示如何使用 `describe-cluster-parameters`。

AWS CLI

擷取指定叢集參數GroupThis範例會擷取具名參數群組的參數。根據預設，輸出為 JSON 格式。命令：

```
aws redshift describe-cluster-parameters --parameter-group-name
 myclusterparametergroup
```

結果：

```
{
   "Parameters": [
      {
         "Description": "Sets the display format for date and time values.",
         "DataType": "string",
         "IsModifiable": true,
         "Source": "engine-default",
```

```
         "ParameterValue": "ISO, MDY",
         "ParameterName": "datestyle"
      },
      {
         "Description": "Sets the number of digits displayed for floating-point
 values",
         "DataType": "integer",
         "IsModifiable": true,
         "AllowedValues": "-15-2",
         "Source": "engine-default",
         "ParameterValue": "0",
         "ParameterName": "extra_float_digits"
      },
      (...remaining output omitted...)
   ]
}
```

您也可以使用 --output text選項，以文字格式取得相同的資訊。命令：

--output text option.Command：

option.Command：

```
aws redshift describe-cluster-parameters --parameter-group-name
 myclusterparametergroup --output text
```

結果：

```
RESPONSEMETADATA    cdac40aa-64cc-11e2-9e70-918437dd236d
Sets the display format for date and time values.   string  True    engine-default
 ISO, MDY        datestyle
Sets the number of digits displayed for floating-point values       integer True
 -15-2   engine-default  0       extra_float_digits
This parameter applies a user-defined label to a group of queries that are run
 during the same session..    string  True    engine-default  default query_group
require ssl for all databaseconnections     boolean True    true,false      engine-
default  false   require_ssl
Sets the schema search order for names that are not schema-qualified.       string
 True    engine-default  $user, public   search_path
Aborts any statement that takes over the specified number of milliseconds.  integer
 True    engine-default  0       statement_timeout
wlm json configuration      string  True    engine-default
 \[{"query_concurrency":5}]      wlm_json_configuration
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClusterParameters。

describe-cluster-security-groups

以下程式碼範例顯示如何使用 describe-cluster-security-groups。

AWS CLI

取得所有叢集安全GroupsThis範例會傳回帳戶所有叢集安全群組的描述。根據預設，輸出為 JSON 格式。命令：

```
aws redshift describe-cluster-security-groups
```

結果：

```
{
   "ClusterSecurityGroups": [
      {
         "OwnerId": "100447751468",
         "Description": "default",
         "ClusterSecurityGroupName": "default",
         "EC2SecurityGroups": \[],
         "IPRanges": [
            {
               "Status": "authorized",
               "CIDRIP": "0.0.0.0/0"
            }
         ]
      },
      {
         "OwnerId": "100447751468",
         "Description": "This is my cluster security group",
         "ClusterSecurityGroupName": "mysecuritygroup",
         "EC2SecurityGroups": \[],
         "IPRanges": \[]
      },
      (...remaining output omitted...)
   ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClusterSecurityGroups。

describe-cluster-snapshots

以下程式碼範例顯示如何使用 describe-cluster-snapshots。

AWS CLI

取得所有叢集SnapshotsThis範例會傳回帳戶所有叢集快照的描述。根據預設，輸出為 JSON 格式。命令：

```
aws redshift describe-cluster-snapshots
```

結果：

```
{
   "Snapshots": [
      {
         "Status": "available",
         "SnapshotCreateTime": "2013-07-17T22:02:22.852Z",
         "EstimatedSecondsToCompletion": -1,
         "AvailabilityZone": "us-east-1a",
         "ClusterVersion": "1.0",
         "MasterUsername": "adminuser",
         "Encrypted": false,
         "OwnerAccount": "111122223333",
         "BackupProgressInMegabytes": 20.0,
         "ElapsedTimeInSeconds": 0,
         "DBName": "dev",
         "CurrentBackupRateInMegabytesPerSecond: 0.0,
         "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
         "ActualIncrementalBackupSizeInMegabytes"; 20.0
         "SnapshotType": "automated",
         "NodeType": "dw.hs1.xlarge",
         "ClusterIdentifier": "mycluster",
         "Port": 5439,
         "TotalBackupSizeInMegabytes": 20.0,
         "NumberOfNodes": "2",
         "SnapshotIdentifier": "cm:mycluster-2013-01-22-22-04-18"
      },
      {
         "EstimatedSecondsToCompletion": 0,
         "OwnerAccount": "111122223333",
         "CurrentBackupRateInMegabytesPerSecond: 0.1534,
         "ActualIncrementalBackupSizeInMegabytes"; 11.0,
```

```
        "NumberOfNodes": "2",
        "Status": "available",
        "ClusterVersion": "1.0",
        "MasterUsername": "adminuser",
        "AccountsWithRestoreAccess": [
           {
              "AccountID": "444455556666"
           } ],
        "TotalBackupSizeInMegabytes": 20.0,
        "DBName": "dev",
        "BackupProgressInMegabytes": 11.0,
        "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
        "ElapsedTimeInSeconds": 0,
        "ClusterIdentifier": "mycluster",
        "SnapshotCreateTime": "2013-07-17T22:04:18.947Z",
        "AvailabilityZone": "us-east-1a",
        "NodeType": "dw.hs1.xlarge",
        "Encrypted": false,
        "SnapshotType": "manual",
        "Port": 5439,
        "SnapshotIdentifier": "my-snapshot-id"
     } ]
  }
  (...remaining output omitted...)
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClusterSnapshots。

describe-cluster-subnet-groups

以下程式碼範例顯示如何使用 describe-cluster-subnet-groups。

AWS CLI

取得所有叢集子網路GroupsThis範例會傳回所有叢集子網路群組的描述。根據預設，輸出為 JSON 格式。命令：

```
aws redshift describe-cluster-subnet-groups
```

結果：

```
{
   "ClusterSubnetGroups": [
```

```
      {
         "Subnets": [
            {
               "SubnetStatus": "Active",
               "SubnetIdentifier": "subnet-763fdd1c",
               "SubnetAvailabilityZone": {
                  "Name": "us-east-1a"
               }
            }
         ],
         "VpcId": "vpc-7e3fdd14",
         "SubnetGroupStatus": "Complete",
         "Description": "My subnet group",
         "ClusterSubnetGroupName": "mysubnetgroup"
      }
   ],
   "ResponseMetadata": {
      "RequestId": "37fa8c89-6990-11e2-8f75-ab4018764c77"
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClusterSubnetGroups。

describe-cluster-tracks

以下程式碼範例顯示如何使用 describe-cluster-tracks。

AWS CLI

描述叢集軌道

下列describe-cluster-tracks範例顯示可用維護軌道的詳細資訊。

```
aws redshift describe-cluster-tracks \
    --maintenance-track-name current
```

輸出：

```
{
    "MaintenanceTracks": [
        {
            "MaintenanceTrackName": "current",
            "DatabaseVersion": "1.0.11420",
```

```
            "UpdateTargets": [
                {
                    "MaintenanceTrackName": "preview_features",
                    "DatabaseVersion": "1.0.11746",
                    "SupportedOperations": [
                        {
                            "OperationName": "restore-from-cluster-snapshot"
                        }
                    ]
                },
                {
                    "MaintenanceTrackName": "trailing",
                    "DatabaseVersion": "1.0.11116",
                    "SupportedOperations": [
                        {
                            "OperationName": "restore-from-cluster-snapshot"
                        },
                        {
                            "OperationName": "modify-cluster"
                        }
                    ]
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的選擇叢集維護軌道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClusterTracks。

describe-cluster-versions

以下程式碼範例顯示如何使用 describe-cluster-versions。

AWS CLI

取得所有叢集VersionsThis範例會傳回所有叢集版本的描述。根據預設，輸出為 JSON 格式。命令：

```
aws redshift describe-cluster-versions
```

結果：

```
{
   "ClusterVersions": [
      {
      "ClusterVersion": "1.0",
      "Description": "Initial release",
      "ClusterParameterGroupFamily": "redshift-1.0"
      } ],
   "ResponseMetadata": {
      "RequestId": "16a53de3-64cc-11e2-bec0-17624ad140dd"
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClusterVersions。

describe-clusters

以下程式碼範例顯示如何使用 describe-clusters。

AWS CLI

取得所有ClustersThis範例會傳回帳戶所有叢集的描述。根據預設，輸出為 JSON 格式。命令：

```
aws redshift describe-clusters
```

結果：

```
{
   "Clusters": [
   {
      "NodeType": "dw.hs1.xlarge",
      "Endpoint": {
         "Port": 5439,
         "Address": "mycluster.coqoarplqhsn.us-east-1.redshift.amazonaws.com"
      },
      "ClusterVersion": "1.0",
      "PubliclyAccessible": "true",
      "MasterUsername": "adminuser",
      "ClusterParameterGroups": [
         {
            "ParameterApplyStatus": "in-sync",
            "ParameterGroupName": "default.redshift-1.0"
         } ],
```

```
      "ClusterSecurityGroups": [
         {
            "Status": "active",
            "ClusterSecurityGroupName": "default"
         } ],
      "AllowVersionUpgrade": true,
      "VpcSecurityGroups": \[],
      "AvailabilityZone": "us-east-1a",
      "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
      "PreferredMaintenanceWindow": "sat:03:30-sat:04:00",
      "AutomatedSnapshotRetentionPeriod": 1,
      "ClusterStatus": "available",
      "ClusterIdentifier": "mycluster",
      "DBName": "dev",
      "NumberOfNodes": 2,
      "PendingModifiedValues": {}
   } ],
   "ResponseMetadata": {
      "RequestId": "65b71cac-64df-11e2-8f5b-e90bd6c77476"
   }
}
```

您也可以使用 `--output text`選項，以文字格式取得相同的資訊。命令：

`--output text` option.Command：

option.Command：

```
aws redshift describe-clusters --output text
```

結果：

```
dw.hs1.xlarge       1.0     true    adminuser       True    us-east-1a
 2013-01-22T21:59:29.559Z        sat:03:30-sat:04:00     1       available
 mycluster       dev     2
ENDPOINT    5439    mycluster.coqoarplqhsn.us-east-1.redshift.amazonaws.com
in-sync     default.redshift-1.0
active      default
PENDINGMODIFIEDVALUES
RESPONSEMETADATA    934281a8-64df-11e2-b07c-f7fbdd006c67
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeClusters。

describe-default-cluster-parameters

以下程式碼範例顯示如何使用 describe-default-cluster-parameters。

AWS CLI

取得預設叢集ParametersThis範例會傳回 redshift-1.0 系列預設叢集參數的描述。根據預設，輸出為 JSON 格式。命令：

```
aws redshift describe-default-cluster-parameters --parameter-group-family
 redshift-1.0
```

結果：

```
{
   "DefaultClusterParameters": {
   "ParameterGroupFamily": "redshift-1.0",
   "Parameters": [
      {
         "Description": "Sets the display format for date and time values.",
         "DataType": "string",
         "IsModifiable": true,
         "Source": "engine-default",
         "ParameterValue": "ISO, MDY",
         "ParameterName": "datestyle"
      },
      {
         "Description": "Sets the number of digits displayed for floating-point
 values",
         "DataType": "integer",
         "IsModifiable": true,
         "AllowedValues": "-15-2",
         "Source": "engine-default",
         "ParameterValue": "0",
         "ParameterName": "extra_float_digits"
      },
      (...remaining output omitted...)
      ]
   }
}
```

若要查看有效參數群組系列的清單，請使用 describe-cluster-parameter-groups命令。

describe-cluster-parameter-groups 命令。

命令。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDefaultClusterParameters。

describe-event-categories

以下程式碼範例顯示如何使用 describe-event-categories。

AWS CLI

描述叢集的事件類別

下列describe-event-categories範例顯示叢集事件類別的詳細資訊。

```
aws redshift describe-event-categories \
    --source-type cluster
```

輸出：

```
{
    "EventCategoriesMapList": [
        {
            "SourceType": "cluster",
            "Events": [
                {
                    "EventId": "REDSHIFT-EVENT-2000",
                    "EventCategories": [
                        "management"
                    ],
                    "EventDescription": "Cluster <cluster name> created at <time in
 UTC>.",
                    "Severity": "INFO"
                },
                {
                    "EventId": "REDSHIFT-EVENT-2001",
                    "EventCategories": [
                        "management"
                    ],
                    "EventDescription": "Cluster <cluster name> deleted at <time in
 UTC>.",
                    "Severity": "INFO"
```

```
                },
                {
                    "EventId": "REDSHIFT-EVENT-3625",
                    "EventCategories": [
                        "monitoring"
                    ],
                    "EventDescription": "The cluster <cluster name> can't be resumed
 with its previous elastic network interface <ENI id>. We will allocate a new
 elastic network interface and associate it with the cluster node.",
                    "Severity": "INFO"
                }
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEventCategories。

describe-event-subscriptions

以下程式碼範例顯示如何使用 describe-event-subscriptions。

AWS CLI

描述事件訂閱

下列describe-event-subscriptions範例顯示指定訂閱的事件通知訂閱。

```
aws redshift describe-event-subscriptions \
    --subscription-name mysubscription
```

輸出：

```
{
    "EventSubscriptionsList": [
        {
            "CustomerAwsId": "123456789012",
            "CustSubscriptionId": "mysubscription",
            "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:MySNStopic",
            "Status": "active",
            "SubscriptionCreationTime": "2019-12-09T21:50:21.332Z",
            "SourceIdsList": [],
```

```
            "EventCategoriesList": [
                "management"
            ],
            "Severity": "ERROR",
            "Enabled": true,
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的訂閱 Amazon Redshift 事件通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEventSubscriptions。

`describe-events`

以下程式碼範例顯示如何使用 `describe-events`。

AWS CLI

描述所有事件 此範例會傳回所有事件。根據預設，輸出為 JSON 格式。命令：

```
aws redshift describe-events
```

結果：

```
{
   "Events": [
      {
      "Date": "2013-01-22T19:17:03.640Z",
      "SourceIdentifier": "myclusterparametergroup",
      "Message": "Cluster parameter group myclusterparametergroup has been
 created.",
      "SourceType": "cluster-parameter-group"
      } ],
   "ResponseMetadata": {
      "RequestId": "9f056111-64c9-11e2-9390-ff04f2c1e638"
   }
}
```

您也可以使用 `--output text`選項，以文字格式取得相同的資訊。命令：

`--output text` option.Command：

option.Command：

```
aws redshift describe-events --output text
```

結果：

```
2013-01-22T19:17:03.640Z    myclusterparametergroup Cluster parameter group
 myclusterparametergroup has been created.       cluster-parameter-group
RESPONSEMETADATA    8e5fe765-64c9-11e2-bce3-e56f52c50e17
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEvents。

describe-hsm-client-certificates

以下程式碼範例顯示如何使用 describe-hsm-client-certificates。

AWS CLI

描述 HSM 用戶端憑證

下列describe-hsm-client-certificates範例顯示指定 HSM 用戶端憑證的詳細資訊。

```
aws redshift describe-hsm-client-certificates \
    --hsm-client-certificate-identifier myhsmclientcert
```

輸出：

```
{
    "HsmClientCertificates": [
        {
        "HsmClientCertificateIdentifier": "myhsmclientcert",
        "HsmClientCertificatePublicKey": "-----BEGIN CERTIFICATE-----\
        EXAMPLECAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
        VVMxCzAJBgNVBAEXAMPLERAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
        b24xFDASBgNVBAsTC0lBTSBDb25zEXAMPLEwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
        BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhEXAMPLEDI1MjA0EXAMPLEN
        EXAMPLE0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
        VQQHEwdTZWF0dGEXAMPLEQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
        b2xlMRIwEAYDVQQDEwlUZXN0Q2lsEXAMPLEdBgkqhkiG9w0BCQEWEG5vb25lQGFt
        YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIEXAMPLEMaK0dn+a4GmWIWJ
        21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
```

```
        rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY67EXAMPLEE
        EXAMPLEZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
        nUhVVxYUntneD9EXAMPLE6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
        FFBjvSfpJIlJ00zbhNYS5f6GuoEDEXAMPLEBHjJnyp378OD8uTs7fLvjx79LjSTb
        NYiytVbZPQUQ5Yaxu2jXnimvw3rEXAMPLE=-----END CERTIFICATE-----\n",
        "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的 Amazon Redshift API 許可參考。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeHsmClientCertificates。

describe-hsm-configurations

以下程式碼範例顯示如何使用 describe-hsm-configurations。

AWS CLI

描述 HSM 組態

下列describe-hsm-configurations範例顯示呼叫 AWS 帳戶可用 HSM 組態的詳細資訊。

```
aws redshift describe-hsm-configurations /
    --hsm-configuration-identifier myhsmconnection
```

輸出：

```
{
    "HsmConfigurations": [
        {
            "HsmConfigurationIdentifier": "myhsmconnection",
            "Description": "My HSM connection",
            "HsmIpAddress": "192.0.2.09",
            "HsmPartitionName": "myhsmpartition",
            "Tags": []
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeHsmConfigurations。

describe-logging-status

以下程式碼範例顯示如何使用 describe-logging-status。

AWS CLI

描述叢集的記錄狀態

下列describe-logging-status範例顯示是否記錄叢集的資訊，例如查詢和連線嘗試。

```
aws redshift describe-logging-status \
    --cluster-identifier mycluster
```

輸出：

```
{
    "LoggingEnabled": false
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的資料庫稽核記錄。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLoggingStatus。

describe-node-configuration-options

以下程式碼範例顯示如何使用 describe-node-configuration-options。

AWS CLI

描述節點組態選項

下列describe-node-configuration-options範例顯示可能的節點組態屬性，例如節點類型、節點數量，以及指定叢集快照的磁碟用量。

```
aws redshift describe-node-configuration-options \
    --action-type restore-cluster \
    --snapshot-identifier rs:mycluster-2019-12-09-16-42-43
```

輸出：

```
{
    "NodeConfigurationOptionList": [
```

```
        {
            "NodeType": "dc2.large",
            "NumberOfNodes": 2,
            "EstimatedDiskUtilizationPercent": 19.61
        },
        {
            "NodeType": "dc2.large",
            "NumberOfNodes": 4,
            "EstimatedDiskUtilizationPercent": 9.96
        },
        {
            "NodeType": "ds2.xlarge",
            "NumberOfNodes": 2,
            "EstimatedDiskUtilizationPercent": 1.53
        },
        {
            "NodeType": "ds2.xlarge",
            "NumberOfNodes": 4,
            "EstimatedDiskUtilizationPercent": 0.78
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的購買 Amazon Redshift 預留節點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeNodeConfigurationOptions。

describe-orderable-cluster-options

以下程式碼範例顯示如何使用 describe-orderable-cluster-options。

AWS CLI

描述所有可排序叢集OptionsThis範例會傳回所有可排序叢集選項的說明。根據預設，輸出為 JSON 格式。命令：

```
aws redshift describe-orderable-cluster-options
```

結果：

```
{
   "OrderableClusterOptions": [
```

```
      {
         "NodeType": "dw.hs1.8xlarge",
         "AvailabilityZones": [
            { "Name": "us-east-1a" },
            { "Name": "us-east-1b" },
            { "Name": "us-east-1c" } ],
         "ClusterVersion": "1.0",
         "ClusterType": "multi-node"
      },
      {
         "NodeType": "dw.hs1.xlarge",
         "AvailabilityZones": [
            { "Name": "us-east-1a" },
            { "Name": "us-east-1b" },
            { "Name": "us-east-1c" } ],
         "ClusterVersion": "1.0",
         "ClusterType": "multi-node"
      },
      {
      "NodeType": "dw.hs1.xlarge",
      "AvailabilityZones": [
         { "Name": "us-east-1a" },
         { "Name": "us-east-1b" },
         { "Name": "us-east-1c" } ],
      "ClusterVersion": "1.0",
      "ClusterType": "single-node"
      } ],
   "ResponseMetadata": {
      "RequestId": "f6000035-64cb-11e2-9135-ff82df53a51a"
   }
}
```

您也可以使用 `--output text`選項，以文字格式取得相同的資訊。命令：

`--output text` option.Command：

option.Command：

```
aws redshift describe-orderable-cluster-options --output text
```

結果：

```
dw.hs1.8xlarge      1.0     multi-node
```

```
us-east-1a
us-east-1b
us-east-1c
dw.hs1.xlarge       1.0     multi-node
us-east-1a
us-east-1b
us-east-1c
dw.hs1.xlarge       1.0     single-node
us-east-1a
us-east-1b
us-east-1c
RESPONSEMETADATA    e648696b-64cb-11e2-bec0-17624ad140dd
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeOrderableClusterOptions。

`describe-reserved-node-offerings`

以下程式碼範例顯示如何使用 `describe-reserved-node-offerings`。

AWS CLI

描述預留節點OfferingsThis範例顯示所有可供購買的預留節點方案。命令：

```
aws redshift describe-reserved-node-offerings
```

結果：

```
{
   "ReservedNodeOfferings": [
      {
         "OfferingType": "Heavy Utilization",
         "FixedPrice": "",
         "NodeType": "dw.hs1.xlarge",
         "UsagePrice": "",
         "RecurringCharges": [
            {
               "RecurringChargeAmount": "",
               "RecurringChargeFrequency": "Hourly"
            } ],
         "Duration": 31536000,
         "ReservedNodeOfferingId": "ceb6a579-cf4c-4343-be8b-d832c45ab51c"
      },
```

```
      {
         "OfferingType": "Heavy Utilization",
         "FixedPrice": "",
         "NodeType": "dw.hs1.8xlarge",
         "UsagePrice": "",
         "RecurringCharges": [
            {
            "RecurringChargeAmount": "",
            "RecurringChargeFrequency": "Hourly"
            } ],
         "Duration": 31536000,
         "ReservedNodeOfferingId": "e5a2ff3b-352d-4a9c-ad7d-373c4cab5dd2"
      },
      ...remaining output omitted...
   ],
   "ResponseMetadata": {
      "RequestId": "8b1a1a43-75ff-11e2-9666-e142fe91ddd1"
   }
}
```

如果您想要購買預留節點方案，您可以使用purchase-reserved-node-offering有效的 ReservedNodeOfferingId 來呼叫 。

purchase-reserved-node-offering 使用有效的 ReservedNodeOfferingId。

使用有效的 ReservedNodeOfferingId。

ReservedNodeOfferingId。

.

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReservedNodeOfferings。

describe-reserved-nodes

以下程式碼範例顯示如何使用 describe-reserved-nodes。

AWS CLI

描述預留NodesThis範例顯示已購買的預留節點方案。命令：

```
aws redshift describe-reserved-nodes
```

結果：

```
{
   "ResponseMetadata": {
      "RequestId": "bc29ce2e-7600-11e2-9949-4b361e7420b7"
   },
   "ReservedNodes": [
      {
         "OfferingType": "Heavy Utilization",
         "FixedPrice": "",
         "NodeType": "dw.hs1.xlarge",
         "ReservedNodeId": "1ba8e2e3-bc01-4d65-b35d-a4a3e931547e",
         "UsagePrice": "",
         "RecurringCharges": [
            {
               "RecurringChargeAmount": "",
               "RecurringChargeFrequency": "Hourly"
            } ],
         "NodeCount": 1,
         "State": "payment-pending",
         "StartTime": "2013-02-13T17:08:39.051Z",
         "Duration": 31536000,
         "ReservedNodeOfferingId": "ceb6a579-cf4c-4343-be8b-d832c45ab51c"
      }
   ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeReservedNodes。

describe-resize

以下程式碼範例顯示如何使用 describe-resize。

AWS CLI

Describe ResizeThis範例說明叢集的最新調整大小。請求適用於 3 個 節點，類型為 dw.hs1.8xlarge.Command：

```
aws redshift describe-resize --cluster-identifier mycluster
```

結果：

```
{
   "Status": "NONE",
   "TargetClusterType": "multi-node",
   "TargetNodeType": "dw.hs1.8xlarge",
   "ResponseMetadata": {
      "RequestId": "9f52b0b4-7733-11e2-aa9b-318b2909bd27"
   },
   "TargetNumberOfNodes": "3"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeResize。

describe-scheduled-actions

以下程式碼範例顯示如何使用 describe-scheduled-actions。

AWS CLI

描述排程動作

下列describe-scheduled-actions範例顯示任何目前排程動作的詳細資訊。

```
aws redshift describe-scheduled-actions
```

輸出：

```
{
    "ScheduledActions": [
        {
            "ScheduledActionName": "resizecluster",
            "TargetAction": {
                "ResizeCluster": {
                    "ClusterIdentifier": "mycluster",
                    "NumberOfNodes": 4,
                    "Classic": false
                }
            },
            "Schedule": "at(2019-12-10T00:07:00)",
            "IamRole": "arn:aws:iam::123456789012:role/myRedshiftRole",
            "State": "ACTIVE",
            "NextInvocations": [
```

```
                "2019-12-10T00:07:00Z"
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeScheduledActions。

describe-snapshot-copy-grants

以下程式碼範例顯示如何使用 describe-snapshot-copy-grants。

AWS CLI

描述快照複製授予

下列describe-snapshot-copy-grants範例顯示指定叢集快照複製授予的詳細資訊。

```
aws redshift describe-snapshot-copy-grants \
    --snapshot-copy-grant-name mysnapshotcopygrantname
```

輸出：

```
{
    "SnapshotCopyGrants": [
        {
            "SnapshotCopyGrantName": "mysnapshotcopygrantname",
            "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
bPxRfih3yCo8nvbEXAMPLEKEY",
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的 Amazon Redshift 資料庫加密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSnapshotCopyGrants。

describe-snapshot-schedules

以下程式碼範例顯示如何使用 describe-snapshot-schedules。

AWS CLI

描述快照排程

下列describe-snapshot-schedules範例顯示指定叢集快照排程的詳細資訊。

```
aws redshift describe-snapshot-schedules \
    --cluster-identifier mycluster \
    --schedule-identifier mysnapshotschedule
```

輸出：

```
{
    "SnapshotSchedules": [
        {
            "ScheduleDefinitions": [
                "rate(12 hours)"
            ],
            "ScheduleIdentifier": "mysnapshotschedule",
            "ScheduleDescription": "My schedule description",
            "Tags": [],
            "AssociatedClusterCount": 1,
            "AssociatedClusters": [
                {
                    "ClusterIdentifier": "mycluster",
                    "ScheduleAssociationState": "ACTIVE"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的自動快照排程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSnapshotSchedules。

describe-storage

以下程式碼範例顯示如何使用 describe-storage。

AWS CLI

描述儲存體

下列describe-storage範例顯示 帳戶備份儲存體和臨時儲存體大小的詳細資訊。

```
aws redshift describe-storage
```

輸出：

```
{
    "TotalBackupSizeInMegaBytes": 193149.0,
    "TotalProvisionedStorageInMegaBytes": 655360.0
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的管理快照儲存。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStorage。

describe-table-restore-status

以下程式碼範例顯示如何使用 describe-table-restore-status。

AWS CLI

描述來自叢集快照的資料表還原請求狀態

下列describe-table-restore-status範例顯示對指定叢集提出的資料表還原請求的詳細資訊。

```
aws redshift describe-table-restore-status /
    --cluster-identifier mycluster
```

輸出：

```
{
    "TableRestoreStatusDetails": [
        {
            "TableRestoreRequestId": "z1116630-0e80-46f4-ba86-bd9670411ebd",
            "Status": "IN_PROGRESS",
            "RequestTime": "2019-12-27T18:22:12.257Z",
            "ClusterIdentifier": "mycluster",
            "SnapshotIdentifier": "mysnapshotid",
            "SourceDatabaseName": "dev",
```

```
            "SourceSchemaName": "public",
            "SourceTableName": "mytable",
            "TargetDatabaseName": "dev",
            "TargetSchemaName": "public",
            "NewTableName": "mytable-clone"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的從快照還原資料表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTableRestoreStatus。

describe-tags

以下程式碼範例顯示如何使用 describe-tags。

AWS CLI

描述標籤

下列describe-tags範例顯示與指定標籤名稱和值相關聯的指定叢集資源。

```
aws redshift describe-tags \
    --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \
    --tag-keys clustertagkey \
    --tag-values clustertagvalue
```

輸出：

```
{
    "TaggedResources": [
                    {
            "Tag": {
                "Key": "clustertagkey",
                "Value": "clustertagvalue"
            },
            "ResourceName": "arn:aws:redshift:us-
west-2:123456789012:cluster:mycluster",
            "ResourceType": "cluster"
        }
    ]
```

```
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的在 Amazon Redshift 中標記資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTags。

disable-snapshot-copy

以下程式碼範例顯示如何使用 disable-snapshot-copy。

AWS CLI

停用叢集的快照複製

下列disable-snapshot-copy範例會停用指定叢集快照的自動複製。

```
aws redshift disable-snapshot-copy \
    --cluster-identifier mycluster
```

輸出：

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "ClusterStatus": "available",
        "ClusterAvailabilityStatus": "Available",
        "MasterUsername": "adminuser",
        "DBName": "dev",
        "Endpoint": {
            "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
            "Port": 5439
        },
        "ClusterCreateTime": "2019-12-05T18:44:36.991Z",
        "AutomatedSnapshotRetentionPeriod": 3,
        "ManualSnapshotRetentionPeriod": -1,
        "ClusterSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sh-i9b431cd",
                "Status": "active"
            }
        ],
```

```
        "ClusterParameterGroups": [
            {
                "ParameterGroupName": "default.redshift-1.0",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "ClusterSubnetGroupName": "default",
        "VpcId": "vpc-b1fel7t9",
        "AvailabilityZone": "us-west-2f",
        "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
        "PendingModifiedValues": {
            "NodeType": "dc2.large",
            "NumberOfNodes": 2,
            "ClusterType": "multi-node"
        },
        "ClusterVersion": "1.0",
        "AllowVersionUpgrade": true,
        "NumberOfNodes": 4,
        "PubliclyAccessible": false,
        "Encrypted": false,
        "Tags": [
            {
                "Key": "mytags",
                "Value": "tag1"
            }
        ],
        "EnhancedVpcRouting": false,
        "IamRoles": [
            {
                "IamRoleArn": "arn:aws:iam::123456789012:role/myRedshiftRole",
                "ApplyStatus": "in-sync"
            }
        ],
        "MaintenanceTrackName": "current",
        "DeferredMaintenanceWindows": [],
        "ExpectedNextSnapshotScheduleTime": "2019-12-10T04:42:43.390Z",
        "ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
        "NextMaintenanceWindowStartTime": "2019-12-14T16:00:00Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的將快照複製到另一個 AWS 區域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableSnapshotCopy。

enable-snapshot-copy

以下程式碼範例顯示如何使用 enable-snapshot-copy。

AWS CLI

啟用叢集的快照複製

下列enable-snapshot-copy範例會啟用指定叢集快照的自動複製。

```
aws redshift enable-snapshot-copy \
    --cluster-identifier mycluster \
    --destination-region us-west-1
```

輸出：

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "ClusterStatus": "available",
        "ClusterAvailabilityStatus": "Available",
        "MasterUsername": "adminuser",
        "DBName": "dev",
        "Endpoint": {
            "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
            "Port": 5439
        },
        "ClusterCreateTime": "2019-12-05T18:44:36.991Z",
        "AutomatedSnapshotRetentionPeriod": 3,
        "ManualSnapshotRetentionPeriod": -1,
        "ClusterSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sh-f4c731cd",
                "Status": "active"
            }
        ],
        "ClusterParameterGroups": [
            {
                "ParameterGroupName": "default.redshift-1.0",
                "ParameterApplyStatus": "in-sync"
            }
```

```
        ],
        "ClusterSubnetGroupName": "default",
        "VpcId": "vpc-b1ael7t9",
        "AvailabilityZone": "us-west-2f",
        "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
        "PendingModifiedValues": {
            "NodeType": "dc2.large",
            "NumberOfNodes": 2,
            "ClusterType": "multi-node"
        },
        "ClusterVersion": "1.0",
        "AllowVersionUpgrade": true,
        "NumberOfNodes": 4,
        "PubliclyAccessible": false,
        "Encrypted": false,
        "ClusterSnapshotCopyStatus": {
            "DestinationRegion": "us-west-1",
            "RetentionPeriod": 7,
            "ManualSnapshotRetentionPeriod": -1
        },
        "Tags": [
            {
                "Key": "mytags",
                "Value": "tag1"
            }
        ],
        "EnhancedVpcRouting": false,
        "IamRoles": [
            {
                "IamRoleArn": "arn:aws:iam::123456789012:role/myRedshiftRole",
                "ApplyStatus": "in-sync"
            }
        ],
        "MaintenanceTrackName": "current",
        "DeferredMaintenanceWindows": [],
        "ExpectedNextSnapshotScheduleTime": "2019-12-10T04:42:43.390Z",
        "ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
        "NextMaintenanceWindowStartTime": "2019-12-14T16:00:00Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的將快照複製到其他 AWS 區域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableSnapshotCopy。

get-cluster-credentials

以下程式碼範例顯示如何使用 get-cluster-credentials。

AWS CLI

取得 AWS 帳戶的叢集登入資料

下列get-cluster-credentials範例會擷取啟用 Amazon Redshift 資料庫存取權的臨時登入資料。

```
aws redshift get-cluster-credentials \
    --db-user adminuser --db-name dev \
    --cluster-identifier mycluster
```

輸出：

```
{
    "DbUser": "IAM:adminuser",
    "DbPassword": "AMAFUyyuros/QjxPTtgzcsuQsqzIasdzJEN04aCtWDzXx1O9d6UmpkBtvEeqFly/
EXAMPLE==",
    "Expiration": "2019-12-10T17:25:05.770Z"
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的使用 Amazon Redshift CLI 或 API 產生 IAM 資料庫登入資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetClusterCredentials。

get-reserved-node-exchange-offerings

以下程式碼範例顯示如何使用 get-reserved-node-exchange-offerings。

AWS CLI

取得預留節點交換方案

下列get-reserved-node-exchange-offerings範例會擷取DC2ReservedNodeOfferings符合指定DC1預留節點的 陣列。

```
aws redshift get-reserved-node-exchange-offerings \
```

```
    --reserved-node-id 12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE
```

輸出：

```
{
    "ReservedNodeOfferings": [
        {
            "ReservedNodeOfferingId": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",
            "NodeType": "dc2.large",
            "Duration": 31536000,
            "FixedPrice": 0.0,
            "UsagePrice": 0.0,
            "CurrencyCode": "USD",
            "OfferingType": "All Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": 0.0,
                    "RecurringChargeFrequency": "Hourly"
                }
            ],
            "ReservedNodeOfferingType": "Regular"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的使用 AWS CLI 升級預留節點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetReservedNodeExchangeOfferings。

modify-cluster-iam-roles

以下程式碼範例顯示如何使用 modify-cluster-iam-roles。

AWS CLI

修改叢集的 IAM 角色

下列modify-cluster-iam-roles範例會從指定的叢集移除指定的 AWS IAM 角色。

```
aws redshift modify-cluster-iam-roles \
    --cluster-identifier mycluster \
```

```
    --remove-iam-roles arn:aws:iam::123456789012:role/myRedshiftRole
```

輸出：

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "ClusterStatus": "available",
        "ClusterAvailabilityStatus": "Available",
        "MasterUsername": "adminuser",
        "DBName": "dev",
        "Endpoint": {
            "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
            "Port": 5439
        },
        "ClusterCreateTime": "2019-12-05T18:44:36.991Z",
        "AutomatedSnapshotRetentionPeriod": 3,
        "ManualSnapshotRetentionPeriod": -1,
        "ClusterSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sh-f9b731sd",
                "Status": "active"
            }
        ],
        "ClusterParameterGroups": [
            {
                "ParameterGroupName": "default.redshift-1.0",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "ClusterSubnetGroupName": "default",
        "VpcId": "vpc-b2fal7t9",
        "AvailabilityZone": "us-west-2f",
        "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
        "PendingModifiedValues": {
            "NodeType": "dc2.large",
            "NumberOfNodes": 2,
            "ClusterType": "multi-node"
        },
        "ClusterVersion": "1.0",
        "AllowVersionUpgrade": true,
```

```
        "NumberOfNodes": 4,
        "PubliclyAccessible": false,
        "Encrypted": false,
        "ClusterSnapshotCopyStatus": {
            "DestinationRegion": "us-west-1",
            "RetentionPeriod": 7,
            "ManualSnapshotRetentionPeriod": -1
        },
        "Tags": [
            {
                "Key": "mytags",
                "Value": "tag1"
            }
        ],
        "EnhancedVpcRouting": false,
        "IamRoles": [],
        "MaintenanceTrackName": "current",
        "DeferredMaintenanceWindows": [],
        "ExpectedNextSnapshotScheduleTime": "2019-12-11T04:42:55.631Z",
        "ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
        "NextMaintenanceWindowStartTime": "2019-12-14T16:00:00Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的針對 Amazon Redshift 使用以身分為基礎的政策 (IAM 政策)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyClusterIamRoles。

modify-cluster-maintenance

以下程式碼範例顯示如何使用 modify-cluster-maintenance。

AWS CLI

修改叢集維護

下列modify-cluster-maintenance範例會將指定叢集的維護延遲 30 天。

```
aws redshift modify-cluster-maintenance \
    --cluster-identifier mycluster \
    --defer-maintenance \
    --defer-maintenance-duration 30
```

輸出：


```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "ClusterStatus": "available",
        "ClusterAvailabilityStatus": "Available",
        "MasterUsername": "adminuser",
        "DBName": "dev",
        "Endpoint": {
            "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
            "Port": 5439
        },
        "ClusterCreateTime": "2019-12-05T18:44:36.991Z",
        "AutomatedSnapshotRetentionPeriod": 3,
        "ManualSnapshotRetentionPeriod": -1,
        "ClusterSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sh-a1a123ab",
                "Status": "active"
            }
        ],
        "ClusterParameterGroups": [
            {
                "ParameterGroupName": "default.redshift-1.0",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "ClusterSubnetGroupName": "default",
        "VpcId": "vpc-b1ael7t9",
        "AvailabilityZone": "us-west-2f",
        "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
        "PendingModifiedValues": {
            "NodeType": "dc2.large",
            "NumberOfNodes": 2,
            "ClusterType": "multi-node"
        },
        "ClusterVersion": "1.0",
        "AllowVersionUpgrade": true,
        "NumberOfNodes": 4,
        "PubliclyAccessible": false,
        "Encrypted": false,
```

```
        "ClusterSnapshotCopyStatus": {
            "DestinationRegion": "us-west-1",
            "RetentionPeriod": 7,
            "ManualSnapshotRetentionPeriod": -1
        },
        "Tags": [
            {
                "Key": "mytags",
                "Value": "tag1"
            }
        ],
        "EnhancedVpcRouting": false,
        "IamRoles": [],
        "MaintenanceTrackName": "current",
        "DeferredMaintenanceWindows": [
            {
                "DeferMaintenanceIdentifier": "dfm-mUdVIfFcT1B4SGhw6fyF",
                "DeferMaintenanceStartTime": "2019-12-10T18:18:39.354Z",
                "DeferMaintenanceEndTime": "2020-01-09T18:18:39.354Z"
            }
        ],
        "ExpectedNextSnapshotScheduleTime": "2019-12-11T04:42:55.631Z",
        "ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
        "NextMaintenanceWindowStartTime": "2020-01-11T16:00:00Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的叢集維護。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyClusterMaintenance。

modify-cluster-parameter-group

以下程式碼範例顯示如何使用 modify-cluster-parameter-group。

AWS CLI

修改參數群組中的參數

下列modify-cluster-parameter-group範例會修改工作負載管理的 wlm_json_configuration 參數。它接受檔案中的參數，其中包含如下所示的 JSON 內容。

```
aws redshift modify-cluster-parameter-group \
```

```
    --parameter-group-name myclusterparametergroup \
    --parameters file://modify_pg.json
```

modify_pg.json 的內容：

```
[
    {
        "ParameterName": "wlm_json_configuration",
        "ParameterValue": "[{\"user_group\":\"example_user_group1\",\"query_group\":
 \"example_query_group1\", \"query_concurrency\":7},{\"query_concurrency\":5}]"
    }
]
```

輸出：

```
{
   "ParameterGroupStatus": "Your parameter group has been updated but changes won't
 get applied until you reboot the associated Clusters.",
   "ParameterGroupName": "myclusterparametergroup",
   "ResponseMetadata": {
      "RequestId": "09974cc0-64cd-11e2-bea9-49e0ce183f07"
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyClusterParameterGroup。

modify-cluster-snapshot-schedule

以下程式碼範例顯示如何使用 modify-cluster-snapshot-schedule。

AWS CLI

修改叢集快照排程

下列modify-cluster-snapshot-schedule範例會從指定的叢集移除指定的快照排程。

```
aws redshift modify-cluster-snapshot-schedule \
    --cluster-identifier mycluster \
    --schedule-identifier mysnapshotschedule \
    --disassociate-schedule
```

此命令不會產生任何輸出。

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的自動快照排程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyClusterSnapshotSchedule。

modify-cluster-snapshot

以下程式碼範例顯示如何使用 modify-cluster-snapshot。

AWS CLI

修改叢集快照

下列modify-cluster-snapshot範例會將指定叢集快照的手動保留期間設定設為 10 天的值。

```
aws redshift modify-cluster-snapshot \
    --snapshot-identifier mycluster-2019-11-06-16-32 \
    --manual-snapshot-retention-period 10
```

輸出：

```
{
    "Snapshot": {
        "SnapshotIdentifier": "mycluster-2019-11-06-16-32",
        "ClusterIdentifier": "mycluster",
        "SnapshotCreateTime": "2019-12-07T00:34:05.633Z",
        "Status": "available",
        "Port": 5439,
        "AvailabilityZone": "us-west-2f",
        "ClusterCreateTime": "2019-12-05T18:44:36.991Z",
        "MasterUsername": "adminuser",
        "ClusterVersion": "1.0",
        "SnapshotType": "manual",
        "NodeType": "dc2.large",
        "NumberOfNodes": 2,
        "DBName": "dev",
        "VpcId": "vpc-b1cel7t9",
        "Encrypted": false,
        "EncryptedWithHSM": false,
        "OwnerAccount": "123456789012",
        "TotalBackupSizeInMegaBytes": 64384.0,
        "ActualIncrementalBackupSizeInMegaBytes": 24.0,
        "BackupProgressInMegaBytes": 24.0,
        "CurrentBackupRateInMegaBytesPerSecond": 13.0011,
```

```
        "EstimatedSecondsToCompletion": 0,
        "ElapsedTimeInSeconds": 1,
        "Tags": [
            {
                "Key": "mytagkey",
                "Value": "mytagvalue"
            }
        ],
        "EnhancedVpcRouting": false,
        "MaintenanceTrackName": "current",
        "ManualSnapshotRetentionPeriod": 10,
        "ManualSnapshotRemainingDays": 6,
        "SnapshotRetentionStartTime": "2019-12-07T00:34:07.479Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的 Amazon Redshift 快照。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyClusterSnapshot。

`modify-cluster-subnet-group`

以下程式碼範例顯示如何使用 `modify-cluster-subnet-group`。

AWS CLI

修改叢集子網路GroupThis範例說明如何修改快取子網路群組中的子網路清單。根據預設，輸出為 JSON 格式。命令：

```
aws redshift modify-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
 --subnet-ids subnet-763fdd1 subnet-ac830e9
```

結果：

```
{
   "ClusterSubnetGroup":
   {
      "Subnets": [
         {
            "SubnetStatus": "Active",
            "SubnetIdentifier": "subnet-763fdd1c",
            "SubnetAvailabilityZone":
```

```
                { "Name": "us-east-1a" }
         },
         {
            "SubnetStatus": "Active",
            "SubnetIdentifier": "subnet-ac830e9",
            "SubnetAvailabilityZone":
               { "Name": "us-east-1b" }
         } ],
      "VpcId": "vpc-7e3fdd14",
      "SubnetGroupStatus": "Complete",
      "Description": "My subnet group",
      "ClusterSubnetGroupName": "mysubnetgroup"
   },
   "ResponseMetadata": {
      "RequestId": "8da93e89-8372-f936-93a8-873918938197a"
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyClusterSubnetGroup。

`modify-cluster`

以下程式碼範例顯示如何使用 `modify-cluster`。

AWS CLI

將安全群組與ClusterThis範例說明如何將叢集安全群組與指定的叢集建立關聯。命令：

```
aws redshift modify-cluster --cluster-identifier mycluster --cluster-security-groups
 mysecuritygroup
```

修改ClusterThis顯示如何將叢集的每週偏好維護時段變更為從週日下午 11：15 開始，到週一上午 3：15 結束的最短四個小時時段。命令：

```
aws redshift modify-cluster --cluster-identifier mycluster --preferred-maintenance-
window Sun:23:15-Mon:03:15
```

變更 ClusterThis範例說明如何變更叢集的主密碼。命令：

```
aws redshift modify-cluster --cluster-identifier mycluster --master-user-password
 A1b2c3d4
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyCluster。

modify-event-subscription

以下程式碼範例顯示如何使用 modify-event-subscription。

AWS CLI

修改事件訂閱

下列modify-event-subscription範例會停用指定的事件通知訂閱。

```
aws redshift modify-event-subscription \
    --subscription-name mysubscription \
    --no-enabled
```

輸出：

```
{
    "EventSubscription": {
        "CustomerAwsId": "123456789012",
        "CustSubscriptionId": "mysubscription",
        "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:MySNStopic",
        "Status": "active",
        "SubscriptionCreationTime": "2019-12-09T21:50:21.332Z",
        "SourceIdsList": [],
        "EventCategoriesList": [
            "management"
        ],
        "Severity": "ERROR",
        "Enabled": false,
        "Tags": []
    }
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的訂閱 Amazon Redshift 事件通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyEventSubscription。

modify-scheduled-action

以下程式碼範例顯示如何使用 modify-scheduled-action。

AWS CLI

修改排程動作

下列modify-scheduled-action範例會將描述新增至指定的現有排程動作。

```
aws redshift modify-scheduled-action \
    --scheduled-action-name myscheduledaction \
    --scheduled-action-description "My scheduled action"
```

輸出：

```
{
    "ScheduledActionName": "myscheduledaction",
    "TargetAction": {
        "ResizeCluster": {
            "ClusterIdentifier": "mycluster",
            "NumberOfNodes": 2,
            "Classic": false
        }
    },
    "Schedule": "at(2019-12-25T00:00:00)",
    "IamRole": "arn:aws:iam::123456789012:role/myRedshiftRole",
    "ScheduledActionDescription": "My scheduled action",
    "State": "ACTIVE",
    "NextInvocations": [
        "2019-12-25T00:00:00Z"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyScheduledAction。

modify-snapshot-copy-retention-period

以下程式碼範例顯示如何使用 modify-snapshot-copy-retention-period。

AWS CLI

修改快照複製保留期

下列modify-snapshot-copy-retention-period範例會修改從來源 AWS 區域複製快照後，在目的地區域中保留指定叢集快照的天數 AWS 。

```
aws redshift modify-snapshot-copy-retention-period \
    --cluster-identifier mycluster \
    --retention-period 15
```

輸出：

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "ClusterStatus": "available",
        "ClusterAvailabilityStatus": "Available",
        "MasterUsername": "adminuser",
        "DBName": "dev",
        "Endpoint": {
            "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
            "Port": 5439
        },
        "ClusterCreateTime": "2019-12-05T18:44:36.991Z",
        "AutomatedSnapshotRetentionPeriod": 3,
        "ManualSnapshotRetentionPeriod": -1,
        "ClusterSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sh-a1a123ab",
                "Status": "active"
            }
        ],
        "ClusterParameterGroups": [
            {
                "ParameterGroupName": "default.redshift-1.0",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "ClusterSubnetGroupName": "default",
        "VpcId": "vpc-b1fet7t9",
        "AvailabilityZone": "us-west-2f",
        "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
        "PendingModifiedValues": {
            "NodeType": "dc2.large",
            "NumberOfNodes": 2,
            "ClusterType": "multi-node"
        },
```

```
        "ClusterVersion": "1.0",
        "AllowVersionUpgrade": true,
        "NumberOfNodes": 4,
        "PubliclyAccessible": false,
        "Encrypted": false,
        "ClusterSnapshotCopyStatus": {
            "DestinationRegion": "us-west-1",
            "RetentionPeriod": 15,
            "ManualSnapshotRetentionPeriod": -1
        },
        "Tags": [
            {
                "Key": "mytags",
                "Value": "tag1"
            }
        ],
        "EnhancedVpcRouting": false,
        "IamRoles": [],
        "MaintenanceTrackName": "current",
        "DeferredMaintenanceWindows": [
            {
                "DeferMaintenanceIdentifier": "dfm-mUdVSfDcT1F4SGhw6fyF",
                "DeferMaintenanceStartTime": "2019-12-10T18:18:39.354Z",
                "DeferMaintenanceEndTime": "2020-01-09T18:18:39.354Z"
            }
        ],
        "NextMaintenanceWindowStartTime": "2020-01-11T16:00:00Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的快照排程格式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifySnapshotCopyRetentionPeriod。

modify-snapshot-schedule

以下程式碼範例顯示如何使用 modify-snapshot-schedule。

AWS CLI

修改快照排程

下列modify-snapshot-schedule範例會將指定快照排程的速率修改為每 10 小時一次。

```
aws redshift modify-snapshot-schedule \
    --schedule-identifier mysnapshotschedule \
    --schedule-definitions "rate(10 hours)"
```

輸出：

```
{
    "ScheduleDefinitions": [
        "rate(10 hours)"
    ],
    "ScheduleIdentifier": "mysnapshotschedule",
    "ScheduleDescription": "My schedule description",
    "Tags": []
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的快照排程格式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifySnapshotSchedule。

purchase-reserved-node-offering

以下程式碼範例顯示如何使用 purchase-reserved-node-offering。

AWS CLI

購買預留NodeThis範例示範如何購買預留節點方案。可透過呼叫 describe-reserved-node-offerings.Command 來reserved-node-offering-id取得：

```
aws redshift purchase-reserved-node-offering --reserved-node-offering-id ceb6a579-cf4c-4343-be8b-d832c45ab51c
```

結果：

```
{
   "ReservedNode": {
      "OfferingType": "Heavy Utilization",
      "FixedPrice": "",
      "NodeType": "dw.hs1.xlarge",
      "ReservedNodeId": "1ba8e2e3-bc01-4d65-b35d-a4a3e931547e",
```

```
      "UsagePrice": "",
      "RecurringCharges": [
         {
            "RecurringChargeAmount": "",
            "RecurringChargeFrequency": "Hourly"
         }
      ],
      "NodeCount": 1,
      "State": "payment-pending",
      "StartTime": "2013-02-13T17:08:39.051Z",
      "Duration": 31536000,
      "ReservedNodeOfferingId": "ceb6a579-cf4c-4343-be8b-d832c45ab51c"
   },
   "ResponseMetadata": {
      "RequestId": "01bda7bf-7600-11e2-b605-2568d7396e7f"
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PurchaseReservedNodeOffering。

reboot-cluster

以下程式碼範例顯示如何使用 reboot-cluster。

AWS CLI

重新啟動ClusterThis範例會重新啟動叢集。根據預設，輸出為 JSON 格式。命令：

```
aws redshift reboot-cluster --cluster-identifier mycluster
```

結果：

```
{
   "Cluster": {
      "NodeType": "dw.hs1.xlarge",
      "Endpoint": {
         "Port": 5439,
         "Address": "mycluster.coqoarplqhsn.us-east-1.redshift.amazonaws.com"
      },
      "ClusterVersion": "1.0",
      "PubliclyAccessible": "true",
      "MasterUsername": "adminuser",
```

```
      "ClusterParameterGroups": [
         {
         "ParameterApplyStatus": "in-sync",
         "ParameterGroupName": "default.redshift-1.0"
         }
      ],
      "ClusterSecurityGroups": [
         {
         "Status": "active",
         "ClusterSecurityGroupName": "default"
         }
      ],
      "AllowVersionUpgrade": true,
      "VpcSecurityGroups": \[],
      "AvailabilityZone": "us-east-1a",
      "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
      "PreferredMaintenanceWindow": "sun:23:15-mon:03:15",
      "AutomatedSnapshotRetentionPeriod": 1,
      "ClusterStatus": "rebooting",
      "ClusterIdentifier": "mycluster",
      "DBName": "dev",
      "NumberOfNodes": 2,
      "PendingModifiedValues": {}
   },
   "ResponseMetadata": {
      "RequestId": "61c8b564-64e8-11e2-8f7d-3b939af52818"
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RebootCluster。

reset-cluster-parameter-group

以下程式碼範例顯示如何使用 reset-cluster-parameter-group。

AWS CLI

參數GroupThis範例說明如何重設參數群組中的所有參數。命令：

```
aws redshift reset-cluster-parameter-group --parameter-group-name
 myclusterparametergroup --reset-all-parameters
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetClusterParameterGroup。

resize-cluster

以下程式碼範例顯示如何使用 resize-cluster。

AWS CLI

調整叢集大小

下列resize-cluster範例會調整指定叢集的大小。

```
aws redshift resize-cluster \
    --cluster-identifier mycluster \
    --cluster-type multi-node \
    --node-type dc2.large \
    --number-of-nodes 6 \
    --classic
```

輸出：

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "ClusterStatus": "resizing",
        "ClusterAvailabilityStatus": "Modifying",
        "MasterUsername": "adminuser",
        "DBName": "dev",
        "Endpoint": {
            "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
            "Port": 5439
        },
        "ClusterCreateTime": "2019-12-05T18:44:36.991Z",
        "AutomatedSnapshotRetentionPeriod": 3,
        "ManualSnapshotRetentionPeriod": -1,
        "ClusterSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sh-a1a123ab",
                "Status": "active"
            }
        ],
        "ClusterParameterGroups": [
            {
                "ParameterGroupName": "default.redshift-1.0",
```

```
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "ClusterSubnetGroupName": "default",
        "VpcId": "vpc-a1abc1a1",
        "AvailabilityZone": "us-west-2f",
        "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
        "PendingModifiedValues": {
            "NodeType": "dc2.large",
            "NumberOfNodes": 6,
            "ClusterType": "multi-node"
        },
        "ClusterVersion": "1.0",
        "AllowVersionUpgrade": true,
        "NumberOfNodes": 4,
        "PubliclyAccessible": false,
        "Encrypted": false,
        "ClusterSnapshotCopyStatus": {
            "DestinationRegion": "us-west-1",
            "RetentionPeriod": 15,
            "ManualSnapshotRetentionPeriod": -1
        },
        "Tags": [
            {
                "Key": "mytags",
                "Value": "tag1"
            }
        ],
        "EnhancedVpcRouting": false,
        "IamRoles": [],
        "MaintenanceTrackName": "current",
        "DeferredMaintenanceWindows": [
            {
                "DeferMaintenanceIdentifier": "dfm-mUdVCfDcT1B4SGhw6fyF",
                "DeferMaintenanceStartTime": "2019-12-10T18:18:39.354Z",
                "DeferMaintenanceEndTime": "2020-01-09T18:18:39.354Z"
            }
        ],
        "NextMaintenanceWindowStartTime": "2020-01-11T16:00:00Z",
        "ResizeInfo": {
            "ResizeType": "ClassicResize",
            "AllowCancelResize": true
        }
    }
```

```
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的調整叢集的大小。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResizeCluster。

`restore-from-cluster-snapshot`

以下程式碼範例顯示如何使用 `restore-from-cluster-snapshot`。

AWS CLI

從SnapshotThis範例會從快照還原叢集。命令：

```
aws redshift restore-from-cluster-snapshot --cluster-identifier mycluster-clone --
snapshot-identifier my-snapshot-id
```

結果：

```
{
   "Cluster": {
      "NodeType": "dw.hs1.xlarge",
      "ClusterVersion": "1.0",
      "PubliclyAccessible": "true",
      "MasterUsername": "adminuser",
      "ClusterParameterGroups": [
         {
         "ParameterApplyStatus": "in-sync",
         "ParameterGroupName": "default.redshift-1.0"
         }
      ],
      "ClusterSecurityGroups": [
         {
         "Status": "active",
         "ClusterSecurityGroupName": "default"
         }
      ],
      "AllowVersionUpgrade": true,
      "VpcSecurityGroups": \[],
      "PreferredMaintenanceWindow": "sun:23:15-mon:03:15",
      "AutomatedSnapshotRetentionPeriod": 1,
      "ClusterStatus": "creating",
```

```
        "ClusterIdentifier": "mycluster-clone",
        "DBName": "dev",
        "NumberOfNodes": 2,
        "PendingModifiedValues": {}
    },
    "ResponseMetadata": {
        "RequestId": "77fd512b-64e3-11e2-8f5b-e90bd6c77476"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreFromClusterSnapshot。

restore-table-from-cluster-snapshot

以下程式碼範例顯示如何使用 restore-table-from-cluster-snapshot。

AWS CLI

從叢集快照還原資料表

下列restore-table-from-cluster-snapshot範例會從指定叢集快照中的指定資料表建立新的資料表。

```
aws redshift restore-table-from-cluster-snapshot /
    --cluster-identifier mycluster /
    --snapshot-identifier mycluster-2019-11-19-16-17 /
    --source-database-name dev /
    --source-schema-name public /
    --source-table-name mytable /
    --target-database-name dev /
    --target-schema-name public /
    --new-table-name mytable-clone
```

輸出：

```
{
    "TableRestoreStatus": {
        "TableRestoreRequestId": "a123a12b-abc1-1a1a-a123-a1234ab12345",
        "Status": "PENDING",
        "RequestTime": "2019-12-20T00:20:16.402Z",
        "ClusterIdentifier": "mycluster",
        "SnapshotIdentifier": "mycluster-2019-11-19-16-17",
```

```
        "SourceDatabaseName": "dev",
        "SourceSchemaName": "public",
        "SourceTableName": "mytable",
        "TargetDatabaseName": "dev",
        "TargetSchemaName": "public",
        "NewTableName": "mytable-clone"
    }
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的從快照還原資料表。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreTableFromClusterSnapshot。

`revoke-cluster-security-group-ingress`

以下程式碼範例顯示如何使用 `revoke-cluster-security-group-ingress`。

AWS CLI

從 EC2 安全GroupThis範例撤銷對名為 Amazon EC2 安全群組的存取權。命令：

```
aws redshift revoke-cluster-security-group-ingress --cluster-security-group-name
 mysecuritygroup --ec2-security-group-name myec2securitygroup --ec2-security-group-
owner-id 123445677890
```

撤銷對 CIDR rangeThis範例撤銷對 CIDR 範圍的存取權。命令：

```
aws redshift revoke-cluster-security-group-ingress --cluster-security-group-name
 mysecuritygroup --cidrip 192.168.100.100/32
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RevokeClusterSecurityGroupIngress。

`revoke-snapshot-access`

以下程式碼範例顯示如何使用 `revoke-snapshot-access`。

AWS CLI

撤銷 AWS 帳戶還原SnapshotThis範例會撤銷 AWS 帳戶444455556666還原快照 的授權my-snapshot-id。根據預設，輸出為 JSON 格式。命令：

```
aws redshift revoke-snapshot-access --snapshot-id my-snapshot-id --account-with-
restore-access 444455556666
```

結果：

```
{
   "Snapshot": {
      "Status": "available",
      "SnapshotCreateTime": "2013-07-17T22:04:18.947Z",
      "EstimatedSecondsToCompletion": 0,
      "AvailabilityZone": "us-east-1a",
      "ClusterVersion": "1.0",
      "MasterUsername": "adminuser",
      "Encrypted": false,
      "OwnerAccount": "111122223333",
      "BackupProgressInMegabytes": 11.0,
      "ElapsedTimeInSeconds": 0,
      "DBName": "dev",
      "CurrentBackupRateInMegabytesPerSecond: 0.1534,
      "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
      "ActualIncrementalBackupSizeInMegabytes"; 11.0,
      "SnapshotType": "manual",
      "NodeType": "dw.hs1.xlarge",
      "ClusterIdentifier": "mycluster",
      "TotalBackupSizeInMegabytes": 20.0,
      "Port": 5439,
      "NumberOfNodes": 2,
      "SnapshotIdentifier": "my-snapshot-id"
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RevokeSnapshotAccess。

rotate-encryption-key

以下程式碼範例顯示如何使用 rotate-encryption-key。

AWS CLI

輪換叢集的加密金鑰

下列rotate-encryption-key範例會輪換指定叢集的加密金鑰。

```
aws redshift rotate-encryption-key \
    --cluster-identifier mycluster
```

輸出：

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "ClusterStatus": "rotating-keys",
        "ClusterAvailabilityStatus": "Modifying",
        "MasterUsername": "adminuser",
        "DBName": "dev",
        "Endpoint": {
            "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
            "Port": 5439
        },
        "ClusterCreateTime": "2019-12-10T19:25:45.886Z",
        "AutomatedSnapshotRetentionPeriod": 30,
        "ManualSnapshotRetentionPeriod": -1,
        "ClusterSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sh-a1a123ab",
                "Status": "active"
            }
        ],
        "ClusterParameterGroups": [
            {
                "ParameterGroupName": "default.redshift-1.0",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "ClusterSubnetGroupName": "default",
        "VpcId": "vpc-a1abc1a1",
        "AvailabilityZone": "us-west-2a",
        "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
        "PendingModifiedValues": {},
        "ClusterVersion": "1.0",
        "AllowVersionUpgrade": true,
        "NumberOfNodes": 2,
        "PubliclyAccessible": false,
        "Encrypted": true,
```

```
        "Tags": [],
        "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
bPxRfih3yCo8nvbEXAMPLEKEY",
        "EnhancedVpcRouting": false,
        "IamRoles": [
            {
                "IamRoleArn": "arn:aws:iam::123456789012:role/myRedshiftRole",
                "ApplyStatus": "in-sync"
            }
        ],
        "MaintenanceTrackName": "current",
        "DeferredMaintenanceWindows": [],
        "NextMaintenanceWindowStartTime": "2019-12-14T16:00:00Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Redshift 叢集管理指南》中的 Amazon Redshift 資料庫加密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RotateEncryptionKey。

使用 的 Amazon Rekognition 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon Rekognition 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`compare-faces`

以下程式碼範例顯示如何使用 `compare-faces`。

如需詳細資訊，請參閱比較映像中的人臉。

AWS CLI

比較兩個影像中的臉部

下列compare-faces命令會比較存放在 Amazon S3 儲存貯體中的兩個影像中的臉部。

```
aws rekognition compare-faces \
    --source-image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"source.jpg"}}' \
    --target-image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"target.jpg"}}'
```

輸出：

```
{
    "UnmatchedFaces": [],
    "FaceMatches": [
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.12368916720151901,
                    "Top": 0.16007372736930847,
                    "Left": 0.5901257991790771,
                    "Height": 0.25140416622161865
                },
                "Confidence": 100.0,
                "Pose": {
                    "Yaw": -3.7351467609405518,
                    "Roll": -0.10309021919965744,
                    "Pitch": 0.8637830018997192
                },
                "Quality": {
                    "Sharpness": 95.51618957519531,
                    "Brightness": 65.29893493652344
                },
                "Landmarks": [
                    {
                        "Y": 0.26721030473709106,
                        "X": 0.6204193830490112,
                        "Type": "eyeLeft"
                    },
                    {
                        "Y": 0.26831310987472534,
                        "X": 0.6776827573776245,
                        "Type": "eyeRight"
```

```
                    },
                    {
                        "Y": 0.3514654338359833,
                        "X": 0.6241428852081299,
                        "Type": "mouthLeft"
                    },
                    {
                        "Y": 0.35258132219314575,
                        "X": 0.6713621020317078,
                        "Type": "mouthRight"
                    },
                    {
                        "Y": 0.3140771687030792,
                        "X": 0.6428444981575012,
                        "Type": "nose"
                    }
                ]
            },
            "Similarity": 100.0
        }
    ],
    "SourceImageFace": {
        "BoundingBox": {
            "Width": 0.12368916720151901,
            "Top": 0.16007372736930847,
            "Left": 0.5901257991790771,
            "Height": 0.25140416622161865
        },
        "Confidence": 100.0
    }
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的在影像中比較臉部。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CompareFaces。

create-collection

以下程式碼範例顯示如何使用 create-collection。

如需更多資訊，請參閱建立集合。

AWS CLI

建立集合

下列create-collection命令會建立具有指定名稱的集合。

```
aws rekognition create-collection \
    --collection-id "MyCollection"
```

輸出：

```
{
    "CollectionArn": "aws:rekognition:us-west-2:123456789012:collection/
MyCollection",
    "FaceModelVersion": "4.0",
    "StatusCode": 200
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的建立集合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCollection。

create-stream-processor

以下程式碼範例顯示如何使用 create-stream-processor。

AWS CLI

建立新的串流處理器

下列create-stream-processor範例會使用指定的組態建立新的串流處理器。

```
aws rekognition create-stream-processor --name my-stream-processor\
    --input '{"KinesisVideoStream":{"Arn":"arn:aws:kinesisvideo:us-
west-2:123456789012:stream/macwebcam/1530559711205"}}'\
    --stream-processor-output '{"KinesisDataStream":{"Arn":"arn:aws:kinesis:us-
west-2:123456789012:stream/AmazonRekognitionRekStream"}}'\
    --role-arn arn:aws:iam::123456789012:role/AmazonRekognitionDetect\
    --settings '{"FaceSearch":
{"CollectionId":"MyCollection","FaceMatchThreshold":85.5}}'
```

輸出：

```
{
    "StreamProcessorArn": "arn:aws:rekognition:us-
west-2:123456789012:streamprocessor/my-stream-processor"
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的使用串流影片。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateStreamProcessor。

delete-collection

以下程式碼範例顯示如何使用 delete-collection。

如需更多資訊，請參閱刪除集合。

AWS CLI

刪除集合

下列delete-collection命令會刪除指定的集合。

```
aws rekognition delete-collection \
    --collection-id MyCollection
```

輸出：

```
{
    "StatusCode": 200
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的刪除集合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCollection。

delete-faces

以下程式碼範例顯示如何使用 delete-faces。

如需詳細資訊，請參閱從集合中刪除人臉。

AWS CLI

從集合中刪除臉部

下列delete-faces命令會從集合中刪除指定的臉部。

```
aws rekognition delete-faces \
    --collection-id MyCollection
    --face-ids '["0040279c-0178-436e-b70a-e61b074e96b0"]'
```

輸出：

```
{
    "DeletedFaces": [
        "0040279c-0178-436e-b70a-e61b074e96b0"
    ]
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的從集合刪除臉部。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFaces。

delete-stream-processor

以下程式碼範例顯示如何使用 delete-stream-processor。

AWS CLI

刪除串流處理器

下列delete-stream-processor命令會刪除指定的串流處理器。

```
aws rekognition delete-stream-processor \
    --name my-stream-processor
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的使用串流影片。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteStreamProcessor。

describe-collection

以下程式碼範例顯示如何使用 describe-collection。

如需詳細資訊，請參閱描述集合。

AWS CLI

描述集合

下列describe-collection範例顯示指定集合的詳細資訊。

```
aws rekognition describe-collection \
    --collection-id MyCollection
```

輸出：

```
{
    "FaceCount": 200,
    "CreationTimestamp": 1569444828.274,
    "CollectionARN": "arn:aws:rekognition:us-west-2:123456789012:collection/
MyCollection",
    "FaceModelVersion": "4.0"
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的描述集合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCollection。

describe-stream-processor

以下程式碼範例顯示如何使用 describe-stream-processor。

AWS CLI

取得串流處理器的相關資訊

下列describe-stream-processor命令會顯示指定串流處理器的詳細資訊。

```
aws rekognition describe-stream-processor \
    --name my-stream-processor
```

輸出：

```
{
    "Status": "STOPPED",
    "Name": "my-stream-processor",
    "LastUpdateTimestamp": 1532449292.712,
```

```
    "Settings": {
        "FaceSearch": {
            "FaceMatchThreshold": 80.0,
            "CollectionId": "my-collection"
        }
    },
    "RoleArn": "arn:aws:iam::123456789012:role/AmazonRekognitionDetectStream",
    "StreamProcessorArn": "arn:aws:rekognition:us-
west-2:123456789012:streamprocessor/my-stream-processpr",
    "Output": {
        "KinesisDataStream": {
            "Arn": "arn:aws:kinesis:us-west-2:123456789012:stream/
AmazonRekognitionRekStream"
        }
    },
    "Input": {
        "KinesisVideoStream": {
            "Arn": "arn:aws:kinesisvideo:us-west-2:123456789012:stream/
macwebcam/123456789012"
        }
    },
    "CreationTimestamp": 1532449292.712
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的使用串流影片。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStreamProcessor。

detect-faces

以下程式碼範例顯示如何使用 detect-faces。

如需詳細資訊，請參閱在映像中偵測人臉。

AWS CLI

偵測影像中的臉部

下列detect-faces命令會偵測存放在 Amazon S3 儲存貯體中指定映像中的臉部。

```
aws rekognition detect-faces \
    --image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"MyFriend.jpg"}}' \
    --attributes "ALL"
```

輸出：


```
{
    "FaceDetails": [
        {
            "Confidence": 100.0,
            "Eyeglasses": {
                "Confidence": 98.91107940673828,
                "Value": false
            },
            "Sunglasses": {
                "Confidence": 99.7966537475586,
                "Value": false
            },
            "Gender": {
                "Confidence": 99.56611633300781,
                "Value": "Male"
            },
            "Landmarks": [
                {
                    "Y": 0.26721030473709106,
                    "X": 0.6204193830490112,
                    "Type": "eyeLeft"
                },
                {
                    "Y": 0.26831310987472534,
                    "X": 0.6776827573776245,
                    "Type": "eyeRight"
                },
                {
                    "Y": 0.3514654338359833,
                    "X": 0.6241428852081299,
                    "Type": "mouthLeft"
                },
                {
                    "Y": 0.35258132219314575,
                    "X": 0.6713621020317078,
                    "Type": "mouthRight"
                },
                {
                    "Y": 0.3140771687030792,
                    "X": 0.6428444981575012,
                    "Type": "nose"
                },
```

```
                {
                    "Y": 0.24662546813488007,
                    "X": 0.6001564860343933,
                    "Type": "leftEyeBrowLeft"
                },
                {
                    "Y": 0.24326619505882263,
                    "X": 0.6303644776344299,
                    "Type": "leftEyeBrowRight"
                },
                {
                    "Y": 0.23818562924861908,
                    "X": 0.6146903038024902,
                    "Type": "leftEyeBrowUp"
                },
                {
                    "Y": 0.24373626708984375,
                    "X": 0.6640064716339111,
                    "Type": "rightEyeBrowLeft"
                },
                {
                    "Y": 0.24877218902111053,
                    "X": 0.7025929093360901,
                    "Type": "rightEyeBrowRight"
                },
                {
                    "Y": 0.23938551545143127,
                    "X": 0.6823262572288513,
                    "Type": "rightEyeBrowUp"
                },
                {
                    "Y": 0.265746533870697,
                    "X": 0.6112898588180542,
                    "Type": "leftEyeLeft"
                },
                {
                    "Y": 0.2676128149032593,
                    "X": 0.6317071914672852,
                    "Type": "leftEyeRight"
                },
                {
                    "Y": 0.262735515832901,
                    "X": 0.6201658248901367,
                    "Type": "leftEyeUp"
```

```
                },
                {
                    "Y": 0.27025148272514343,
                    "X": 0.6206279993057251,
                    "Type": "leftEyeDown"
                },
                {
                    "Y": 0.268223375082016,
                    "X": 0.6658390760421753,
                    "Type": "rightEyeLeft"
                },
                {
                    "Y": 0.2672517001628876,
                    "X": 0.687832236289978,
                    "Type": "rightEyeRight"
                },
                {
                    "Y": 0.26383838057518005,
                    "X": 0.6769183874130249,
                    "Type": "rightEyeUp"
                },
                {
                    "Y": 0.27138751745224,
                    "X": 0.676596462726593,
                    "Type": "rightEyeDown"
                },
                {
                    "Y": 0.32283174991607666,
                    "X": 0.6350004076957703,
                    "Type": "noseLeft"
                },
                {
                    "Y": 0.3219289481639862,
                    "X": 0.6567046642303467,
                    "Type": "noseRight"
                },
                {
                    "Y": 0.3420318365097046,
                    "X": 0.6450609564781189,
                    "Type": "mouthUp"
                },
                {
                    "Y": 0.3664324879646301,
                    "X": 0.6455618143081665,
```

```
                    "Type": "mouthDown"
                },
                {
                    "Y": 0.26721030473709106,
                    "X": 0.6204193830490112,
                    "Type": "leftPupil"
                },
                {
                    "Y": 0.26831310987472534,
                    "X": 0.6776827573776245,
                    "Type": "rightPupil"
                },
                {
                    "Y": 0.26343393325805664,
                    "X": 0.5946047306060791,
                    "Type": "upperJawlineLeft"
                },
                {
                    "Y": 0.3543180525302887,
                    "X": 0.6044883728027344,
                    "Type": "midJawlineLeft"
                },
                {
                    "Y": 0.4084877669811249,
                    "X": 0.6477024555206299,
                    "Type": "chinBottom"
                },
                {
                    "Y": 0.3562754988670349,
                    "X": 0.707981526851654,
                    "Type": "midJawlineRight"
                },
                {
                    "Y": 0.26580461859703064,
                    "X": 0.7234612107276917,
                    "Type": "upperJawlineRight"
                }
            ],
            "Pose": {
                "Yaw": -3.7351467609405518,
                "Roll": -0.10309021919965744,
                "Pitch": 0.8637830018997192
            },
            "Emotions": [
```

```
                {
                    "Confidence": 8.74203109741211,
                    "Type": "SURPRISED"
                },
                {
                    "Confidence": 2.501944065093994,
                    "Type": "ANGRY"
                },
                {
                    "Confidence": 0.7378743290901184,
                    "Type": "DISGUSTED"
                },
                {
                    "Confidence": 3.5296201705932617,
                    "Type": "HAPPY"
                },
                {
                    "Confidence": 1.7162904739379883,
                    "Type": "SAD"
                },
                {
                    "Confidence": 9.518536567687988,
                    "Type": "CONFUSED"
                },
                {
                    "Confidence": 0.45474427938461304,
                    "Type": "FEAR"
                },
                {
                    "Confidence": 72.79895782470703,
                    "Type": "CALM"
                }
            ],
            "AgeRange": {
                "High": 48,
                "Low": 32
            },
            "EyesOpen": {
                "Confidence": 98.93987274169922,
                "Value": true
            },
            "BoundingBox": {
                "Width": 0.12368916720151901,
                "Top": 0.16007372736930847,
```

```
                "Left": 0.5901257991790771,
                "Height": 0.25140416622161865
            },
            "Smile": {
                "Confidence": 93.4493179321289,
                "Value": false
            },
            "MouthOpen": {
                "Confidence": 90.53053283691406,
                "Value": false
            },
            "Quality": {
                "Sharpness": 95.51618957519531,
                "Brightness": 65.29893493652344
            },
            "Mustache": {
                "Confidence": 89.85221099853516,
                "Value": false
            },
            "Beard": {
                "Confidence": 86.1991195678711,
                "Value": true
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的在映像中偵測臉部。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetectFaces。

`detect-labels`

以下程式碼範例顯示如何使用 `detect-labels`。

如需詳細資訊，請參閱偵測映像中的標籤。

AWS CLI

偵測影像中的標籤

下列`detect-labels`範例會偵測存放在 Amazon S3 儲存貯體之映像中的場景和物件。

```
aws rekognition detect-labels \
```

```
    --image '{"S3Object":{"Bucket":"bucket","Name":"image"}}'
```

輸出：

```
{
    "Labels": [
        {
            "Instances": [],
            "Confidence": 99.15271759033203,
            "Parents": [
                {
                    "Name": "Vehicle"
                },
                {
                    "Name": "Transportation"
                }
            ],
            "Name": "Automobile"
        },
        {
            "Instances": [],
            "Confidence": 99.15271759033203,
            "Parents": [
                {
                    "Name": "Transportation"
                }
            ],
            "Name": "Vehicle"
        },
        {
            "Instances": [],
            "Confidence": 99.15271759033203,
            "Parents": [],
            "Name": "Transportation"
        },
        {
            "Instances": [
                {
                    "BoundingBox": {
                        "Width": 0.10616336017847061,
                        "Top": 0.5039216876029968,
                        "Left": 0.0037978808395564556,
                        "Height": 0.18528179824352264
```

```
            },
            "Confidence": 99.15271759033203
        },
        {
            "BoundingBox": {
                "Width": 0.2429988533258438,
                "Top": 0.5251884460449219,
                "Left": 0.7309805154800415,
                "Height": 0.21577216684818268
            },
            "Confidence": 99.1286392211914
        },
        {
            "BoundingBox": {
                "Width": 0.14233611524105072,
                "Top": 0.5333095788955688,
                "Left": 0.6494812965393066,
                "Height": 0.15528248250484467
            },
            "Confidence": 98.48368072509766
        },
        {
            "BoundingBox": {
                "Width": 0.11086395382881165,
                "Top": 0.5354844927787781,
                "Left": 0.10355594009160995,
                "Height": 0.10271988064050674
            },
            "Confidence": 96.45606231689453
        },
        {
            "BoundingBox": {
                "Width": 0.06254628300666809,
                "Top": 0.5573825240135193,
                "Left": 0.46083059906959534,
                "Height": 0.053911514580249786
            },
            "Confidence": 93.65448760986328
        },
        {
            "BoundingBox": {
                "Width": 0.10105438530445099,
                "Top": 0.534368634223938,
                "Left": 0.5743985772132874,
```

```
                "Height": 0.12226245552301407
            },
            "Confidence": 93.06217193603516
        },
        {
            "BoundingBox": {
                "Width": 0.056389667093753815,
                "Top": 0.5235804319381714,
                "Left": 0.9427769780158997,
                "Height": 0.17163699865341187
            },
            "Confidence": 92.6864013671875
        },
        {
            "BoundingBox": {
                "Width": 0.06003860384225845,
                "Top": 0.5441341400146484,
                "Left": 0.22409997880458832,
                "Height": 0.06737709045410156
            },
            "Confidence": 90.4227066040039
        },
        {
            "BoundingBox": {
                "Width": 0.02848697081208229,
                "Top": 0.5107086896896362,
                "Left": 0,
                "Height": 0.19150497019290924
            },
            "Confidence": 86.65286254882812
        },
        {
            "BoundingBox": {
                "Width": 0.04067881405353546,
                "Top": 0.5566273927688599,
                "Left": 0.316415935754776,
                "Height": 0.03428703173995018
            },
            "Confidence": 85.36471557617188
        },
        {
            "BoundingBox": {
                "Width": 0.043411049991846085,
                "Top": 0.5394920110702515,
```

```
                    "Left": 0.18293385207653046,
                    "Height": 0.0893595889210701
                },
                "Confidence": 82.21705627441406
            },
            {
                "BoundingBox": {
                    "Width": 0.031183116137981415,
                    "Top": 0.5579366683959961,
                    "Left": 0.2853088080883026,
                    "Height": 0.03989990055561066
                },
                "Confidence": 81.0157470703125
            },
            {
                "BoundingBox": {
                    "Width": 0.031113790348172188,
                    "Top": 0.5504819750785828,
                    "Left": 0.2580395042896271,
                    "Height": 0.056484755128622055
                },
                "Confidence": 56.13441467285156
            },
            {
                "BoundingBox": {
                    "Width": 0.08586374670267105,
                    "Top": 0.5438792705535889,
                    "Left": 0.5128012895584106,
                    "Height": 0.08550430089235306
                },
                "Confidence": 52.37760925292969
            }
        ],
        "Confidence": 99.15271759033203,
        "Parents": [
            {
                "Name": "Vehicle"
            },
            {
                "Name": "Transportation"
            }
        ],
        "Name": "Car"
    },
```

```
        {
            "Instances": [],
            "Confidence": 98.9914321899414,
            "Parents": [],
            "Name": "Human"
        },
        {
            "Instances": [
                {
                    "BoundingBox": {
                        "Width": 0.19360728561878204,
                        "Top": 0.35072067379951477,
                        "Left": 0.43734854459762573,
                        "Height": 0.2742200493812561
                    },
                    "Confidence": 98.9914321899414
                },
                {
                    "BoundingBox": {
                        "Width": 0.03801717236638069,
                        "Top": 0.5010883808135986,
                        "Left": 0.9155802130699158,
                        "Height": 0.06597328186035156
                    },
                    "Confidence": 85.02790832519531
                }
            ],
            "Confidence": 98.9914321899414,
            "Parents": [],
            "Name": "Person"
        },
        {
            "Instances": [],
            "Confidence": 93.24951934814453,
            "Parents": [],
            "Name": "Machine"
        },
        {
            "Instances": [
                {
                    "BoundingBox": {
                        "Width": 0.03561960905790329,
                        "Top": 0.6468243598937988,
                        "Left": 0.7850857377052307,
```

```
                "Height": 0.08878646790981293
            },
            "Confidence": 93.24951934814453
        },
        {
            "BoundingBox": {
                "Width": 0.02217046171426773,
                "Top": 0.6149078607559204,
                "Left": 0.04757237061858177,
                "Height": 0.07136218994855881
            },
            "Confidence": 91.5025863647461
        },
        {
            "BoundingBox": {
                "Width": 0.016197510063648224,
                "Top": 0.6274210214614868,
                "Left": 0.6472989320755005,
                "Height": 0.04955997318029404
            },
            "Confidence": 85.14686584472656
        },
        {
            "BoundingBox": {
                "Width": 0.020207518711686134,
                "Top": 0.6348286867141724,
                "Left": 0.7295016646385193,
                "Height": 0.07059963047504425
            },
            "Confidence": 83.34547424316406
        },
        {
            "BoundingBox": {
                "Width": 0.020280985161662102,
                "Top": 0.6171894669532776,
                "Left": 0.08744934946298599,
                "Height": 0.05297485366463661
            },
            "Confidence": 79.9981460571289
        },
        {
            "BoundingBox": {
                "Width": 0.018318990245461464,
                "Top": 0.623889148235321,
```

```
                    "Left": 0.6836880445480347,
                    "Height": 0.06730121374130249
                },
                "Confidence": 78.87144470214844
            },
            {
                "BoundingBox": {
                    "Width": 0.021310249343514442,
                    "Top": 0.6167286038398743,
                    "Left": 0.004064912907779217,
                    "Height": 0.08317798376083374
                },
                "Confidence": 75.89361572265625
            },
            {
                "BoundingBox": {
                    "Width": 0.03604431077837944,
                    "Top": 0.7030032277107239,
                    "Left": 0.9254803657531738,
                    "Height": 0.04569442570209503
                },
                "Confidence": 64.402587890625
            },
            {
                "BoundingBox": {
                    "Width": 0.009834849275648594,
                    "Top": 0.5821820497512817,
                    "Left": 0.28094568848609924,
                    "Height": 0.01964157074689865
                },
                "Confidence": 62.79907989501953
            },
            {
                "BoundingBox": {
                    "Width": 0.01475677452981472,
                    "Top": 0.6137543320655823,
                    "Left": 0.5950819253921509,
                    "Height": 0.039063986390829086
                },
                "Confidence": 59.40483474731445
            }
        ],
        "Confidence": 93.24951934814453,
        "Parents": [
```

```
                {
                    "Name": "Machine"
                }
            ],
            "Name": "Wheel"
        },
        {
            "Instances": [],
            "Confidence": 92.61514282226562,
            "Parents": [],
            "Name": "Road"
        },
        {
            "Instances": [],
            "Confidence": 92.37877655029297,
            "Parents": [
                {
                    "Name": "Person"
                }
            ],
            "Name": "Sport"
        },
        {
            "Instances": [],
            "Confidence": 92.37877655029297,
            "Parents": [
                {
                    "Name": "Person"
                }
            ],
            "Name": "Sports"
        },
        {
            "Instances": [
                {
                    "BoundingBox": {
                        "Width": 0.12326609343290329,
                        "Top": 0.6332163214683533,
                        "Left": 0.44815489649772644,
                        "Height": 0.058117982000112534
                    },
                    "Confidence": 92.37877655029297
                }
            ],
```

```
            "Confidence": 92.37877655029297,
            "Parents": [
                {
                    "Name": "Person"
                },
                {
                    "Name": "Sport"
                }
            ],
            "Name": "Skateboard"
        },
        {
            "Instances": [],
            "Confidence": 90.62931060791016,
            "Parents": [
                {
                    "Name": "Person"
                }
            ],
            "Name": "Pedestrian"
        },
        {
            "Instances": [],
            "Confidence": 88.81334686279297,
            "Parents": [],
            "Name": "Asphalt"
        },
        {
            "Instances": [],
            "Confidence": 88.81334686279297,
            "Parents": [],
            "Name": "Tarmac"
        },
        {
            "Instances": [],
            "Confidence": 88.23201751708984,
            "Parents": [],
            "Name": "Path"
        },
        {
            "Instances": [],
            "Confidence": 80.26520538330078,
            "Parents": [],
            "Name": "Urban"
```

```
        },
        {
            "Instances": [],
            "Confidence": 80.26520538330078,
            "Parents": [
                {
                    "Name": "Building"
                },
                {
                    "Name": "Urban"
                }
            ],
            "Name": "Town"
        },
        {
            "Instances": [],
            "Confidence": 80.26520538330078,
            "Parents": [],
            "Name": "Building"
        },
        {
            "Instances": [],
            "Confidence": 80.26520538330078,
            "Parents": [
                {
                    "Name": "Building"
                },
                {
                    "Name": "Urban"
                }
            ],
            "Name": "City"
        },
        {
            "Instances": [],
            "Confidence": 78.37934875488281,
            "Parents": [
                {
                    "Name": "Car"
                },
                {
                    "Name": "Vehicle"
                },
                {
```

```
                    "Name": "Transportation"
                }
            ],
            "Name": "Parking Lot"
        },
        {
            "Instances": [],
            "Confidence": 78.37934875488281,
            "Parents": [
                {
                    "Name": "Car"
                },
                {
                    "Name": "Vehicle"
                },
                {
                    "Name": "Transportation"
                }
            ],
            "Name": "Parking"
        },
        {
            "Instances": [],
            "Confidence": 74.37590026855469,
            "Parents": [
                {
                    "Name": "Building"
                },
                {
                    "Name": "Urban"
                },
                {
                    "Name": "City"
                }
            ],
            "Name": "Downtown"
        },
        {
            "Instances": [],
            "Confidence": 69.84622955322266,
            "Parents": [
                {
                    "Name": "Road"
                }
```

```
            ],
            "Name": "Intersection"
        },
        {
            "Instances": [],
            "Confidence": 57.68518829345703,
            "Parents": [
                {
                    "Name": "Sports Car"
                },
                {
                    "Name": "Car"
                },
                {
                    "Name": "Vehicle"
                },
                {
                    "Name": "Transportation"
                }
            ],
            "Name": "Coupe"
        },
        {
            "Instances": [],
            "Confidence": 57.68518829345703,
            "Parents": [
                {
                    "Name": "Car"
                },
                {
                    "Name": "Vehicle"
                },
                {
                    "Name": "Transportation"
                }
            ],
            "Name": "Sports Car"
        },
        {
            "Instances": [],
            "Confidence": 56.59492111206055,
            "Parents": [
                {
                    "Name": "Path"
```

```
                }
            ],
            "Name": "Sidewalk"
        },
        {
            "Instances": [],
            "Confidence": 56.59492111206055,
            "Parents": [
                {
                    "Name": "Path"
                }
            ],
            "Name": "Pavement"
        },
        {
            "Instances": [],
            "Confidence": 55.58770751953125,
            "Parents": [
                {
                    "Name": "Building"
                },
                {
                    "Name": "Urban"
                }
            ],
            "Name": "Neighborhood"
        }
    ],
    "LabelModelVersion": "2.0"
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的在映像中偵測標籤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetectLabels。

`detect-moderation-labels`

以下程式碼範例顯示如何使用 `detect-moderation-labels`。

如需詳細資訊，請參閱偵測不適合的映像。

AWS CLI

偵測影像中的不安全內容

下列detect-moderation-labels命令會偵測存放在 Amazon S3 儲存貯體之指定映像中的不安全內容。

```
aws rekognition detect-moderation-labels \
    --image "S3Object={Bucket=MyImageS3Bucket,Name=gun.jpg}"
```

輸出：

```
{
    "ModerationModelVersion": "3.0",
    "ModerationLabels": [
        {
            "Confidence": 97.29618072509766,
            "ParentName": "Violence",
            "Name": "Weapon Violence"
        },
        {
            "Confidence": 97.29618072509766,
            "ParentName": "",
            "Name": "Violence"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的偵測不安全的影像。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetectModerationLabels。

detect-text

以下程式碼範例顯示如何使用 detect-text。

如需更多資訊，請參閱偵測映像中的文字。

AWS CLI

偵測影像中的文字

下列detect-text命令會偵測指定映像中的文字。

```
aws rekognition detect-text \
    --image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"ExamplePicture.jpg"}}'
```

輸出：


```
{
    "TextDetections": [
        {
            "Geometry": {
                "BoundingBox": {
                    "Width": 0.24624845385551453,
                    "Top": 0.28288066387176514,
                    "Left": 0.391388863325119,
                    "Height": 0.022687450051307678
                },
                "Polygon": [
                    {
                        "Y": 0.28288066387176514,
                        "X": 0.391388863325119
                    },
                    {
                        "Y": 0.2826388478279114,
                        "X": 0.6376373171806335
                    },
                    {
                        "Y": 0.30532628297805786,
                        "X": 0.637677013874054
                    },
                    {
                        "Y": 0.305568128824234,
                        "X": 0.39142853021621704
                    }
                ]
            },
            "Confidence": 94.35709381103516,
            "DetectedText": "ESTD 1882",
            "Type": "LINE",
            "Id": 0
        },
        {
            "Geometry": {
                "BoundingBox": {
                    "Width": 0.33933889865875244,
                    "Top": 0.32603850960731506,
                    "Left": 0.34534579515457153,
                    "Height": 0.07126858830451965
                },
```

```
                "Polygon": [
                    {
                        "Y": 0.32603850960731506,
                        "X": 0.34534579515457153
                    },
                    {
                        "Y": 0.32633158564567566,
                        "X": 0.684684693813324
                    },
                    {
                        "Y": 0.3976001739501953,
                        "X": 0.684575080871582
                    },
                    {
                        "Y": 0.3973070979118347,
                        "X": 0.345236212015152
                    }
                ]
            },
            "Confidence": 99.95779418945312,
            "DetectedText": "BRAINS",
            "Type": "LINE",
            "Id": 1
        },
        {
            "Confidence": 97.22098541259766,
            "Geometry": {
                "BoundingBox": {
                    "Width": 0.061079490929841995,
                    "Top": 0.2843210697174072,
                    "Left": 0.391391396522522,
                    "Height": 0.021029088646173477
                },
                "Polygon": [
                    {
                        "Y": 0.2843210697174072,
                        "X": 0.391391396522522
                    },
                    {
                        "Y": 0.2828207015991211,
                        "X": 0.4524524509906769
                    },
                    {
                        "Y": 0.3038259446620941,
```

```
                    "X": 0.4534534513950348
                },
                {
                    "Y": 0.30532634258270264,
                    "X": 0.3923923969268799
                }
            ]
        },
        "DetectedText": "ESTD",
        "ParentId": 0,
        "Type": "WORD",
        "Id": 2
    },
    {
        "Confidence": 91.49320983886719,
        "Geometry": {
            "BoundingBox": {
                "Width": 0.07007007300853729,
                "Top": 0.2828207015991211,
                "Left": 0.5675675868988037,
                "Height": 0.02250562608242035
            },
            "Polygon": [
                {
                    "Y": 0.2828207015991211,
                    "X": 0.5675675868988037
                },
                {
                    "Y": 0.2828207015991211,
                    "X": 0.6376376152038574
                },
                {
                    "Y": 0.30532634258270264,
                    "X": 0.6376376152038574
                },
                {
                    "Y": 0.30532634258270264,
                    "X": 0.5675675868988037
                }
            ]
        },
        "DetectedText": "1882",
        "ParentId": 0,
        "Type": "WORD",
```

```
            "Id": 3
        },
        {
            "Confidence": 99.95779418945312,
            "Geometry": {
                "BoundingBox": {
                    "Width": 0.33933934569358826,
                    "Top": 0.32633158564567566,
                    "Left": 0.3453453481197357,
                    "Height": 0.07127484679222107
                },
                "Polygon": [
                    {
                        "Y": 0.32633158564567566,
                        "X": 0.3453453481197357
                    },
                    {
                        "Y": 0.32633158564567566,
                        "X": 0.684684693813324
                    },
                    {
                        "Y": 0.39759939908981323,
                        "X": 0.6836836934089661
                    },
                    {
                        "Y": 0.39684921503067017,
                        "X": 0.3453453481197357
                    }
                ]
            },
            "DetectedText": "BRAINS",
            "ParentId": 1,
            "Type": "WORD",
            "Id": 4
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetectText。

disassociate-faces

以下程式碼範例顯示如何使用 disassociate-faces。

AWS CLI

```
aws rekognition disassociate-faces --face-ids list-of-face-ids
  --user-id user-id --collection-id collection-name --region region-name
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateFaces。

get-celebrity-info

以下程式碼範例顯示如何使用 get-celebrity-info。

AWS CLI

取得名人的相關資訊

下列get-celebrity-info命令會顯示指定名人的相關資訊。id 參數來自先前對 的呼叫recognize-celebrities。

```
aws rekognition get-celebrity-info --id nnnnnnn
```

輸出：

```
{
    "Name": "Celeb A",
    "Urls": [
        "www.imdb.com/name/aaaaaaaaa"
    ]
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的取得名人的相關資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCelebrityInfo。

get-celebrity-recognition

以下程式碼範例顯示如何使用 get-celebrity-recognition。

AWS CLI

取得名人辨識操作的結果

下列get-celebrity-recognition命令會顯示您之前透過呼叫 開始的名人辨識操作的結果start-celebrity-recognition。

```
aws rekognition get-celebrity-recognition  \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

輸出：

```
{
    "NextToken": "3D01ClxlCiT31VsRDkAO3IybLb/h5AtDWSGuhYi
+N1FIJwwPtAkuKzDhL2rV3GcwmNt77+12",
    "Celebrities": [
        {
            "Timestamp": 0,
            "Celebrity": {
                "Confidence": 96.0,
                "Face": {
                    "BoundingBox": {
                        "Width": 0.70333331823349,
                        "Top": 0.16750000417232513,
                        "Left": 0.19555555284023285,
                        "Height": 0.3956249952316284
                    },
                    "Landmarks": [
                        {
                            "Y": 0.31031012535095215,
                            "X": 0.441436767578125,
                            "Type": "eyeLeft"
                        },
                        {
                            "Y": 0.3081788718700409,
                            "X": 0.6437258720397949,
                            "Type": "eyeRight"
                        },
                        {
                            "Y": 0.39542075991630554,
                            "X": 0.5572493076324463,
                            "Type": "nose"
                        },
                        {
                            "Y": 0.4597957134246826,
                            "X": 0.4579732120037079,
                            "Type": "mouthLeft"
```

```
                    },
                    {
                        "Y": 0.45688048005104065,
                        "X": 0.6349081993103027,
                        "Type": "mouthRight"
                    }
                ],
                "Pose": {
                    "Yaw": 8.943398475646973,
                    "Roll": -2.0309247970581055,
                    "Pitch": -0.5674862861633301
                },
                "Quality": {
                    "Sharpness": 99.40211486816406,
                    "Brightness": 89.47132110595703
                },
                "Confidence": 99.99861145019531
            },
            "Name": "CelebrityA",
            "Urls": [
                "www.imdb.com/name/111111111"
            ],
            "Id": "nnnnnn"
        }
    },
    {
        "Timestamp": 467,
        "Celebrity": {
            "Confidence": 99.0,
            "Face": {
                "BoundingBox": {
                    "Width": 0.6877777576446533,
                    "Top": 0.18437500298023224,
                    "Left": 0.20555555820465088,
                    "Height": 0.3868750035762787
                },
                "Landmarks": [
                    {
                        "Y": 0.31895750761032104,
                        "X": 0.4411413371562958,
                        "Type": "eyeLeft"
                    },
                    {
                        "Y": 0.3140959143638611,
```

```
                        "X": 0.6523157954216003,
                        "Type": "eyeRight"
                    },
                    {
                        "Y": 0.4016456604003906,
                        "X": 0.5682755708694458,
                        "Type": "nose"
                    },
                    {
                        "Y": 0.46894142031669617,
                        "X": 0.4597797095775604,
                        "Type": "mouthLeft"
                    },
                    {
                        "Y": 0.46971091628074646,
                        "X": 0.6286435127258301,
                        "Type": "mouthRight"
                    }
                ],
                "Pose": {
                    "Yaw": 10.433465957641602,
                    "Roll": -3.347442388534546,
                    "Pitch": 1.3709543943405151
                },
                "Quality": {
                    "Sharpness": 99.5531005859375,
                    "Brightness": 88.5764389038086
                },
                "Confidence": 99.99148559570312
            },
            "Name": "Jane Celebrity",
            "Urls": [
                "www.imdb.com/name/111111111"
            ],
            "Id": "nnnnnn"
        }
    }
],
"JobStatus": "SUCCEEDED",
"VideoMetadata": {
    "Format": "QuickTime / MOV",
    "FrameRate": 29.978118896484375,
    "Codec": "h264",
    "DurationMillis": 4570,
```

```
        "FrameHeight": 1920,
        "FrameWidth": 1080
    }
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的在儲存的影片中辨識名人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCelebrityRecognition。

get-content-moderation

以下程式碼範例顯示如何使用 get-content-moderation。

AWS CLI

取得不安全內容操作的結果

下列get-content-moderation命令會顯示您先前透過呼叫 啟動的不安全內容操作的結果start-content-moderation。

```
aws rekognition get-content-moderation \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

輸出：

```
{
    "NextToken": "dlhcKMHMzpCBGFukz6IO3JMcWiJAamCVhXHt3r6b4b5Tfbyw3q7o+Jeezt
+ZpgfOnW9FCCgQ",
    "ModerationLabels": [
        {
            "Timestamp": 0,
            "ModerationLabel": {
                "Confidence": 97.39583587646484,
                "ParentName": "",
                "Name": "Violence"
            }
        },
        {
            "Timestamp": 0,
            "ModerationLabel": {
                "Confidence": 97.39583587646484,
                "ParentName": "Violence",
                "Name": "Weapon Violence"
```

```
            }
        }
    ],
    "JobStatus": "SUCCEEDED",
    "VideoMetadata": {
        "Format": "QuickTime / MOV",
        "FrameRate": 29.97515869140625,
        "Codec": "h264",
        "DurationMillis": 6039,
        "FrameHeight": 1920,
        "FrameWidth": 1080
    }
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的偵測不安全的存放影片。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetContentModeration。

get-face-detection

以下程式碼範例顯示如何使用 get-face-detection。

AWS CLI

取得臉部偵測操作的結果

下列get-face-detection命令會顯示您先前透過呼叫 開始的臉部偵測操作的結果start-face-detection。

```
aws rekognition get-face-detection \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

輸出：

```
{
    "Faces": [
        {
            "Timestamp": 467,
            "Face": {
                "BoundingBox": {
                    "Width": 0.1560753583908081,
                    "Top": 0.13555361330509186,
                    "Left": -0.0952017530798912,
```

```
                    "Height": 0.6934483051300049
                },
                "Landmarks": [
                    {
                        "Y": 0.4013825058937073,
                        "X": -0.041750285774469376,
                        "Type": "eyeLeft"
                    },
                    {
                        "Y": 0.41695496439933777,
                        "X": 0.027979329228401184,
                        "Type": "eyeRight"
                    },
                    {
                        "Y": 0.6375303268432617,
                        "X": -0.04034662991762161,
                        "Type": "mouthLeft"
                    },
                    {
                        "Y": 0.6497718691825867,
                        "X": 0.013960429467260838,
                        "Type": "mouthRight"
                    },
                    {
                        "Y": 0.5238034129142761,
                        "X": 0.008022055961191654,
                        "Type": "nose"
                    }
                ],
                "Pose": {
                    "Yaw": -58.07863998413086,
                    "Roll": 1.9384294748306274,
                    "Pitch": -24.66305160522461
                },
                "Quality": {
                    "Sharpness": 83.14741516113281,
                    "Brightness": 25.75942611694336
                },
                "Confidence": 87.7622299194336
            }
        },
        {
            "Timestamp": 967,
            "Face": {
```

```
            "BoundingBox": {
                "Width": 0.28559377789497375,
                "Top": 0.19436298310756683,
                "Left": 0.024553587660193443,
                "Height": 0.7216082215309143
            },
            "Landmarks": [
                {
                    "Y": 0.4650231599807739,
                    "X": 0.16269078850746155,
                    "Type": "eyeLeft"
                },
                {
                    "Y": 0.4843238294124603,
                    "X": 0.2782580852508545,
                    "Type": "eyeRight"
                },
                {
                    "Y": 0.71530681848526,
                    "X": 0.1741468608379364,
                    "Type": "mouthLeft"
                },
                {
                    "Y": 0.7310671210289001,
                    "X": 0.26857468485832214,
                    "Type": "mouthRight"
                },
                {
                    "Y": 0.582602322101593,
                    "X": 0.2566150426864624,
                    "Type": "nose"
                }
            ],
            "Pose": {
                "Yaw": 11.487052917480469,
                "Roll": 5.074230670928955,
                "Pitch": 15.396159172058105
            },
            "Quality": {
                "Sharpness": 73.32209777832031,
                "Brightness": 54.96497344970703
            },
            "Confidence": 99.99998474121094
        }
```

```
        }
    ],
    "NextToken":
 "OzL223pDKy9116O/02KXRqFIEAwxjy4PkgYcm3hSo0rdysbXg5Ex0eFgTGEj0ADEac6S037U",
    "JobStatus": "SUCCEEDED",
    "VideoMetadata": {
        "Format": "QuickTime / MOV",
        "FrameRate": 29.970617294311523,
        "Codec": "h264",
        "DurationMillis": 6806,
        "FrameHeight": 1080,
        "FrameWidth": 1920
    }
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的在儲存的影片中偵測臉部。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFaceDetection。

get-face-search

以下程式碼範例顯示如何使用 get-face-search。

AWS CLI

取得臉部搜尋操作的結果

下列get-face-search命令會顯示您先前透過呼叫 開始的臉部搜尋操作的結果start-face-search。

```
aws rekognition get-face-search  \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

輸出：

```
{
    "Persons": [
        {
            "Timestamp": 467,
            "FaceMatches": [],
            "Person": {
                "Index": 0,
                "Face": {
```

```
                    "BoundingBox": {
                        "Width": 0.1560753583908081,
                        "Top": 0.13555361330509186,
                        "Left": -0.0952017530798912,
                        "Height": 0.6934483051300049
                    },
                    "Landmarks": [
                        {
                            "Y": 0.4013825058937073,
                            "X": -0.041750285774469376,
                            "Type": "eyeLeft"
                        },
                        {
                            "Y": 0.41695496439933777,
                            "X": 0.027979329228401184,
                            "Type": "eyeRight"
                        },
                        {
                            "Y": 0.6375303268432617,
                            "X": -0.04034662991762161,
                            "Type": "mouthLeft"
                        },
                        {
                            "Y": 0.6497718691825867,
                            "X": 0.013960429467260838,
                            "Type": "mouthRight"
                        },
                        {
                            "Y": 0.5238034129142761,
                            "X": 0.008022055961191654,
                            "Type": "nose"
                        }
                    ],
                    "Pose": {
                        "Yaw": -58.07863998413086,
                        "Roll": 1.9384294748306274,
                        "Pitch": -24.66305160522461
                    },
                    "Quality": {
                        "Sharpness": 83.14741516113281,
                        "Brightness": 25.75942611694336
                    },
                    "Confidence": 87.7622299194336
                }
```

```
            }
        },
        {
            "Timestamp": 967,
            "FaceMatches": [
                {
                    "Face": {
                        "BoundingBox": {
                            "Width": 0.12368900328874588,
                            "Top": 0.16007399559020996,
                            "Left": 0.5901259779930115,
                            "Height": 0.2514039874076843
                        },
                        "FaceId": "056a95fa-2060-4159-9cab-7ed4daa030fa",
                        "ExternalImageId": "image3.jpg",
                        "Confidence": 100.0,
                        "ImageId": "08f8a078-8929-37fd-8e8f-aadf690e8232"
                    },
                    "Similarity": 98.44476318359375
                }
            ],
            "Person": {
                "Index": 1,
                "Face": {
                    "BoundingBox": {
                        "Width": 0.28559377789497375,
                        "Top": 0.19436298310756683,
                        "Left": 0.024553587660193443,
                        "Height": 0.7216082215309143
                    },
                    "Landmarks": [
                        {
                            "Y": 0.4650231599807739,
                            "X": 0.16269078850746155,
                            "Type": "eyeLeft"
                        },
                        {
                            "Y": 0.4843238294124603,
                            "X": 0.2782580852508545,
                            "Type": "eyeRight"
                        },
                        {
                            "Y": 0.71530681848526,
                            "X": 0.1741468608379364,
```

```
                            "Type": "mouthLeft"
                        },
                        {
                            "Y": 0.7310671210289001,
                            "X": 0.26857468485832214,
                            "Type": "mouthRight"
                        },
                        {
                            "Y": 0.582602322101593,
                            "X": 0.2566150426864624,
                            "Type": "nose"
                        }
                    ],
                    "Pose": {
                        "Yaw": 11.487052917480469,
                        "Roll": 5.074230670928955,
                        "Pitch": 15.396159172058105
                    },
                    "Quality": {
                        "Sharpness": 73.32209777832031,
                        "Brightness": 54.96497344970703
                    },
                    "Confidence": 99.99998474121094
                }
            }
        }
    ],
    "NextToken": "5bkgcezyuaqhtWk3C8OTW6cjRghrwV9XDMivm5B3MXm+Lv6G+L+GejyFHPhoNa/
ldXIC4c/d",
    "JobStatus": "SUCCEEDED",
    "VideoMetadata": {
        "Format": "QuickTime / MOV",
        "FrameRate": 29.970617294311523,
        "Codec": "h264",
        "DurationMillis": 6806,
        "FrameHeight": 1080,
        "FrameWidth": 1920
    }
}
```


如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的搜尋儲存的臉部影片。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFaceSearch。

get-label-detection

以下程式碼範例顯示如何使用 get-label-detection。

AWS CLI

取得物件和場景偵測操作的結果

下列get-label-detection命令會顯示您先前透過呼叫 啟動的物件和場景偵測操作的結果start-label-detection。

```
aws rekognition get-label-detection  \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

輸出：

```
{
    "Labels": [
        {
            "Timestamp": 0,
            "Label": {
                "Instances": [],
                "Confidence": 50.19071578979492,
                "Parents": [
                    {
                        "Name": "Person"
                    },
                    {
                        "Name": "Crowd"
                    }
                ],
                "Name": "Audience"
            }
        },
        {
            "Timestamp": 0,
            "Label": {
                "Instances": [],
                "Confidence": 55.74115753173828,
                "Parents": [
                    {
                        "Name": "Room"
                    },
```

```
                    {
                        "Name": "Indoors"
                    },
                    {
                        "Name": "School"
                    }
                ],
                "Name": "Classroom"
            }
        }
    ],
    "JobStatus": "SUCCEEDED",
    "LabelModelVersion": "2.0",
    "VideoMetadata": {
        "Format": "QuickTime / MOV",
        "FrameRate": 29.970617294311523,
        "Codec": "h264",
        "DurationMillis": 6806,
        "FrameHeight": 1080,
        "FrameWidth": 1920
    },
    "NextToken": "BMugzAi4L72IERzQdbpyMQuEFBsjlo5W0Yx3mfG+sR9mm98E1/
CpObenspRfs/5FBQFs4X7G"
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的偵測影片中的標籤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLabelDetection。

get-person-tracking

以下程式碼範例顯示如何使用 get-person-tracking。

AWS CLI

取得人員路徑操作的結果

下列get-person-tracking命令會顯示您先前透過呼叫 啟動的人員路徑操作的結果start-person-tracking。

```
aws rekognition get-person-tracking  \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

輸出：


```
{
    "Persons": [
        {
            "Timestamp": 500,
            "Person": {
                "BoundingBox": {
                    "Width": 0.4151041805744171,
                    "Top": 0.07870370149612427,
                    "Left": 0.0,
                    "Height": 0.9212962985038757
                },
                "Index": 0
            }
        },
        {
            "Timestamp": 567,
            "Person": {
                "BoundingBox": {
                    "Width": 0.4755208194255829,
                    "Top": 0.07777778059244156,
                    "Left": 0.0,
                    "Height": 0.9194444417953491
                },
                "Index": 0
            }
        }
    ],
    "NextToken": "D/vRIYNyhG79ugdta3f+8cRg9oSRo
+HigGOuxRiYpTn0ExnqTi1CJektVAc4HrAXDv25eHYk",
    "JobStatus": "SUCCEEDED",
    "VideoMetadata": {
        "Format": "QuickTime / MOV",
        "FrameRate": 29.970617294311523,
        "Codec": "h264",
        "DurationMillis": 6806,
        "FrameHeight": 1080,
        "FrameWidth": 1920
    }
}
```


如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的人員路徑。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPersonTracking。

index-faces

以下程式碼範例顯示如何使用 index-faces。

如需詳細資訊，請參閱將人臉新增至集合。

AWS CLI

將臉部新增至集合

下列index-faces命令會將影像中找到的臉部新增至指定的集合。

```
aws rekognition index-faces \
    --image '{"S3Object":{"Bucket":"MyVideoS3Bucket","Name":"MyPicture.jpg"}}' \
    --collection-id MyCollection \
    --max-faces 1 \
    --quality-filter "AUTO" \
    --detection-attributes "ALL" \
    --external-image-id "MyPicture.jpg"
```

輸出：

```
{
    "FaceRecords": [
        {
            "FaceDetail": {
                "Confidence": 99.993408203125,
                "Eyeglasses": {
                    "Confidence": 99.11750030517578,
                    "Value": false
                },
                "Sunglasses": {
                    "Confidence": 99.98249053955078,
                    "Value": false
                },
                "Gender": {
                    "Confidence": 99.92769622802734,
                    "Value": "Male"
                },
                "Landmarks": [
                    {
```

```
                    "Y": 0.26750367879867554,
                    "X": 0.6202793717384338,
                    "Type": "eyeLeft"
                },
                {
                    "Y": 0.26642778515815735,
                    "X": 0.6787431836128235,
                    "Type": "eyeRight"
                },
                {
                    "Y": 0.31361380219459534,
                    "X": 0.6421601176261902,
                    "Type": "nose"
                },
                {
                    "Y": 0.3495299220085144,
                    "X": 0.6216195225715637,
                    "Type": "mouthLeft"
                },
                {
                    "Y": 0.35194727778434753,
                    "X": 0.669899046421051,
                    "Type": "mouthRight"
                },
                {
                    "Y": 0.26844894886016846,
                    "X": 0.6210268139839172,
                    "Type": "leftPupil"
                },
                {
                    "Y": 0.26707562804222107,
                    "X": 0.6817160844802856,
                    "Type": "rightPupil"
                },
                {
                    "Y": 0.24834522604942322,
                    "X": 0.6018546223640442,
                    "Type": "leftEyeBrowLeft"
                },
                {
                    "Y": 0.24397172033786774,
                    "X": 0.6172008514404297,
                    "Type": "leftEyeBrowUp"
                },
```

```
                    {
                        "Y": 0.24677404761314392,
                        "X": 0.6339119076728821,
                        "Type": "leftEyeBrowRight"
                    },
                    {
                        "Y": 0.24582654237747192,
                        "X": 0.6619398593902588,
                        "Type": "rightEyeBrowLeft"
                    },
                    {
                        "Y": 0.23973053693771362,
                        "X": 0.6804757118225098,
                        "Type": "rightEyeBrowUp"
                    },
                    {
                        "Y": 0.24441994726657867,
                        "X": 0.6978968977928162,
                        "Type": "rightEyeBrowRight"
                    },
                    {
                        "Y": 0.2695908546447754,
                        "X": 0.6085202693939209,
                        "Type": "leftEyeLeft"
                    },
                    {
                        "Y": 0.26716896891593933,
                        "X": 0.6315826177597046,
                        "Type": "leftEyeRight"
                    },
                    {
                        "Y": 0.26289820671081543,
                        "X": 0.6202316880226135,
                        "Type": "leftEyeUp"
                    },
                    {
                        "Y": 0.27123287320137024,
                        "X": 0.6205548048019409,
                        "Type": "leftEyeDown"
                    },
                    {
                        "Y": 0.2668408751487732,
                        "X": 0.6663622260093689,
                        "Type": "rightEyeLeft"
```

```
            },
            {
                "Y": 0.26741549372673035,
                "X": 0.6910083889961243,
                "Type": "rightEyeRight"
            },
            {
                "Y": 0.2614026665687561,
                "X": 0.6785826086997986,
                "Type": "rightEyeUp"
            },
            {
                "Y": 0.27075251936912537,
                "X": 0.6789616942405701,
                "Type": "rightEyeDown"
            },
            {
                "Y": 0.3211299479007721,
                "X": 0.6324167847633362,
                "Type": "noseLeft"
            },
            {
                "Y": 0.32276326417922974,
                "X": 0.6558475494384766,
                "Type": "noseRight"
            },
            {
                "Y": 0.34385165572166443,
                "X": 0.6444970965385437,
                "Type": "mouthUp"
            },
            {
                "Y": 0.3671635091304779,
                "X": 0.6459195017814636,
                "Type": "mouthDown"
            }
        ],
        "Pose": {
            "Yaw": -9.54541015625,
            "Roll": -0.5709401965141296,
            "Pitch": 0.6045494675636292
        },
        "Emotions": [
            {
```

```
                "Confidence": 39.90074157714844,
                "Type": "HAPPY"
            },
            {
                "Confidence": 23.38753890991211,
                "Type": "CALM"
            },
            {
                "Confidence": 5.840933322906494,
                "Type": "CONFUSED"
            }
        ],
        "AgeRange": {
            "High": 63,
            "Low": 45
        },
        "EyesOpen": {
            "Confidence": 99.80887603759766,
            "Value": true
        },
        "BoundingBox": {
            "Width": 0.18562500178813934,
            "Top": 0.1618015021085739,
            "Left": 0.5575000047683716,
            "Height": 0.24770642817020416
        },
        "Smile": {
            "Confidence": 99.69740295410156,
            "Value": false
        },
        "MouthOpen": {
            "Confidence": 99.97393798828125,
            "Value": false
        },
        "Quality": {
            "Sharpness": 95.54405975341797,
            "Brightness": 63.867706298828125
        },
        "Mustache": {
            "Confidence": 97.05007934570312,
            "Value": false
        },
        "Beard": {
            "Confidence": 87.34505462646484,
```

```
                    "Value": false
                }
            },
            "Face": {
                "BoundingBox": {
                    "Width": 0.18562500178813934,
                    "Top": 0.1618015021085739,
                    "Left": 0.5575000047683716,
                    "Height": 0.24770642817020416
                },
                "FaceId": "ce7ed422-2132-4a11-ab14-06c5c410f29f",
                "ExternalImageId": "example-image.jpg",
                "Confidence": 99.993408203125,
                "ImageId": "8d67061e-90d2-598f-9fbd-29c8497039c0"
            }
        }
    ],
    "UnindexedFaces": [],
    "FaceModelVersion": "3.0",
    "OrientationCorrection": "ROTATE_0"
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的將臉部新增至集合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 IndexFaces。

list-collections

以下程式碼範例顯示如何使用 `list-collections`。

如需詳細資訊，請參閱列出的集合。

AWS CLI

列出可用的集合

下列`list-collections`命令會列出 AWS 帳戶中可用的集合。

```
aws rekognition list-collections
```

輸出：

```
{
```

```
    "FaceModelVersions": [
        "2.0",
        "3.0",
        "3.0",
        "3.0",
        "4.0",
        "1.0",
        "3.0",
        "4.0",
        "4.0",
        "4.0"
    ],
    "CollectionIds": [
        "MyCollection1",
        "MyCollection2",
        "MyCollection3",
        "MyCollection4",
        "MyCollection5",
        "MyCollection6",
        "MyCollection7",
        "MyCollection8",
        "MyCollection9",
        "MyCollection10"
    ]
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的列出集合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListCollections。

list-faces

以下程式碼範例顯示如何使用 list-faces。

如需更多資訊，請參閱集合中列出的人臉。

AWS CLI

列出集合中的臉部

下列list-faces命令會列出指定集合中的臉部。

```
aws rekognition list-faces \
```

```
    --collection-id MyCollection
```

輸出：

```
{
    "FaceModelVersion": "3.0",
    "Faces": [
        {
            "BoundingBox": {
                "Width": 0.5216310024261475,
                "Top": 0.3256250023841858,
                "Left": 0.13394300639629364,
                "Height": 0.3918749988079071
            },
            "FaceId": "0040279c-0178-436e-b70a-e61b074e96b0",
            "ExternalImageId": "image1.jpg",
            "Confidence": 100.0,
            "ImageId": "f976e487-3719-5e2d-be8b-ea2724c26991"
        },
        {
            "BoundingBox": {
                "Width": 0.5074880123138428,
                "Top": 0.3774999976158142,
                "Left": 0.18302799761295319,
                "Height": 0.3812499940395355
            },
            "FaceId": "086261e8-6deb-4bc0-ac73-ab22323cc38d",
            "ExternalImageId": "image2.jpg",
            "Confidence": 99.99930572509766,
            "ImageId": "ae1593b0-a8f6-5e24-a306-abf529e276fa"
        },
        {
            "BoundingBox": {
                "Width": 0.5574039816856384,
                "Top": 0.37187498807907104,
                "Left": 0.14559100568294525,
                "Height": 0.4181250035762787
            },
            "FaceId": "11c4bd3c-19c5-4eb8-aecc-24feb93a26e1",
            "ExternalImageId": "image3.jpg",
            "Confidence": 99.99960327148438,
            "ImageId": "80739b4d-883f-5b78-97cf-5124038e26b9"
        },
```

```
        {
            "BoundingBox": {
                "Width": 0.18562500178813934,
                "Top": 0.1618019938468933,
                "Left": 0.5575000047683716,
                "Height": 0.24770599603652954
            },
            "FaceId": "13692fe4-990a-4679-b14a-5ac23d135eab",
            "ExternalImageId": "image4.jpg",
            "Confidence": 99.99340057373047,
            "ImageId": "8df18239-9ad1-5acd-a46a-6581ff98f51b"
        },
        {
            "BoundingBox": {
                "Width": 0.5307819843292236,
                "Top": 0.2862499952316284,
                "Left": 0.1564060002565384,
                "Height": 0.3987500071525574
            },
            "FaceId": "2eb5f3fd-e2a9-4b1c-a89f-afa0a518fe06",
            "ExternalImageId": "image5.jpg",
            "Confidence": 99.99970245361328,
            "ImageId": "3c314792-197d-528d-bbb6-798ed012c150"
        },
        {
            "BoundingBox": {
                "Width": 0.5773710012435913,
                "Top": 0.34437501430511475,
                "Left": 0.12396000325679779,
                "Height": 0.4337500035762787
            },
            "FaceId": "57189455-42b0-4839-a86c-abda48b13174",
            "ExternalImageId": "image6.jpg",
            "Confidence": 100.0,
            "ImageId": "0aff2f37-e7a2-5dbc-a3a3-4ef6ec18eaa0"
        },
        {
            "BoundingBox": {
                "Width": 0.5349419713020325,
                "Top": 0.29124999046325684,
                "Left": 0.16389399766921997,
                "Height": 0.40187498927116394
            },
            "FaceId": "745f7509-b1fa-44e0-8b95-367b1359638a",
```

```
            "ExternalImageId": "image7.jpg",
            "Confidence": 99.99979400634766,
            "ImageId": "67a34327-48d1-5179-b042-01e52ccfeada"
        },
        {
            "BoundingBox": {
                "Width": 0.41499999165534973,
                "Top": 0.09187500178813934,
                "Left": 0.28083300590515137,
                "Height": 0.3112500011920929
            },
            "FaceId": "8d3cfc70-4ba8-4b36-9644-90fba29c2dac",
            "ExternalImageId": "image8.jpg",
            "Confidence": 99.99769592285156,
            "ImageId": "a294da46-2cb1-5cc4-9045-61d7ca567662"
        },
        {
            "BoundingBox": {
                "Width": 0.48166701197624207,
                "Top": 0.20999999344348907,
                "Left": 0.21250000596046448,
                "Height": 0.36125001311302185
            },
            "FaceId": "bd4ceb4d-9acc-4ab7-8ef8-1c2d2ba0a66a",
            "ExternalImageId": "image9.jpg",
            "Confidence": 99.99949645996094,
            "ImageId": "5e1a7588-e5a0-5ee3-bd00-c642518dfe3a"
        },
        {
            "BoundingBox": {
                "Width": 0.18562500178813934,
                "Top": 0.1618019938468933,
                "Left": 0.5575000047683716,
                "Height": 0.24770599603652954
            },
            "FaceId": "ce7ed422-2132-4a11-ab14-06c5c410f29f",
            "ExternalImageId": "image10.jpg",
            "Confidence": 99.99340057373047,
            "ImageId": "8d67061e-90d2-598f-9fbd-29c8497039c0"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的列出集合中的臉部。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFaces。

`list-stream-processors`

以下程式碼範例顯示如何使用 `list-stream-processors`。

AWS CLI

列出您帳戶中的串流處理器

下列`list-stream-processors`命令會列出您帳戶中的串流處理器，以及每個處理器的狀態。

```
aws rekognition list-stream-processors
```

輸出：

```
{
    "StreamProcessors": [
        {
            "Status": "STOPPED",
            "Name": "my-stream-processor"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的使用串流影片。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListStreamProcessors。

`recognize-celebrities`

以下程式碼範例顯示如何使用 `recognize-celebrities`。

如需詳細資訊，請參閱在映像中辨識名人。

AWS CLI

辨識影像中的名人

下列`recognize-celebrities`命令會辨識存放在 Amazon S3 儲存貯體中指定映像中的名人：

```
aws rekognition recognize-celebrities \
    --image "S3Object={Bucket=MyImageS3Bucket,Name=moviestars.jpg}"
```

輸出：

```
{
    "UnrecognizedFaces": [
        {
            "BoundingBox": {
                "Width": 0.14416666328907013,
                "Top": 0.07777778059244156,
                "Left": 0.625,
                "Height": 0.2746031880378723
            },
            "Confidence": 99.9990234375,
            "Pose": {
                "Yaw": 10.80408763885498,
                "Roll": -12.761146545410156,
                "Pitch": 10.96889877319336
            },
            "Quality": {
                "Sharpness": 94.1185531616211,
                "Brightness": 79.18367004394531
            },
            "Landmarks": [
                {
                    "Y": 0.18220913410186768,
                    "X": 0.6702951788902283,
                    "Type": "eyeLeft"
                },
                {
                    "Y": 0.16337193548679352,
                    "X": 0.7188183665275574,
                    "Type": "eyeRight"
                },
                {
                    "Y": 0.20739148557186127,
                    "X": 0.7055801749229431,
                    "Type": "nose"
                },
                {
                    "Y": 0.2889308035373688,
                    "X": 0.687512218952179,
```

```
                    "Type": "mouthLeft"
                },
                {
                    "Y": 0.2706988751888275,
                    "X": 0.7250053286552429,
                    "Type": "mouthRight"
                }
            ]
        }
    ],
    "CelebrityFaces": [
        {
            "MatchConfidence": 100.0,
            "Face": {
                "BoundingBox": {
                    "Width": 0.14000000059604645,
                    "Top": 0.1190476194024086,
                    "Left": 0.82833331823349,
                    "Height": 0.2666666805744171
                },
                "Confidence": 99.99359130859375,
                "Pose": {
                    "Yaw": -10.509642601013184,
                    "Roll": -14.51749324798584,
                    "Pitch": 13.799399375915527
                },
                "Quality": {
                    "Sharpness": 78.74752044677734,
                    "Brightness": 42.201324462890625
                },
                "Landmarks": [
                    {
                        "Y": 0.2290833294391632,
                        "X": 0.8709492087364197,
                        "Type": "eyeLeft"
                    },
                    {
                        "Y": 0.20639978349208832,
                        "X": 0.9153988361358643,
                        "Type": "eyeRight"
                    },
                    {
                        "Y": 0.25417643785476685,
                        "X": 0.8907724022865295,
```

```
                    "Type": "nose"
                },
                {
                    "Y": 0.32729196548461914,
                    "X": 0.8876466155052185,
                    "Type": "mouthLeft"
                },
                {
                    "Y": 0.3115464746952057,
                    "X": 0.9238573312759399,
                    "Type": "mouthRight"
                }
            ]
        },
        "Name": "Celeb A",
        "Urls": [
            "www.imdb.com/name/aaaaaaaaa"
        ],
        "Id": "1111111"
    },
    {
        "MatchConfidence": 97.0,
        "Face": {
            "BoundingBox": {
                "Width": 0.13333334028720856,
                "Top": 0.24920634925365448,
                "Left": 0.4449999928474426,
                "Height": 0.2539682686328888
            },
            "Confidence": 99.99979400634766,
            "Pose": {
                "Yaw": 6.557040691375732,
                "Roll": -7.316643714904785,
                "Pitch": 9.272967338562012
            },
            "Quality": {
                "Sharpness": 83.23492431640625,
                "Brightness": 78.83267974853516
            },
            "Landmarks": [
                {
                    "Y": 0.3625510632991791,
                    "X": 0.48898839950561523,
                    "Type": "eyeLeft"
```

```
                    },
                    {
                        "Y": 0.35366007685661316,
                        "X": 0.5313721299171448,
                        "Type": "eyeRight"
                    },
                    {
                        "Y": 0.3894785940647125,
                        "X": 0.5173314809799194,
                        "Type": "nose"
                    },
                    {
                        "Y": 0.44889405369758606,
                        "X": 0.5020005702972412,
                        "Type": "mouthLeft"
                    },
                    {
                        "Y": 0.4408611059188843,
                        "X": 0.5351271629333496,
                        "Type": "mouthRight"
                    }
                ]
            },
            "Name": "Celeb B",
            "Urls": [
                "www.imdb.com/name/bbbbbbbbb"
            ],
            "Id": "2222222"
        },
        {
            "MatchConfidence": 100.0,
            "Face": {
                "BoundingBox": {
                    "Width": 0.12416666746139526,
                    "Top": 0.2968254089355469,
                    "Left": 0.2150000035762787,
                    "Height": 0.23650793731212616
                },
                "Confidence": 99.99958801269531,
                "Pose": {
                    "Yaw": 7.801797866821289,
                    "Roll": -8.326810836791992,
                    "Pitch": 7.844768047332764
                },
```

```
                "Quality": {
                    "Sharpness": 86.93206024169922,
                    "Brightness": 79.81291198730469
                },
                "Landmarks": [
                    {
                        "Y": 0.4027804136276245,
                        "X": 0.2575301229953766,
                        "Type": "eyeLeft"
                    },
                    {
                        "Y": 0.3934555947780609,
                        "X": 0.2956969439983368,
                        "Type": "eyeRight"
                    },
                    {
                        "Y": 0.4309830069541931,
                        "X": 0.2837020754814148,
                        "Type": "nose"
                    },
                    {
                        "Y": 0.48186683654785156,
                        "X": 0.26812544465065,
                        "Type": "mouthLeft"
                    },
                    {
                        "Y": 0.47338807582855225,
                        "X": 0.29905644059181213,
                        "Type": "mouthRight"
                    }
                ]
            },
            "Name": "Celeb C",
            "Urls": [
                "www.imdb.com/name/ccccccccc"
            ],
            "Id": "3333333"
        },
        {
            "MatchConfidence": 97.0,
            "Face": {
                "BoundingBox": {
                    "Width": 0.11916666477918625,
                    "Top": 0.3698412775993347,
```

```
                "Left": 0.008333333767950535,
                "Height": 0.22698412835597992
            },
            "Confidence": 99.99999237060547,
            "Pose": {
                "Yaw": 16.38478660583496,
                "Roll": -1.0260354280471802,
                "Pitch": 5.975185394287109
            },
            "Quality": {
                "Sharpness": 83.23492431640625,
                "Brightness": 61.408443450927734
            },
            "Landmarks": [
                {
                    "Y": 0.4632347822189331,
                    "X": 0.049406956881284714,
                    "Type": "eyeLeft"
                },
                {
                    "Y": 0.46388113498687744,
                    "X": 0.08722897619009018,
                    "Type": "eyeRight"
                },
                {
                    "Y": 0.5020678639411926,
                    "X": 0.0758260041475296,
                    "Type": "nose"
                },
                {
                    "Y": 0.544157862663269,
                    "X": 0.054029736667871475,
                    "Type": "mouthLeft"
                },
                {
                    "Y": 0.5463630557060242,
                    "X": 0.08464983850717545,
                    "Type": "mouthRight"
                }
            ]
        },
        "Name": "Celeb D",
        "Urls": [
            "www.imdb.com/name/ddddddddd"
```

```
            ],
            "Id": "4444444"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的在映像中辨識名人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RecognizeCelebrities。

search-faces-by-image

以下程式碼範例顯示如何使用 search-faces-by-image。

如需詳細資訊，請參閱搜尋人臉 (映像)。

AWS CLI

在符合影像中最大臉部的集合中搜尋臉部。

下列search-faces-by-image命令會搜尋集合中符合指定影像中最大臉部的臉部：

```
aws rekognition search-faces-by-image \
    --image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"ExamplePerson.jpg"}}' \
    --collection-id MyFaceImageCollection

{
    "SearchedFaceBoundingBox": {
        "Width": 0.18562500178813934,
        "Top": 0.1618015021085739,
        "Left": 0.5575000047683716,
        "Height": 0.24770642817020416
    },
    "SearchedFaceConfidence": 99.993408203125,
    "FaceMatches": [
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.18562500178813934,
                    "Top": 0.1618019938468933,
                    "Left": 0.5575000047683716,
                    "Height": 0.24770599603652954
                },
                "FaceId": "ce7ed422-2132-4a11-ab14-06c5c410f29f",
```

```
            "ExternalImageId": "example-image.jpg",
            "Confidence": 99.99340057373047,
            "ImageId": "8d67061e-90d2-598f-9fbd-29c8497039c0"
        },
        "Similarity": 99.97913360595703
    },
    {
        "Face": {
            "BoundingBox": {
                "Width": 0.18562500178813934,
                "Top": 0.1618019938468933,
                "Left": 0.5575000047683716,
                "Height": 0.24770599603652954
            },
            "FaceId": "13692fe4-990a-4679-b14a-5ac23d135eab",
            "ExternalImageId": "image3.jpg",
            "Confidence": 99.99340057373047,
            "ImageId": "8df18239-9ad1-5acd-a46a-6581ff98f51b"
        },
        "Similarity": 99.97913360595703
    },
    {
        "Face": {
            "BoundingBox": {
                "Width": 0.41499999165534973,
                "Top": 0.09187500178813934,
                "Left": 0.28083300590515137,
                "Height": 0.3112500011920929
            },
            "FaceId": "8d3cfc70-4ba8-4b36-9644-90fba29c2dac",
            "ExternalImageId": "image2.jpg",
            "Confidence": 99.99769592285156,
            "ImageId": "a294da46-2cb1-5cc4-9045-61d7ca567662"
        },
        "Similarity": 99.18069458007812
    },
    {
        "Face": {
            "BoundingBox": {
                "Width": 0.48166701197624207,
                "Top": 0.20999999344348907,
                "Left": 0.21250000596046448,
                "Height": 0.36125001311302185
            },
```

```
            "FaceId": "bd4ceb4d-9acc-4ab7-8ef8-1c2d2ba0a66a",
            "ExternalImageId": "image1.jpg",
            "Confidence": 99.99949645996094,
            "ImageId": "5e1a7588-e5a0-5ee3-bd00-c642518dfe3a"
        },
        "Similarity": 98.66607666015625
    },
    {
        "Face": {
            "BoundingBox": {
                "Width": 0.5349419713020325,
                "Top": 0.29124999046325684,
                "Left": 0.16389399766921997,
                "Height": 0.40187498927116394
            },
            "FaceId": "745f7509-b1fa-44e0-8b95-367b1359638a",
            "ExternalImageId": "image9.jpg",
            "Confidence": 99.99979400634766,
            "ImageId": "67a34327-48d1-5179-b042-01e52ccfeada"
        },
        "Similarity": 98.24278259277344
    },
    {
        "Face": {
            "BoundingBox": {
                "Width": 0.5307819843292236,
                "Top": 0.2862499952316284,
                "Left": 0.1564060002565384,
                "Height": 0.3987500071525574
            },
            "FaceId": "2eb5f3fd-e2a9-4b1c-a89f-afa0a518fe06",
            "ExternalImageId": "image10.jpg",
            "Confidence": 99.99970245361328,
            "ImageId": "3c314792-197d-528d-bbb6-798ed012c150"
        },
        "Similarity": 98.10665893554688
    },
    {
        "Face": {
            "BoundingBox": {
                "Width": 0.5074880123138428,
                "Top": 0.3774999976158142,
                "Left": 0.18302799761295319,
                "Height": 0.3812499940395355
```

```
                },
                "FaceId": "086261e8-6deb-4bc0-ac73-ab22323cc38d",
                "ExternalImageId": "image6.jpg",
                "Confidence": 99.99930572509766,
                "ImageId": "ae1593b0-a8f6-5e24-a306-abf529e276fa"
            },
            "Similarity": 98.10526275634766
        },
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.5574039816856384,
                    "Top": 0.37187498807907104,
                    "Left": 0.14559100568294525,
                    "Height": 0.4181250035762787
                },
                "FaceId": "11c4bd3c-19c5-4eb8-aecc-24feb93a26e1",
                "ExternalImageId": "image5.jpg",
                "Confidence": 99.99960327148438,
                "ImageId": "80739b4d-883f-5b78-97cf-5124038e26b9"
            },
            "Similarity": 97.94659423828125
        },
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.5773710012435913,
                    "Top": 0.34437501430511475,
                    "Left": 0.12396000325679779,
                    "Height": 0.4337500035762787
                },
                "FaceId": "57189455-42b0-4839-a86c-abda48b13174",
                "ExternalImageId": "image8.jpg",
                "Confidence": 100.0,
                "ImageId": "0aff2f37-e7a2-5dbc-a3a3-4ef6ec18eaa0"
            },
            "Similarity": 97.93476867675781
        }
    ],

 "FaceModelVersion": "3.0"
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的使用映像搜尋臉部。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchFacesByImage。

search-faces

以下程式碼範例顯示如何使用 search-faces。

如需詳細資訊，請參閱搜尋人臉 (人臉 ID)。

AWS CLI

在符合臉部 ID 的集合中搜尋臉部。

下列search-faces命令會搜尋集合中符合指定臉部 ID 的臉部。

```
aws rekognition search-faces \
    --face-id 8d3cfc70-4ba8-4b36-9644-90fba29c2dac \
    --collection-id MyCollection
```

輸出：

```
{
    "SearchedFaceId": "8d3cfc70-4ba8-4b36-9644-90fba29c2dac",
    "FaceModelVersion": "3.0",
    "FaceMatches": [
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.48166701197624207,
                    "Top": 0.20999999344348907,
                    "Left": 0.21250000596046448,
                    "Height": 0.36125001311302185
                },
                "FaceId": "bd4ceb4d-9acc-4ab7-8ef8-1c2d2ba0a66a",
                "ExternalImageId": "image1.jpg",
                "Confidence": 99.99949645996094,
                "ImageId": "5e1a7588-e5a0-5ee3-bd00-c642518dfe3a"
            },
            "Similarity": 99.30997467041016
        },
        {
            "Face": {
```

```
                "BoundingBox": {
                    "Width": 0.18562500178813934,
                    "Top": 0.1618019938468933,
                    "Left": 0.5575000047683716,
                    "Height": 0.24770599603652954
                },
                "FaceId": "ce7ed422-2132-4a11-ab14-06c5c410f29f",
                "ExternalImageId": "example-image.jpg",
                "Confidence": 99.99340057373047,
                "ImageId": "8d67061e-90d2-598f-9fbd-29c8497039c0"
            },
            "Similarity": 99.24862670898438
        },
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.18562500178813934,
                    "Top": 0.1618019938468933,
                    "Left": 0.5575000047683716,
                    "Height": 0.24770599603652954
                },
                "FaceId": "13692fe4-990a-4679-b14a-5ac23d135eab",
                "ExternalImageId": "image3.jpg",
                "Confidence": 99.99340057373047,
                "ImageId": "8df18239-9ad1-5acd-a46a-6581ff98f51b"
            },
            "Similarity": 99.24862670898438
        },
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.5349419713020325,
                    "Top": 0.29124999046325684,
                    "Left": 0.16389399766921997,
                    "Height": 0.40187498927116394
                },
                "FaceId": "745f7509-b1fa-44e0-8b95-367b1359638a",
                "ExternalImageId": "image9.jpg",
                "Confidence": 99.99979400634766,
                "ImageId": "67a34327-48d1-5179-b042-01e52ccfeada"
            },
            "Similarity": 96.73158264160156
        },
        {
```

```
            "Face": {
                "BoundingBox": {
                    "Width": 0.5307819843292236,
                    "Top": 0.2862499952316284,
                    "Left": 0.1564060002565384,
                    "Height": 0.3987500071525574
                },
                "FaceId": "2eb5f3fd-e2a9-4b1c-a89f-afa0a518fe06",
                "ExternalImageId": "image10.jpg",
                "Confidence": 99.99970245361328,
                "ImageId": "3c314792-197d-528d-bbb6-798ed012c150"
            },
            "Similarity": 96.48291015625
        },
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.5074880123138428,
                    "Top": 0.3774999976158142,
                    "Left": 0.18302799761295319,
                    "Height": 0.3812499940395355
                },
                "FaceId": "086261e8-6deb-4bc0-ac73-ab22323cc38d",
                "ExternalImageId": "image6.jpg",
                "Confidence": 99.99930572509766,
                "ImageId": "ae1593b0-a8f6-5e24-a306-abf529e276fa"
            },
            "Similarity": 96.43287658691406
        },
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.5574039816856384,
                    "Top": 0.37187498807907104,
                    "Left": 0.14559100568294525,
                    "Height": 0.4181250035762787
                },
                "FaceId": "11c4bd3c-19c5-4eb8-aecc-24feb93a26e1",
                "ExternalImageId": "image5.jpg",
                "Confidence": 99.99960327148438,
                "ImageId": "80739b4d-883f-5b78-97cf-5124038e26b9"
            },
            "Similarity": 95.25305938720703
        },
```

```
        {
            "Face": {
                "BoundingBox": {
                    "Width": 0.5773710012435913,
                    "Top": 0.34437501430511475,
                    "Left": 0.12396000325679779,
                    "Height": 0.4337500035762787
                },
                "FaceId": "57189455-42b0-4839-a86c-abda48b13174",
                "ExternalImageId": "image8.jpg",
                "Confidence": 100.0,
                "ImageId": "0aff2f37-e7a2-5dbc-a3a3-4ef6ec18eaa0"
            },
            "Similarity": 95.22837829589844
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的使用臉部 ID 搜尋臉部。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchFaces。

start-celebrity-recognition

以下程式碼範例顯示如何使用 start-celebrity-recognition。

AWS CLI

開始辨識已儲存影片中的名人

下列start-celebrity-recognition命令會啟動任務，在存放在 Amazon S3 儲存貯體的指定影片檔案中尋找名人。

```
aws rekognition start-celebrity-recognition \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

輸出：

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的在儲存的影片中辨識名人。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartCelebrityRecognition。

start-content-moderation

以下程式碼範例顯示如何使用 start-content-moderation。

AWS CLI

開始辨識已儲存影片中的不安全內容

下列start-content-moderation命令會啟動任務，以偵測存放在 Amazon S3 儲存貯體的指定影片檔案中不安全的內容。

```
aws rekognition start-content-moderation \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

輸出：

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的偵測不安全的存放影片。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartContentModeration。

start-face-detection

以下程式碼範例顯示如何使用 start-face-detection。

AWS CLI

偵測影片中的臉部

下列start-face-detection命令會啟動任務，以偵測存放在 Amazon S3 儲存貯體的指定影片檔案中的臉部。

```
aws rekognition start-face-detection
```

```
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

輸出：

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的在存放的影片中偵測臉部。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartFaceDetection。

start-face-search

以下程式碼範例顯示如何使用 start-face-search。

AWS CLI

搜尋集合中與影片中偵測到的臉部相符的臉部

下列start-face-search命令會啟動任務，以搜尋集合中與 Amazon S3 儲存貯體中指定影片檔案中偵測到的臉部相符的臉部。

```
aws rekognition start-face-search \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}" \
    --collection-id collection
```

輸出：

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的搜尋儲存的臉部影片。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartFaceSearch。

start-label-detection

以下程式碼範例顯示如何使用 start-label-detection。

AWS CLI

偵測影片中的物件和場景

下列start-label-detection命令會啟動任務，以偵測存放在 Amazon S3 儲存貯體中指定影片檔案中的物件和場景。

```
aws rekognition start-label-detection \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

輸出：

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的偵測影片中的標籤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartLabelDetection。

start-person-tracking

以下程式碼範例顯示如何使用 start-person-tracking。

AWS CLI

在儲存的影片中開始人員的路徑

下列start-person-tracking命令會啟動任務，以追蹤使用者在存放於 Amazon S3 儲存貯體的指定影片片段中採取的路徑：

```
aws rekognition start-person-tracking \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

輸出：

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的人員路徑。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartPersonTracking。

start-stream-processor

以下程式碼範例顯示如何使用 start-stream-processor。

AWS CLI

啟動串流處理器

下列start-stream-processor命令會啟動指定的影片串流處理器。

```
aws rekognition start-stream-processor \
    --name my-stream-processor
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的使用串流影片。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartStreamProcessor。

stop-stream-processor

以下程式碼範例顯示如何使用 stop-stream-processor。

AWS CLI

停止執行中的串流處理器

下列stop-stream-processor命令會停止指定的執行中串流處理器。

```
aws rekognition stop-stream-processor \
    --name my-stream-processor
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Rekognition 開發人員指南》中的使用串流影片。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopStreamProcessor。

AWS RAM 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS RAM。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`accept-resource-share-invitation`

以下程式碼範例顯示如何使用 `accept-resource-share-invitation`。

AWS CLI

接受資源共享邀請

下列`accept-resource-share-invitation`範例接受指定的資源共享邀請。受邀帳戶中的委託人可以立即開始使用共享中的資源。

```
aws ram accept-resource-share-invitation \
    --resource-share-invitation-arn arn:aws:ram:us-west-2:111111111111:resource-share-invitation/1e3477be-4a95-46b4-bbe0-c4001EXAMPLE
```

輸出：

```
{
    "resourceShareInvitation": {
        "resourceShareInvitationArn": "arn:aws:ram:us-west-2:111111111111:resource-share-invitation/1e3477be-4a95-46b4-bbe0-c4001EXAMPLE",
        "resourceShareName": "MyLicenseShare",
        "resourceShareArn": "arn:aws:ram:us-west-2:111111111111:resource-share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE",
        "senderAccountId": "111111111111",
```

```
        "receiverAccountId": "222222222222",
        "invitationTimestamp": "2021-09-22T15:07:35.620000-07:00",
        "status": "ACCEPTED"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptResourceShareInvitation。

associate-resource-share-permission

以下程式碼範例顯示如何使用 associate-resource-share-permission。

AWS CLI

將 RAM 受管許可與資源共享建立關聯

下列associate-resource-share-permission範例會將相關資源類型的現有受管許可取代為指定的受管許可。對相關資源類型之所有資源的存取由新的許可管理。

```
aws ram associate-resource-share-permission \
    --permission-arn arn:aws:ram::aws:permission/
AWSRAMPermissionGlueDatabaseReadWrite \
    --replace \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE
```

輸出：

```
{
    "returnValue": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateResourceSharePermission。

associate-resource-share

以下程式碼範例顯示如何使用 associate-resource-share。

AWS CLI

範例 1：將資源與資源共用建立關聯

下列associate-resource-share範例會將授權組態新增至指定的資源共用。

```
aws ram associate-resource-share \
    --resource-share arn:aws:ram:us-west-2:123456789012:resource-share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE \
    --resource-arns arn:aws:license-manager:us-west-2:123456789012:license-configuration:lic-36be0485f5ae379cc74cf8e92EXAMPLE
```

輸出：

```
{
    "resourceShareAssociations": [
        {
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE",
            "associatedEntity": "arn:aws:license-manager:us-west-2:123456789012:license-configuration:lic-36be0485f5ae379cc74cf8e92EXAMPLE",
            "associationType": "RESOURCE",
            "status": "ASSOCIATING",
            "external": false
        }
    ]
}
```

範例 2：將委託人與資源共享建立關聯

下列associate-resource-share範例會將指定資源共享的存取權授予指定組織單位中的所有帳戶。

```
aws ram associate-resource-share \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE \
    --principals arn:aws:organizations::123456789012:ou/o-63bEXAMPLE/ou-46xi-rEXAMPLE
```

輸出：

```
{
    "resourceShareAssociations": [
        {
            "status": "ASSOCIATING",
            "associationType": "PRINCIPAL",
```

```
            "associatedEntity": "arn:aws:organizations::123456789012:ou/
o-63bEXAMPLE/ou-46xi-rEXAMPLE",
            "external": false,
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateResourceShare。

create-resource-share

以下程式碼範例顯示如何使用 create-resource-share。

AWS CLI

範例 1：建立資源共享

下列create-resource-share範例會建立具有指定名稱的空資源共享。您必須分別將資源、委託人和許可新增至共享。

```
aws ram create-resource-share \
    --name MyNewResourceShare
```

輸出：

```
{
    "resourceShare": {
        "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/4476c27d-8feb-4b21-afe9-7de23EXAMPLE",
        "name": "MyNewResourceShare",
        "owningAccountId": "123456789012",
        "allowExternalPrincipals": true,
        "status": "ACTIVE",
        "creationTime": 1634586271.302,
        "lastUpdatedTime": 1634586271.302
    }
}
```

範例 2：以委託人身分與 AWS 帳戶建立資源共享

下列create-resource-share範例會建立資源共用，並授予指定 AWS 帳戶 (222222222222) 的存取權。如果指定的委託人不屬於同一個 AWS 組織，則會傳送邀請，且必須先接受才能授予存取權。

```
aws ram create-resource-share \
    --name MyNewResourceShare \
    --principals 222222222222
```

範例 3：建立僅限您 AWS Organization 的資源共用

以下create-resource-share範例建立的資源共享僅限於您帳戶所屬 AWS 組織的帳戶，並將指定的 OU 新增為委託人。該 OU 中的所有帳戶都可以使用資源共享中的資源。

```
aws ram create-resource-share \
    --name MyNewResourceShare \
    --no-allow-external-principals \
    --principals arn:aws:organizations::123456789012:ou/o-63bEXAMPLE/ou-46xi-
rEXAMPLE
```

輸出：

```
{
    "resourceShare": {
        "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7be8694e-095c-41ca-9ce8-7be4aEXAMPLE",
        "name": "MyNewResourceShare",
        "owningAccountId": "123456789012",
        "allowExternalPrincipals": false,
        "status": "ACTIVE",
        "creationTime": 1634587042.49,
        "lastUpdatedTime": 1634587042.49
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateResourceShare。

delete-resource-share

以下程式碼範例顯示如何使用 delete-resource-share。

AWS CLI

刪除資源共享

下列delete-resource-share範例會刪除指定的資源共享。

```
aws ram delete-resource-share \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE
```

下列輸出表示成功：

```
{
    "returnValue": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteResourceShare。

disassociate-resource-share-permission

以下程式碼範例顯示如何使用 disassociate-resource-share-permission。

AWS CLI

從資源共用中移除資源類型的 RAM 受管許可

下列disassociate-resource-share-permission範例會從指定的資源共用中移除 Glue 資料庫的 RAM 受管許可。

```
aws ram disassociate-resource-share-permission \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE \
    --permission-arn arn:aws:ram::aws:permission/AWSRAMPermissionGlueDatabaseReadWrite
```

輸出：

```
{
    "returnValue": true
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateResourceSharePermission。

`disassociate-resource-share`

以下程式碼範例顯示如何使用 `disassociate-resource-share`。

AWS CLI

從資源共用中移除資源

下列`disassociate-resource-share`範例會從指定的資源共用中移除指定的資源，在此情況下為 VPC 子網路。任何有權存取資源共享的主體都無法再對該資源執行操作。

```
aws ram disassociate-resource-share \
    --resource-arns arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0250c25a1fEXAMPLE \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-
b505-7e2a-420d-6f5d3EXAMPLE
```

輸出：

```
{
    "resourceShareAssociations": [
        "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
        "associatedEntity": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0250c25a1fEXAMPLE",
        "associationType": "RESOURCE",
        "status": "DISASSOCIATING",
        "external": false
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateResourceShare。

`enable-sharing-with-aws-organization`

以下程式碼範例顯示如何使用 `enable-sharing-with-aws-organization`。

AWS CLI

啟用跨 AWS Organizations 的資源共用

下列enable-sharing-with-aws-organization範例可讓您跨組織和組織單位進行資源共用。

```
aws ram enable-sharing-with-aws-organization
```

以下輸出表示成功。

```
{
    "returnValue": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableSharingWithAwsOrganization。

get-permission

以下程式碼範例顯示如何使用 get-permission。

AWS CLI

擷取 RAM 受管許可的詳細資訊

下列get-permission範例顯示指定 RAM 受管許可預設版本的詳細資訊。

```
aws ram get-permission \
    --permission-arn arn:aws:ram::aws:permission/
AWSRAMPermissionGlueTableReadWriteForDatabase
```

輸出：

```
{
    "permission": {
        "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueTableReadWriteForDatabase",
        "version": "2",
        "defaultVersion": true,
        "name": "AWSRAMPermissionGlueTableReadWriteForDatabase",
```

```
        "resourceType": "glue:Database",
        "permission": "{\"Effect\":\"Allow\",\"Action\":[\"glue:GetTable
\", \"glue:UpdateTable\", \"glue:DeleteTable\", \"glue:BatchDeleteTable\",
 \"glue:BatchDeleteTableVersion\", \"glue:GetTableVersion\", \"glue:GetTableVersions
\", \"glue:GetPartition\", \"glue:GetPartitions\", \"glue:BatchGetPartition\",
 \"glue:BatchCreatePartition\", \"glue:CreatePartition\", \"glue:UpdatePartition
\", \"glue:BatchDeletePartition\", \"glue:DeletePartition\", \"glue:GetTables\",
 \"glue:SearchTables\"]}",
        "creationTime": 1624912434.431,
        "lastUpdatedTime": 1624912434.431,
        "isResourceTypeDefault": false
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPermission。

get-resource-policies

以下程式碼範例顯示如何使用 get-resource-policies。

AWS CLI

取得資源的政策

下列get-resource-policies範例顯示與資源共享相關聯之指定資源的資源型許可政策。

```
aws ram get-resource-policies \
    --resource-arns arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0250c25a1fEXAMPLE
```

輸出：

```
{
    "policies": [
         "{\"Version\":\"2008-10-17\",\"Statement\":[{\"Sid\":\"RamStatement1\",
\"Effect\":\"Allow\",\"Principal\":{\"AWS\":[]},\"Action\":[\"ec2:RunInstances
\",\"ec2:CreateNetworkInterface\",\"ec2:DescribeSubnets\"],\"Resource\":
\"arn:aws:ec2:us-west-2:123456789012:subnet/subnet-0250c25a1fEXAMPLE\"}]}"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResourcePolicies。

get-resource-share-associations

以下程式碼範例顯示如何使用 get-resource-share-associations。

AWS CLI

範例 1：列出所有資源類型的所有資源關聯

下列get-resource-share-associations範例列出所有資源共用中所有資源類型的資源關聯。

```
aws ram get-resource-share-associations \
    --association-type RESOURCE
```

輸出：

```
{
    "resourceShareAssociations": [
        {
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
            "associatedEntity": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0250c25a1fEXAMPLE",
            "resourceShareName": "MySubnetShare",
            "associationType": "RESOURCE",
            "status": "ASSOCIATED",
            "creationTime": 1565303590.973,
            "lastUpdatedTime": 1565303591.695,
            "external": false
        },
        {
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/8167bdfe-4480-4a01-8632-315e0EXAMPLE",
            "associatedEntity": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-36be0485f5ae379cc74cf8e92EXAMPLE",
            "resourceShareName": "MyLicenseShare",
            "associationType": "RESOURCE",
            "status": "ASSOCIATED",
            "creationTime": 1632342958.457,
            "lastUpdatedTime": 1632342958.907,
            "external": false
        }
    ]
```

```
}
```

範例 2：列出資源共享的主體關聯

下列get-resource-share-associations範例僅列出指定資源共用的委託人關聯。

```
aws ram get-resource-share-associations \
   --resource-share-arns arn:aws:ram:us-west-2:123456789012:resource-
share/7be8694e-095c-41ca-9ce8-7be4aEXAMPLE \
   --association-type PRINCIPAL
```

輸出：

```
{
    "resourceShareAssociations": [
        {
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7be8694e-095c-41ca-9ce8-7be4aEXAMPLE",
            "resourceShareName": "MyNewResourceShare",
            "associatedEntity": "arn:aws:organizations::123456789012:ou/
o-63bEXAMPLE/ou-46xi-rEXAMPLE",
            "associationType": "PRINCIPAL",
            "status": "ASSOCIATED",
            "creationTime": 1634587042.49,
            "lastUpdatedTime": 1634587044.291,
            "external": false
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResourceShareAssociations。

get-resource-share-invitations

以下程式碼範例顯示如何使用 get-resource-share-invitations。

AWS CLI

列出您的資源共享邀請

下列get-resource-share-invitations範例列出您目前的資源共用邀請。

```
aws ram get-resource-share-invitations
```

輸出：

```
{
    "resourceShareInvitations": [
        {
            "resourceShareInvitationArn": "arn:aws:ram:us-
west2-1:111111111111:resource-share-invitation/32b639f0-14b8-7e8f-55ea-
e6117EXAMPLE",
            "resourceShareName": "project-resource-share",
            "resourceShareArn": "arn:aws:ram:us-west-2:111111111111:resource-share/
fcb639f0-1449-4744-35bc-a983fEXAMPLE",
            "senderAccountId": "111111111111",
            "receiverAccountId": "222222222222",
            "invitationTimestamp": 1565312166.258,
            "status": "PENDING"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResourceShareInvitations。

get-resource-shares

以下程式碼範例顯示如何使用 get-resource-shares。

AWS CLI

範例 1：列出您擁有的資源共用並與他人共用

下列get-resource-shares範例列出已建立和正在與他人共用的資源共用。

```
aws ram get-resource-shares \
    --resource-owner SELF
```

輸出：

```
{
    "resourceShares": [
        {
```

```
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/3ab63985-99d9-1cd2-7d24-75e93EXAMPLE",
            "name": "my-resource-share",
            "owningAccountId": "123456789012",
            "allowExternalPrincipals": false,
            "status": "ACTIVE",
            "tags": [
                {
                    "key": "project",
                    "value": "lima"
                }
            ]
            "creationTime": 1565295733.282,
            "lastUpdatedTime": 1565295733.282
        },
        {
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
            "name": "my-resource-share",
            "owningAccountId": "123456789012",
            "allowExternalPrincipals": true,
            "status": "ACTIVE",
            "creationTime": 1565295733.282,
            "lastUpdatedTime": 1565295733.282
        }
    ]
}
```

範例 2：列出其他人擁有並與您共用的資源共用

下列get-resource-shares範例列出其他人建立並與您共用的資源共用。在此範例中，沒有 。

```
aws ram get-resource-shares \
    --resource-owner OTHER-ACCOUNTS
```

輸出：

```
{
    "resourceShares": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResourceShares。

list-pending-invitation-resources

以下程式碼範例顯示如何使用 list-pending-invitation-resources。

AWS CLI

列出待定資源共享中可用的資源

下列list-pending-invitation-resources範例列出與指定邀請相關聯的資源共用中的所有資源。

```
aws ram list-pending-invitation-resources \
    --resource-share-invitation-arn arn:aws:ram:us-west-2:123456789012:resource-share-invitation/1e3477be-4a95-46b4-bbe0-c4001EXAMPLE
```

輸出：

```
{
   "resources": [
        {
            "arn": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-04a555b0e6EXAMPLE",
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7be8694e-095c-41ca-9ce8-7be4aEXAMPLE",
            "creationTime": 1634676051.269,
            "lastUpdatedTime": 1634676052.07,
            "status": "AVAILABLE",
            "type": "ec2:Subnet"
        },
        {
            "arn": "arn:aws:license-manager:us-west-2:123456789012:license-
configuration:lic-36be0485f5ae379cc74cf8e92EXAMPLE",
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
            "creationTime": 1624912434.431,
            "lastUpdatedTime": 1624912434.431,
            "status": "AVAILABLE",
            "type": "license-manager:LicenseConfiguration"
      }
   ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPendingInvitationResources。

list-permissions

以下程式碼範例顯示如何使用 list-permissions。

AWS CLI

列出可用的 RAM 受管許可

下列list-permissions範例列出僅適用於 Glue 資料庫資源類型的所有 RAM AWS 受管許可。

```
aws ram list-permissions \
    --resource-type glue:Database
```

輸出：

```
{
    "permissions": [
        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMDefaultPermissionGlueDatabase",
            "version": "1",
            "defaultVersion": true,
            "name": "AWSRAMDefaultPermissionGlueDatabase",
            "resourceType": "glue:Database",
            "creationTime": 1592007820.935,
            "lastUpdatedTime": 1592007820.935,
            "isResourceTypeDefault": true
        },
        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueAllTablesReadWriteForDatabase",
            "version": "2",
            "defaultVersion": true,
            "name": "AWSRAMPermissionGlueAllTablesReadWriteForDatabase",
            "resourceType": "glue:Database",
            "creationTime": 1624912413.323,
            "lastUpdatedTime": 1624912413.323,
            "isResourceTypeDefault": false
        },
        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueDatabaseReadWrite",
            "version": "2",
```

```
            "defaultVersion": true,
            "name": "AWSRAMPermissionGlueDatabaseReadWrite",
            "resourceType": "glue:Database",
            "creationTime": 1624912417.4,
            "lastUpdatedTime": 1624912417.4,
            "isResourceTypeDefault": false
        },
        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueTableReadWriteForDatabase",
            "version": "2",
            "defaultVersion": true,
            "name": "AWSRAMPermissionGlueTableReadWriteForDatabase",
            "resourceType": "glue:Database",
            "creationTime": 1624912434.431,
            "lastUpdatedTime": 1624912434.431,
            "isResourceTypeDefault": false
        }
    ]
}
```

下列list-permissions範例顯示所有資源類型的可用 RAM 受管許可。

```
aws ram list-permissions
```

輸出：

```
{
    "permissions": [
        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMBlankEndEntityCertificateAPICSRPassthroughIssuanceCertificateAuthority",
            "version": "1",
            "defaultVersion": true,
            "name":
 "AWSRAMBlankEndEntityCertificateAPICSRPassthroughIssuanceCertificateAuthority",
            "resourceType": "acm-pca:CertificateAuthority",
            "creationTime": 1623264861.085,
            "lastUpdatedTime": 1623264861.085,
            "isResourceTypeDefault": false
        },
        {
            "arn": "arn:aws:ram::aws:permission/AWSRAMDefaultPermissionAppMesh",
```

```
            "version": "1",
            "defaultVersion": true,
            "name": "AWSRAMDefaultPermissionAppMesh",
            "resourceType": "appmesh:Mesh",
            "creationTime": 1589307188.584,
            "lastUpdatedTime": 1589307188.584,
            "isResourceTypeDefault": true
        },
        ...TRUNCATED FOR BREVITY...
        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMSubordinateCACertificatePathLen0IssuanceCertificateAuthority",
            "version": "1",
            "defaultVersion": true,
            "name":
 "AWSRAMSubordinateCACertificatePathLen0IssuanceCertificateAuthority",
            "resourceType": "acm-pca:CertificateAuthority",
            "creationTime": 1623264876.75,
            "lastUpdatedTime": 1623264876.75,
            "isResourceTypeDefault": false
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPermissions。

list-principals

以下程式碼範例顯示如何使用 list-principals。

AWS CLI

列出可存取資源的主體

下列list-principals範例顯示可透過任何資源共用存取指定類型資源的委託人清單。

```
aws ram list-principals \
    --resource-type ec2:Subnet
```

輸出：

```
{
```

```
    "principals": [
        {
            "id": "arn:aws:organizations::123456789012:ou/o-gx7EXAMPLE/ou-29c5-
zEXAMPLE",
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
            "creationTime": 1565298209.737,
            "lastUpdatedTime": 1565298211.019,
            "external": false
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPrincipals。

`list-resource-share-permissions`

以下程式碼範例顯示如何使用 `list-resource-share-permissions`。

AWS CLI

列出目前連接至資源共享的所有 RAM 受管許可

下列`list-resource-share-permissions`範例列出連接至指定資源共享的所有 RAM 受管許可。

```
aws ram list-resource-share-permissions \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE
```

輸出：

```
{
    "permissions": [
        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMDefaultPermissionLicenseConfiguration",
            "version": "1",
            "resourceType": "license-manager:LicenseConfiguration",
            "status": "ASSOCIATED",
            "lastUpdatedTime": 1632342984.234
        },
```

```
        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueDatabaseReadWrite",
            "version": "2",
            "resourceType": "glue:Database",
            "status": "ASSOCIATED",
            "lastUpdatedTime": 1632512462.297
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourceSharePermissions。

list-resource-types

以下程式碼範例顯示如何使用 list-resource-types。

AWS CLI

列出 RAM AWS 支援的資源類型

下列list-resource-types範例列出 RAM AWS 目前支援的所有資源類型。

```
aws ram list-resource-types
```

輸出：

```
{
    "resourceTypes": [
        {
            "resourceType": "route53resolver:FirewallRuleGroup",
            "serviceName": "route53resolver"
        },
        {
            "resourceType": "ec2:LocalGatewayRouteTable",
            "serviceName": "ec2"
        },
        ...OUTPUT TRUNCATED FOR BREVITY...
        {
            "resourceType": "ec2:Subnet",
            "serviceName": "ec2"
        },
```

```
        {
            "resourceType": "ec2:TransitGatewayMulticastDomain",
            "serviceName": "ec2"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourceTypes。

`list-resources`

以下程式碼範例顯示如何使用 `list-resources`。

AWS CLI

列出與資源共享相關聯的資源

下列`list-resources`範例列出指定資源共用中屬於指定資源類型的所有資源。

```
aws ram list-resources \
    --resource-type ec2:Subnet \
    --resource-owner SELF \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE
```

輸出：

```
{
    "resources": [
        {
            "arn": "aarn:aws:ec2:us-west-2:123456789012:subnet/subnet-0250c25a1f4e15235",
            "type": "ec2:Subnet",
            "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
            "creationTime": 1565301545.023,
            "lastUpdatedTime": 1565301545.947
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResources。

promote-resource-share-created-from-policy

以下程式碼範例顯示如何使用 promote-resource-share-created-from-policy。

AWS CLI

將以資源政策為基礎的資源共用提升為 RAM AWS 中的完整功能

下列promote-resource-share-created-from-policy範例會採用您透過連接以資源為基礎的政策而隱含建立的資源共用，並將其轉換為與 AWS RAM 主控台及其 CLI 和 API 操作完全運作。

```
aws ram promote-resource-share-created-from-policy \
    --resource-share-arn arn:aws:ram:us-east-1:123456789012:resource-share/91fa8429-2d06-4032-909a-90909EXAMPLE
```

輸出：

```
{
    "returnValue": true
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PromoteResourceShareCreatedFromPolicy。

reject-resource-share-invitation

以下程式碼範例顯示如何使用 reject-resource-share-invitation。

AWS CLI

拒絕資源共享邀請

下列reject-resource-share-invitation範例會拒絕指定的資源共享邀請。

```
aws ram reject-resource-share-invitation \
    --resource-share-invitation-arn arn:aws:ram:us-west-2:111111111111:resource-share-invitation/32b639f0-14b8-7e8f-55ea-e6117EXAMPLE
```

輸出：

```
"resourceShareInvitations": [
    {
        "resourceShareInvitationArn": "arn:aws:ram:us-west2-1:111111111111:resource-
share-invitation/32b639f0-14b8-7e8f-55ea-e6117EXAMPLE",
        "resourceShareName": "project-resource-share",
        "resourceShareArn": "arn:aws:ram:us-west-2:111111111111:resource-share/
fcb639f0-1449-4744-35bc-a983fEXAMPLE",
        "senderAccountId": "111111111111",
        "receiverAccountId": "222222222222",
        "invitationTimestamp": 1565319592.463,
        "status": "REJECTED"
    }
]
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RejectResourceShareInvitation。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤新增至資源共享

下列tag-resource範例會將標籤索引鍵project和相關聯的值lima新增至指定的資源共享。

```
aws ram tag-resource \
    --tags key=project,value=lima \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-
b505-7e2a-420d-6f5d3EXAMPLE
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源共享中移除標籤

下列untag-resource範例會從指定的資源共用中移除project標籤索引鍵和相關聯的值。

```
aws ram untag-resource \
    --tag-keys project \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-resource-share

以下程式碼範例顯示如何使用 update-resource-share。

AWS CLI

更新資源共享

下列update-resource-share範例會變更指定的資源共用，以允許不在 AWS 組織的外部主體。

```
aws ram update-resource-share \
    --allow-external-principals \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE
```

輸出：

```
{
    "resourceShare": {
        "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
        "name": "my-resource-share",
        "owningAccountId": "123456789012",
        "allowExternalPrincipals": true,
        "status": "ACTIVE",
        "creationTime": 1565295733.282,
        "lastUpdatedTime": 1565303080.023
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateResourceShare。

使用 的資源總管範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Resource Explorer 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-default-view`

以下程式碼範例顯示如何使用 `associate-default-view`。

AWS CLI

將 Resource Explorer 檢視設定為其 AWS 區域的預設值

下列`associate-default-view`範例會將 ARN 指定的檢視設定為您呼叫操作之 AWS 區域的預設檢視。

```
aws resource-explorer-2 associate-default-view \
    --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-View/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

輸出：

```
{
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}
```

如需詳細資訊，請參閱 AWS Resource Explorer 使用者指南中的在 AWS 區域中設定預設檢視。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateDefaultView。

batch-get-view

以下程式碼範例顯示如何使用 batch-get-view。

AWS CLI

擷取多個 Resource Explorer 檢視的詳細資訊

下列batch-get-view範例顯示其 ARNs 所指定之兩個檢視的詳細資訊。使用空格分隔 --view-arn 參數中的多個 ARNs。

```
aws resource-explorer-2 batch-get-view \
    --view-arns arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222, \
                arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

輸出：

```
{
    "Views": [
        {
            "Filters": {
                "FilterString": "service:ec2"
            },
            "IncludedProperties": [
                {
                    "Name": "tags"
                }
            ],
            "LastUpdatedAt": "2022-07-13T21:33:45.249000+00:00",
            "Owner": "123456789012",
            "Scope": "arn:aws:iam::123456789012:root",
            "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222"
        },
        {
            "Filters": {
                "FilterString": ""
            },
            "IncludedProperties": [
                {
                    "Name": "tags"
```

```
                }
            ],
            "LastUpdatedAt": "2022-07-13T20:34:11.314000+00:00",
            "Owner": "123456789012",
            "Scope": "arn:aws:iam::123456789012:root",
            "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-
Main-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
        }
    ]
    "Errors": []
}
```

如需檢視的詳細資訊，請參閱《Resource Explorer 使用者指南》中的關於 AWS Resource Explorer 檢視。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetView。

`create-index`

以下程式碼範例顯示如何使用 `create-index`。

AWS CLI

建立索引以開啟 AWS 區域中的資源總管

下列`create-index`範例會在呼叫 操作的 AWS 區域中建立本機索引。CLI AWS 會自動產生隨機`client-token`參數值， AWS 如果您未指定值，則將其包含在對 的呼叫中。

```
aws resource-explorer-2 create-index \
    --region us-east-1
```

輸出：

```
{
    "Arn": "arn:aws:resource-explorer-2:us-east-1:123456789012:index/EXAMPLE8-90ab-
cdef-fedc-EXAMPLE22222c",
    "CreatedAt": "2022-11-01T20:00:59.149Z",
    "State": "CREATING"
}
```

建立本機索引後，您可以執行 update-index-type 命令，將其轉換為帳戶的彙總器索引。

如需詳細資訊，請參閱《Resource Explorer 使用者指南》中的在 AWS 區域中開啟 Resource Explorer 以編製資源索引。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateIndex。

create-view

以下程式碼範例顯示如何使用 create-view。

AWS CLI

範例 1：為 AWS 區域中的索引建立未篩選的檢視

下列create-view範例會在指定的 AWS 區域中建立檢視，傳回區域中的所有結果，而不進行任何篩選。檢視包含傳回結果上的選用標籤欄位。由於此檢視是在包含彙總工具索引的區域中建立，因此可以包含帳戶中包含資源總管索引之所有區域的結果。

```
aws resource-explorer-2 create-view \
    --view-name My-Main-View \
    --included-properties Name=tags \
    --region us-east-1
```

輸出：

```
{
    "View": {
        "Filters": {
            "FilterString": ""
        },
        "IncludedProperties": [
            {
                "Name": "tags"
            }
        ],
        "LastUpdatedAt": "2022-07-13T20:34:11.314000+00:00",
        "Owner": "123456789012",
        "Scope": "arn:aws:iam::123456789012:root",
        "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
    }
}
```

範例 2：建立僅傳回與 Amazon EC2 相關聯資源的檢視

以下會在 AWS 區域中`create-view`建立檢視`us-east-1`，僅傳回區域中與 Amazon EC2 服務相關聯的資源。檢視包含傳回結果的選用`Tags`欄位。由於此檢視是在包含彙總工具索引的區域中建立，因此可以包含帳戶中包含資源總管索引之所有區域的結果。

```
aws resource-explorer-2 create-view \
    --view-name My-EC2-Only-View \
    --included-properties Name=tags \
    --filters FilterString="service:ec2" \
    --region us-east-1
```

輸出：

```
{
    "View": {
        "Filters": {
            "FilterString": "service:ec2"
        },
        "IncludedProperties": [
            {
                "Name":"tags"
            }
        ],
        "LastUpdatedAt": "2022-07-13T21:35:09.059Z",
        "Owner": "123456789012",
        "Scope": "arn:aws:iam::123456789012:root",
        "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-
Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222"
    }
}
```

如需詳細資訊，請參閱 AWS Resource Explorer 使用者指南中的建立要搜尋的檢視。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateView。

delete-index

以下程式碼範例顯示如何使用 `delete-index`。

AWS CLI

刪除 AWS 區域索引以關閉 區域中的 Resource Explorer

下列delete-index範例會刪除您提出請求之 AWS 區域中指定的 Resource Explorer 索引。

```
aws resource-explorer-2 delete-index \
    --arn arn:aws:resource-explorer-2:us-west-2:123456789012:index/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222 \
    --region us-west-2
```

輸出：

```
{
    "Arn": "arn:aws:resource-explorer-2:us-west-2:123456789012:index/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222",
    "State": "DELETING"
}
```

如需刪除索引的詳細資訊，請參閱《AWS Resource Explorer 使用者指南》中的在 AWS 區域中關閉 Resource Explorer。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteIndex。

delete-view

以下程式碼範例顯示如何使用 delete-view。

AWS CLI

刪除資源總管檢視

下列delete-view範例會刪除其 ARN 指定的檢視。

```
aws resource-explorer-2 delete-view \
    --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

輸出：

```
{
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}
```

如需詳細資訊，請參閱 AWS Resource Explorer 使用者指南中的刪除檢視。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteView。

disassociate-default-view

以下程式碼範例顯示如何使用 disassociate-default-view。

AWS CLI

移除 AWS 區域的預設 Resource Explorer 檢視

以下內容會disassociate-default-view移除您呼叫 操作之 AWS 區域的預設 Resource Explorer 檢視。執行此操作後，區域中的所有搜尋操作都必須明確指定檢視，否則操作會失敗。

```
aws resource-explorer-2 disassociate-default-view
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Resource Explorer 使用者指南中的在 AWS 區域中設定預設檢視。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateDefaultView。

get-default-view

以下程式碼範例顯示如何使用 get-default-view。

AWS CLI

擷取資源總管檢視，這是其 AWS 區域的預設檢視

下列get-default-view範例會擷取檢視的 ARN，該檢視是您呼叫 操作之 AWS 區域的預設值。

```
aws resource-explorer-2 get-default-view
```

輸出：

```
{
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/default-
view/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
```

```
}
```

如需詳細資訊，請參閱 AWS Resource Explorer 使用者指南中的在 AWS 區域中設定預設檢視。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDefaultView。

get-index

以下程式碼範例顯示如何使用 get-index。

AWS CLI

範例 1：擷取 Resource Explorer 彙總工具索引的詳細資訊

下列get-index範例顯示指定 AWS 區域中 Resource Explorer 索引的詳細資訊。由於指定的區域包含帳戶的彙總器索引，因此輸出會列出將資料複寫至此區域的索引的區域。

```
aws resource-explorer-2 get-index \
    --region us-east-1
```

輸出：

```
{
    "Arn": "arn:aws:resource-explorer-2:us-east-1:123456789012:index/EXAMPLE8-90ab-
cdef-fedc-EXAMPLE11111",
    "CreatedAt": "2022-07-12T18:59:10.503000+00:00",
    "LastUpdatedAt": "2022-07-13T18:41:58.799000+00:00",
    "ReplicatingFrom": [
        "ap-south-1",
        "us-west-2"
    ],
    "State": "ACTIVE",
    "Tags": {},
    "Type": "AGGREGATOR"
}
```

範例 2：擷取 Resource Explorer 本機索引的詳細資訊

下列get-index範例顯示指定 AWS 區域中 Resource Explorer 索引的詳細資訊。由於指定的區域包含本機索引，輸出會列出從此區域的索引複寫資料的 區域。

```
aws resource-explorer-2 get-index \
```

```
    --region us-west-2
```

輸出：

```
{
    "Arn": "arn:aws:resource-explorer-2:us-west-2:123456789012:index/EXAMPLE8-90ab-
cdef-fedc-EXAMPLE22222",
    "CreatedAt": "2022-07-12T18:59:10.503000+00:00",
    "LastUpdatedAt": "2022-07-13T18:41:58.799000+00:00",
    "ReplicatingTo": [
        "us-west-2"
    ],
    "State": "ACTIVE",
    "Tags": {},
    "Type": "LOCAL"
}
```

如需索引的詳細資訊，請參閱《Resource Explorer 使用者指南》中的檢查已開啟 Resource Explorer AWSAWS 的區域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIndex。

get-view

以下程式碼範例顯示如何使用 get-view。

AWS CLI

擷取 Resource Explorer 檢視的詳細資訊

下列get-view範例顯示其 ARN 所指定檢視的詳細資訊。

```
aws resource-explorer-2 get-view \
    --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

輸出：

```
{
  "Tags" : {},
  "View" : {
```

```
        "Filters" : {
            "FilterString" : "service:ec2"
        },
        "IncludedProperties" : [
            {
                "Name" : "tags"
            }
        ],
        "LastUpdatedAt" : "2022-07-13T21:33:45.249Z",
        "Owner" : "123456789012",
        "Scope" : "arn:aws:iam::123456789012:root",
        "ViewArn" : "arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-
Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
  }
}
```

如需檢視的詳細資訊，請參閱《Resource Explorer 使用者指南》中的關於 AWS Resource Explorer 檢視。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetView。

`list-indexes`

以下程式碼範例顯示如何使用 `list-indexes`。

AWS CLI

列出 Resource Explorer 具有索引 AWS 的區域

下列`list-indexes`範例列出 Resource Explorer 具有索引的所有區域的索引。回應會指定每個索引的類型、其 AWS 區域及其 ARN。

```
aws resource-explorer-2 list-indexes
```

輸出：

```
{
    "Indexes": [
        {
            "Arn": "arn:aws:resource-explorer-2:us-west-2:123456789012:index/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111",
```

```
            "Region": "us-west-2",
            "Type": "AGGREGATOR"
        },
        {
            "Arn": "arn:aws:resource-explorer-2:us-east-1:123456789012:index/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222",
            "Region": "us-east-1",
            "Type": "LOCAL"
        },
        {
            "Arn": "arn:aws:resource-explorer-2:us-east-2:123456789012:index/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE33333",
            "Region": "us-east-2",
            "Type": "LOCAL"
        },
        {
            "Arn": "arn:aws:resource-explorer-2:us-west-1:123456789012:index/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE44444",
            "Region": "us-west-1",
            "Type": "LOCAL"
        }
    ]
}
```

如需索引的詳細資訊，請參閱《Resource Explorer 使用者指南》中的檢查已開啟 Resource Explorer AWSAWS 的區域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListIndexes。

`list-supported-resource-types`

以下程式碼範例顯示如何使用 `list-supported-resource-types`。

AWS CLI

列出 Resource Explorer 具有索引 AWS 的區域

下列`list-supported-resource-types`範例列出 &AREXlong; 目前支援的所有資源類型。範例回應包含一個`NextToken`值，表示有更多輸出可供其他呼叫擷取。

```
aws resource-explorer-2 list-supported-resource-types \
    --max-items 10
```

輸出：

```
{
    "ResourceTypes": [
        {
            "ResourceType": "cloudfront:cache-policy",
            "Service": "cloudfront"
        },
        {
            "ResourceType": "cloudfront:distribution",
            "Service": "cloudfront"
        },
        {
            "ResourceType": "cloudfront:function",
            "Service": "cloudfront"
        },
        {
            "ResourceType": "cloudfront:origin-access-identity",
            "Service": "cloudfront"
        },
        {
            "ResourceType": "cloudfront:origin-request-policy",
            "Service": "cloudfront"
        },
        {
            "ResourceType": "cloudfront:realtime-log-config",
            "Service": "cloudfront"
        },
        {
            "ResourceType": "cloudfront:response-headers-policy",
            "Service": "cloudfront"
        },
        {
            "ResourceType": "cloudwatch:alarm",
            "Service": "cloudwatch"
        },
        {
            "ResourceType": "cloudwatch:dashboard",
            "Service": "cloudwatch"
        },
        {
            "ResourceType": "cloudwatch:insight-rule",
            "Service": "cloudwatch"
        }
```

```
    ],
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMH0="
}
```

若要取得輸出的下一個部分，請再次呼叫 操作，並將上一個呼叫的NextToken回應值傳遞為 的值--starting-token。重複 ，直到 NextToken不存在回應為止。

```
aws resource-explorer-2 list-supported-resource-types \
    --max-items 10  \
    --starting-
token eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMH0=
```

輸出：

```
{
    "ResourceTypes": [
        {
            "ResourceType": "cloudwatch:metric-stream",
            "Service": "cloudwatch"
        },
        {
            "ResourceType": "dynamodb:table",
            "Service": "dynamodb"
        },
        {
            "ResourceType": "ec2:capacity-reservation",
            "Service": "ec2"
        },
        {
            "ResourceType": "ec2:capacity-reservation-fleet",
            "Service": "ec2"
        },
        {
            "ResourceType": "ec2:client-vpn-endpoint",
            "Service": "ec2"
        },
        {
            "ResourceType": "ec2:customer-gateway",
            "Service": "ec2"
        },
        {
            "ResourceType": "ec2:dedicated-host",
            "Service": "ec2"
```

```
        },
        {
            "ResourceType": "ec2:dhcp-options",
            "Service": "ec2"
        },
        {
            "ResourceType": "ec2:egress-only-internet-gateway",
            "Service": "ec2"
        },
        {
            "ResourceType": "ec2:elastic-gpu",
            "Service": "ec2"
        }
    ],
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyMH0="
}
```

如需索引的詳細資訊，請參閱《Resource Explorer 使用者指南》中的檢查已開啟 Resource Explorer AWSAWS 的區域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSupportedResourceTypes。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出連接至 Resource Explorer 檢視或索引的標籤

下列`list-tags-for-resource`範例列出附加至檢視的標籤索引鍵和值對，以及指定的 ARN。您必須從包含 資源 AWS 的區域呼叫 操作。

```
aws resource-explorer-2 list-tags-for-resource \
    --resource-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

輸出：

```
{
    "Tags": {
        "application": "MainCorpApp",
```

```
        "department": "1234"
    }
}
```

如需標記檢視的詳細資訊，請參閱《AWS Resource Explorer 使用者指南》中的標記存取控制的檢視。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`list-views`

以下程式碼範例顯示如何使用 `list-views`。

AWS CLI

列出 AWS 區域中可用的 Resource Explorer 檢視

下列`list-views`範例列出您叫用 操作的區域中所有可用的檢視。

```
aws resource-explorer-2 list-views
```

輸出：

```
{
    "Views": [
        "arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-View/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111",
        "arn:aws:resource-explorer-2:us-east-1:123456789012:view/Default-All-
Resources-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222",
        "arn:aws:resource-explorer-2:us-east-1:123456789012:view/Production-Only-
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE33333"
    ]
}
```

如需檢視的詳細資訊，請參閱《Resource Explorer 使用者指南》中的關於 AWS Resource Explorer 檢視。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListViews。

`search`

以下程式碼範例顯示如何使用 `search`。

AWS CLI

範例 1：使用預設檢視進行搜尋

下列search範例顯示指定 中與 服務相關聯的所有資源。搜尋會使用區域的預設檢視。範例回應包含一個NextToken值，表示有更多輸出可供其他呼叫擷取。

```
aws resource-explorer-2 search \
    --query-string "service:iam"
```

輸出：

```
{
    "Count": {
        "Complete": true,
        "TotalResources": 55
    },
    "NextToken":
 "AG9VOEF1KLEXAMPLEOhJHVwo5chEXAMPLER5XiEpNrgsEXAMPLE...b0CmOFOryHEXAMPLE",
    "Resources": [{
        "Arn": "arn:aws:iam::123456789012:policy/service-role/Some-Policy-For-A-
Service-Role",
        "LastReportedAt": "2022-07-21T12:34:42Z",
        "OwningAccountId": "123456789012",
        "Properties": [],
        "Region": "global",
        "ResourceType": "iam:policy",
        "Service": "iam"
    }, {
        "Arn": "arn:aws:iam::123456789012:policy/service-role/Another-Policy-For-A-
Service-Role",
        "LastReportedAt": "2022-07-21T12:34:42Z",
        "OwningAccountId": "123456789012",
        "Properties": [],
        "Region": "global",
        "ResourceType": "iam:policy",
        "Service": "iam"
    }, {
       ... TRUNCATED FOR BREVITY ...
    }],
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/my-default-
view/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}
```

範例 2：使用指定的檢視進行搜尋

下列search範例搜尋會顯示指定區域中透過指定檢視 AWS 可見的所有資源 ("*")。由於附加至檢視的篩選條件，結果僅包含與 Amazon EC2 相關聯的資源。

```
aws resource-explorer-2 search \
    -- query-string "*" \
    -- view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-view/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222
```

輸出：

```
HTTP/1.1 200 OK
Date: Tue, 01 Nov 2022 20:00:59 GMT
Content-Type: application/json
Content-Length: <PayloadSizeBytes>

    {
        "Count": {
            "Complete": true,
            "TotalResources": 67
        },
        "Resources": [{
            "Arn": "arn:aws:ec2:us-east-1:123456789012:network-acl/acl-1a2b3c4d",
            "LastReportedAt": "2022-07-21T18:52:02Z",
            "OwningAccountId": "123456789012",
            "Properties": [{
                "Data": [{
                    "Key": "Department",
                    "Value": "AppDevelopment"
                }, {
                    "Key": "Environment",
                    "Value": "Production"
                }],
                "LastReportedAt": "2021-11-15T14:48:29Z",
                "Name": "tags"
            }],
            "Region": "us-east-1",
            "ResourceType": "ec2:network-acl",
            "Service": "ec2"
        }, {
            "Arn": "arn:aws:ec2:us-east-1:123456789012:subnet/subnet-1a2b3c4d",
            "LastReportedAt": "2022-07-21T21:22:23Z",
```

```
            "OwningAccountId": "123456789012",
            "Properties": [{
                "Data": [{
                    "Key": "Department",
                    "Value": "AppDevelopment"
                }, {
                    "Key": "Environment",
                    "Value": "Production"
                }],
                "LastReportedAt": "2021-07-29T19:02:39Z",
                "Name": "tags"
            }],
            "Region": "us-east-1",
            "ResourceType": "ec2:subnet",
            "Service": "ec2"
        }, {
            "Arn": "arn:aws:ec2:us-east-1:123456789012:dhcp-options/dopt-1a2b3c4d",
            "LastReportedAt": "2022-07-21T06:08:53Z",
            "OwningAccountId": "123456789012",
            "Properties": [{
                "Data": [{
                    "Key": "Department",
                    "Value": "AppDevelopment"
                }, {
                    "Key": "Environment",
                    "Value": "Production"
                }],
                "LastReportedAt": "2021-11-15T15:11:05Z",
                "Name": "tags"
            }],
            "Region": "us-east-1",
            "ResourceType": "ec2:dhcpoptions",
            "Service": "ec2"
        }, {
            ... TRUNCATED FOR BREVITY ...
        }],
        "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-
view/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222"
    }
```

如需詳細資訊，請參閱《AWS 資源總管使用者指南》中的使用資源總管搜尋資源。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的搜尋。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

標記資源總管檢視

下列tag-resource範例會將值為「生產」的標籤金鑰「環境」新增至具有指定 ARN 的檢視。

```
aws resource-explorer-2 tag-resource \
    --resource-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View//
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111 \
    --tags environment=production
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Resource Explorer 使用者指南中的標記檢視以進行存取控制。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從 Resource Explorer 檢視中移除標籤

下列untag-resource範例會從具有指定 ARN 的檢視中移除金鑰名稱為「環境」的任何標籤。

```
aws resource-explorer-2 untag-resource \
    --resource-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View//
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111 \
    --tag-keys environment
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Resource Explorer 使用者指南中的標記檢視以進行存取控制。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-index-type

以下程式碼範例顯示如何使用 update-index-type。

AWS CLI

變更 Resource Explorer 索引的類型

下列update-index-type範例會將指定的索引從 類型轉換為 local 類型aggregator，以開啟搜尋帳戶中所有 AWS 區域之資源的功能。您必須將請求傳送至包含您要更新之索引 AWS 的區域。

```
aws resource-explorer-2 update-index-type \
    --arn arn:aws:resource-explorer-2:us-east-1:123456789012:index/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111 \
    --type aggregator \
    --region us-east-1
```

輸出：

```
{
    "Arn":"arn:aws:resource-explorer-2:us-east-1:123456789012:index/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111",
    "LastUpdatedAt":"2022-07-13T18:41:58.799Z",
    "State":"updating",
    "Type":"aggregator"
}
```

如需變更索引類型的詳細資訊，請參閱《AWS Resource Explorer 使用者指南》中的透過建立彙總索引開啟跨區域搜尋。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateIndexType。

update-view

以下程式碼範例顯示如何使用 update-view。

AWS CLI

範例 1：更新 Resource Explorer 檢視的 IncludedProperties 欄位

下列update-view範例透過將 `tags`新增至選用的 來更新指定的檢視`IncludedProperties`。執行此操作後，使用此檢視的搜尋操作包含連接到結果中顯示之資源的標籤的相關資訊。

```
aws resource-explorer-2 update-view \
    --included-properties Name=tags \
    --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222
```

輸出：

```
{
    "View": {
        "Filters": {
            "FilterString": ""
        },
        "IncludedProperties": [
            {
                "Name": "tags"
            }
        ],
        "LastUpdatedAt": "2022-07-19T17:41:21.710000+00:00",
        "Owner": "123456789012",
        "Scope": "arn:aws:iam::123456789012:root",
        "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-
Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
    }
}
```

範例 2：更新連接到檢視的篩選條件

下列update-view範例會更新指定的檢視，以使用篩選條件，將結果限制為僅與 Amazon EC2 服務相關聯的資源類型。

```
aws resource-explorer-2 update-view \
    --filters FilterString="service:ec2" \
    --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222
```

輸出：

```
{
```

```
    "View": {
    "Filters": {
        "FilterString": "service:ec2"
    },
    "IncludedProperties": [],
    "LastUpdatedAt": "2022-07-19T17:41:21.710000+00:00",
        "Owner": "123456789012",
        "Scope": "arn:aws:iam::123456789012:root",
        "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222"
    }
}
```

如需檢視的詳細資訊，請參閱 Resource Explorer 使用者指南中的關於 AWS Resource Explorer 檢視。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateView。

使用 的資源群組範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 資源群組來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-group`

以下程式碼範例顯示如何使用 `create-group`。

AWS CLI

範例 1：建立標籤型資源群組

下列create-group範例會在目前區域中建立 Amazon EC2 執行個體的標籤型資源群組。它基於使用索引鍵 Name和值 標記之資源的查詢WebServers。群組名稱為 tbq-WebServer。查詢位於傳遞給 命令的個別 JSON 檔案中。

```
aws resource-groups create-group \
    --name tbq-WebServer \
    --resource-query file://query.json
```

query.json 的內容：

```
{
    "Type": "TAG_FILTERS_1_0",
    "Query": "{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],\"TagFilters\":
[{\"Key\":\"Name\", \"Values\":[\"WebServers\"]}]}"
}
```

輸出：

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-
WebServer",
        "Name": "tbq-WebServer"
    },
    "ResourceQuery": {
        "Type": "TAG_FILTERS_1_0",
        "Query": "{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],\"TagFilters\":
[{\"Key\":\"Name\", \"Values\":[\"WebServers\"]}]}"
    }
}
```

範例 2：建立 CloudFormation 堆疊型資源群組

下列create-group範例會建立名為 的以 AWS CloudFormation 堆疊為基礎的資源群組sampleCFNstackgroup。查詢包含資源群組所支援之指定 CloudFormation 堆疊中的所有 AWS 資源。

```
aws resource-groups create-group \
    --name cbq-CFNstackgroup \
    --resource-query file://query.json
```

query.json 的內容：

```
{
    "Type": "CLOUDFORMATION_STACK_1_0",
    "Query": "{\"ResourceTypeFilters\":[\"AWS::AllSupported\"],\"StackIdentifier
\":\"arn:aws:cloudformation:us-west-2:123456789012:stack/MyCFNStack/1415z9z0-
z39z-11z8-97z5-500z212zz6fz\"}"
}
```

輸出：

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/cbq-
CFNstackgroup",
        "Name": "cbq-CFNstackgroup"
    },
    "ResourceQuery": {
        "Type": "CLOUDFORMATION_STACK_1_0",
        "Query": "{\"ResourceTypeFilters\":[\"AWS::AllSupported\"],\"StackIdentifier
\":\"arn:aws:cloudformation:us-east-2:123456789012:stack/MyCFNStack/1415z9z0-
z39z-11z8-97z5-500z212zz6fz\"}"}'
    }
}
```

如需詳細資訊，請參閱《資源群組使用者指南》中的建立群組。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateGroup。

delete-group

以下程式碼範例顯示如何使用 delete-group。

AWS CLI

更新資源群組的描述

下列delete-group範例會更新指定的資源群組。

```
aws resource-groups delete-group \
    --group-name tbq-WebServer
```

輸出：

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:us-west-2:1234567890:group/tbq-
WebServer",
        "Name": "tbq-WebServer"
    }
}
```

如需詳細資訊，請參閱《資源群組使用者指南》中的刪除群組。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteGroup。

get-group-query

以下程式碼範例顯示如何使用 get-group-query。

AWS CLI

取得連接至資源群組的查詢

下列get-group-query範例顯示連接至指定資源群組的查詢。

```
aws resource-groups get-group-query \
    --group-name tbq-WebServer
```

輸出：

```
{
    "GroupQuery": {
        "GroupName": "tbq-WebServer",
        "ResourceQuery": {
            "Type": "TAG_FILTERS_1_0",
            "Query": "{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],\"TagFilters
\":[{\"Key\":\"Name\", \"Values\":[\"WebServers\"]}]}"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetGroupQuery。

get-group

以下程式碼範例顯示如何使用 get-group。

AWS CLI

取得資源群組的相關資訊

下列get-group範例顯示指定資源群組的詳細資訊。若要取得連接至 群組的查詢，請使用 get-group-query。

```
aws resource-groups get-group \
    --group-name tbq-WebServer
```

輸出：

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-
WebServer",
        "Name": "tbq-WebServer",
        "Description": "A tag-based query resource group of WebServers."
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetGroup。

get-tags

以下程式碼範例顯示如何使用 get-tags。

AWS CLI

擷取連接至資源群組的標籤

下列get-tags範例顯示連接至指定資源群組 （群組本身，而非其成員） 的標籤金鑰和值對。

```
aws resource-groups get-tags \
    --arn arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer
```

輸出：

```
{
    "Arn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",
    "Tags": {
        "QueryType": "tags",
        "QueryResources": "ec2-instances"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTags。

`list-group-resources`

以下程式碼範例顯示如何使用 `list-group-resources`。

AWS CLI

列出資源群組中的所有資源

範例 1：下列`list-resource-groups`範例列出屬於指定資源群組的所有資源。

```
aws resource-groups list-group-resources \
    --group-name tbq-WebServer
```

輸出：

```
{
    "ResourceIdentifiers": [
        {
            "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-09f77fa38c12345ab",
            "ResourceType": "AWS::EC2::Instance"
        }
    ]
}
```

範例 2：下列範例會列出群組中也具有 'AWS：：EC2：：Instance' 之 'resource-type' 的所有資源。：

aws resource-groups list-group-resources --group-name tbq-WebServer --filters Name=resource-type，Values=AWS：：EC2：：Instance

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListGroupResources。

list-groups

以下程式碼範例顯示如何使用 list-groups。

AWS CLI

列出可用的資源群組

下列list-groups範例顯示所有資源群組的清單。

```
aws resource-groups list-groups
```

輸出：

```
{
    "GroupIdentifiers": [
        {
            "GroupName": "tbq-WebServer",
            "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-
WebServer3"
        },
        {
            "GroupName": "cbq-CFNStackQuery",
            "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/cbq-
CFNStackQuery"
        }
    ],
    "Groups": [
        {
            "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-
WebServer",
            "Name": "tbq-WebServer"
        },
        {
            "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/cbq-
CFNStackQuery",
            "Name": "cbq-CFNStackQuery"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListGroups。

list-resource-groups

以下程式碼範例顯示如何使用 list-resource-groups。

AWS CLI

列出資源群組中的所有資源

下列list-resource-groups範例列出屬於指定資源群組的所有資源。

```
aws resource-groups list-group-resources \
    --group-name tbq-WebServer
```

輸出：

```
{
    "ResourceIdentifiers": [
        {
            "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-09f77fa38c12345ab",
            "ResourceType": "AWS::EC2::Instance"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourceGroups。

put-group-configuration

以下程式碼範例顯示如何使用 put-group-configuration。

AWS CLI

將服務組態連接至資源群組

範例 1：下列put-group-configuration範例指定資源群組僅包含 C5或 M5 系列執行個體的 Amazon EC2 容量保留。

```
aws resource-groups put-group-configuration \
```

```
    --group MyTestGroup \
    --configuration file://config.json
```

config.json 的內容：

```
[
    {
        "Type": "AWS::EC2::HostManagement",
        "Parameters": [
            {
                "Name": "allowed-host-families",
                "Values": [ "c5", "m5" ]
            },
                {
                    "Name": "any-host-based-license-configuration",
                    "Values": [ "true" ]
                }
        ]
    },
    {
        "Type": "AWS::ResourceGroups::Generic",
        "Parameters": [
            {
                "Name": "allowed-resource-types",
                "Values": [ "AWS::EC2::Host" ]
            },
            {
                "Name": "deletion-protection",
                "Values": [ "UNLESS_EMPTY" ]
            }
        ]
    }
]
```

如果成功，此命令不會產生輸出。

如需詳細資訊，請參閱《資源群組 API 參考指南》中的資源群組的服務組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutGroupConfiguration。

search-resources

以下程式碼範例顯示如何使用 search-resources。

AWS CLI

尋找符合查詢的資源

下列search-resources範例會擷取符合指定查詢的所有 AWS 資源清單。

```
aws resource-groups search-resources \
    --resource-query file://query.json
```

query.json 的內容：

```
{
    "Type": "TAG_FILTERS_1_0",
    "Query": "{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],\"TagFilters\":
[{\"Key\":\"Patch Group\", \"Values\":[\"Dev\"]}]}"
}
```

輸出：

```
{
    "ResourceIdentifiers": [
        {
            "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-01a23bc45d67890ef",
            "ResourceType": "AWS::EC2::Instance"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchResources。

tag

以下程式碼範例顯示如何使用 tag。

AWS CLI

將標籤連接至資源群組

下列tag範例會將指定的標籤索引鍵和值對連接至指定的資源群組 （ 群組本身，而非其成員 ）。

```
aws resource-groups tag \
```

```
    --tags QueryType=tags,QueryResources=ec2-instances \
    --arn arn:aws:resource-groups:us-west-2:128716708097:group/tbq-WebServer
```

輸出：

```
{
    "Arn": "arn:aws:resource-groups:us-west-2:128716708097:group/tbq-WebServer",
    "Tags": {
        "QueryType": "tags",
        "QueryResources": "ec2-instances"
    }
}
```

如需詳細資訊，請參閱AWS 資源群組使用者指南中的管理標籤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的標籤。

untag

以下程式碼範例顯示如何使用 untag。

AWS CLI

從資源群組移除標籤

下列untags範例會從資源群組本身移除具有指定金鑰的任何標籤，而不是其成員。

```
aws resource-groups untag \
    --arn arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer \
    --keys QueryType
```

輸出：

```
{
    "Arn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",
    "Keys": [
        "QueryType"
    ]
}
```

如需詳細資訊，請參閱AWS 資源群組使用者指南中的管理標籤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的取消標記。

update-group-query

以下程式碼範例顯示如何使用 update-group-query。

AWS CLI

範例 1：更新標籤型資源群組的查詢

下列update-group-query範例會更新連接至指定標籤型資源群組的查詢。

```
aws resource-groups update-group-query \
    --group-name tbq-WebServer \
    --resource-query '{"Type":"TAG_FILTERS_1_0", "Query":"{\"ResourceTypeFilters\":
[\"AWS::EC2::Instance\"],\"TagFilters\":[{\"Key\":\"Name\", \"Values\":[\"WebServers
\"]}]}"}'
```

輸出：

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:us-east-2:123456789012:group/tbq-
WebServer",
        "Name": "tbq-WebServer"
    },
    "ResourceQuery": {
        "Type": "TAG_FILTERS_1_0",
        "Query": "{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],\"TagFilters\":
[{\"Key\":\"Name\", \"Values\":[\"WebServers\"]}]}"
    }
}
```

如需詳細資訊，請參閱《資源群組使用者指南》中的更新群組。 AWS

範例 2：更新 CloudFormation 堆疊型資源群組的查詢

下列update-group-query範例會更新連接至指定 AWS CloudFormation 堆疊型資源群組的查詢。

```
aws resource-groups update-group-query \
    --group-name cbq-CFNstackgroup \
    --resource-query '{"Type": "CLOUDFORMATION_STACK_1_0", "Query":
 "{\"ResourceTypeFilters\":[\"AWS::AllSupported\"],\"StackIdentifier\":
```

```
\"arn:aws:cloudformation:us-west-2:123456789012:stack/MyCFNStack/1415z9z0-
z39z-11z8-97z5-500z212zz6fz\"}"}'
```

輸出：

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/cbq-
CFNstackgroup",
        "Name": "cbq-CFNstackgroup"
    },
    "ResourceQuery": {
        "Type": "CLOUDFORMATION_STACK_1_0",
        "Query": "{\"ResourceTypeFilters\":[\"AWS::AllSupported\"],\"StackIdentifier
\":\"arn:aws:cloudformation:us-west-2:123456789012:stack/MyCFNStack/1415z9z0-
z39z-11z8-97z5-500z212zz6fz\"}"
    }
}
```

如需詳細資訊，請參閱《資源群組使用者指南》中的更新群組。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateGroupQuery。

update-group

以下程式碼範例顯示如何使用 update-group。

AWS CLI

更新資源群組的描述

下列update-group範例會更新指定資源群組的描述。

```
aws resource-groups update-group \
    --group-name tbq-WebServer \
    --description "Resource group for all web server resources."
```

輸出：

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-
WebServer",
```

```
        "Name": "tbq-WebServer"
        "Description": "Resource group for all web server resources."
    }
}
```

如需詳細資訊，請參閱《資源群組使用者指南》中的更新群組。 AWS

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UpdateGroup。

資源群組使用 標記 API 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配資源群組標記 API 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`get-resources`

以下程式碼範例顯示如何使用 `get-resources`。

AWS CLI

取得已標記資源的清單

下列`get-resources`範例顯示帳戶中以指定金鑰名稱和值標記的資源清單。

```
aws resourcegroupstaggingapi get-resources \
    --tag-filters Key=Environment,Values=Production \
    --tags-per-page 100
```

輸出：

```
{
```

```
    "ResourceTagMappingList": [
        {
            "ResourceARN": " arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX/template/0-7sbz2Kz0",
            "Tags": [
                {
                    "Key": "Environment",
                    "Value": "Production"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《資源群組標記 API 參考》中的 GetResources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResources。

get-tag-keys

以下程式碼範例顯示如何使用 get-tag-keys。

AWS CLI

取得所有標籤索引鍵的清單

下列get-tag-keys範例會擷取帳戶中資源使用的所有標籤金鑰名稱清單。

```
aws resourcegroupstaggingapi get-tag-keys
```

輸出：

```
{
    "TagKeys": [
        "Environment",
        "CostCenter",
        "Department"
    ]
}
```

如需詳細資訊，請參閱資源群組標記 API 參考中的 GetTagKeys。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTagKeys。

get-tag-values

以下程式碼範例顯示如何使用 get-tag-values。

AWS CLI

取得所有標籤值的清單

下列get-tag-values範例顯示 中所有資源的指定金鑰所使用的所有值

```
aws resourcegroupstaggingapi get-tag-values \
    --key=Environment
```

輸出：

```
{
    "TagValues": [
        "Alpha",
        "Gamma",
        "Production"
    ]
}
```

如需詳細資訊，請參閱《資源群組標記 API 參考》中的 GetTagValues。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTagValues。

tag-resources

以下程式碼範例顯示如何使用 tag-resources。

AWS CLI

將標籤連接至資源

下列tag-resources範例會使用金鑰名稱和值來標記指定的資源。

```
aws resourcegroupstaggingapi tag-resources \
    --resource-arn-list arn:aws:s3:::MyProductionBucket \
    --tags Environment=Production,CostCenter=1234
```

輸出：

```
{
    "FailedResourcesMap": {}
}
```

如需詳細資訊，請參閱 資源群組標記 API 參考中的 TagResources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TagResources。

untag-resources

以下程式碼範例顯示如何使用 untag-resources。

AWS CLI

從資源移除標籤

下列untag-resources範例會從指定的資源移除指定的標籤索引鍵和任何相關聯的值。

```
aws resourcegroupstaggingapi untag-resources \
    --resource-arn-list arn:aws:s3:::amzn-s3-demo-bucket \
    --tag-keys Environment CostCenter
```

輸出：

```
{
    "FailedResourcesMap": {}
}
```

如需詳細資訊，請參閱資源群組標記 API 參考中的 UntagResources。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UntagResources。

AWS RoboMaker 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS RoboMaker。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

batch-describe-simulation-job

以下程式碼範例顯示如何使用 batch-describe-simulation-job。

AWS CLI

批次描述模擬任務

下列batch-describe-simulation-job範例會擷取三個指定模擬任務的詳細資訊。

命令：

```
aws robomaker batch-describe-simulation-job \
--job arn:aws:robomaker:us-west-2:111111111111:simulation-job/
sim-66bbb3gpxm8x arn:aws:robomaker:us-west-2:111111111111:simulation-job/
sim-p0cpdrrwng2n arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
g8h6tglmblgw
```

輸出：

```
{
    "jobs": [
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/
sim-66bbb3gpxm8x",
            "status": "Completed",
            "lastUpdatedAt": 1548959178.0,
            "failureBehavior": "Continue",
            "clientRequestToken": "6020408e-b05c-4310-9f13-4ed71c5221ed",
            "outputLocation": {
                "s3Bucket": "awsrobomakerobjecttracker-111111111-
bundlesbucket-2lk584kiq1oa",
                "s3Prefix": "output"
            },
            "maxJobDurationInSeconds": 3600,
            "simulationTimeMillis": 0,
```

```
            "iamRole": "arn:aws:iam::111111111111:role/
AWSRoboMakerObjectTracker-154895-SimulationJobRole-14D5ASA7PQE3A",
            "simulationApplications": [
                {
                    "application": "arn:aws:robomaker:us-
west-2:111111111111:simulation-application/
AWSRoboMakerObjectTracker-1548959046124_NPvyfcatq/1548959170096",
                    "applicationVersion": "$LATEST",
                    "launchConfig": {
                        "packageName": "object_tracker_simulation",
                        "launchFile": "local_training.launch",
                        "environmentVariables": {
                            "MARKOV_PRESET_FILE": "object_tracker.py",
                            "MODEL_S3_BUCKET": "awsrobomakerobjecttracker-111111111-
bundlesbucket-2lk584kiq1oa",
                            "MODEL_S3_PREFIX": "model-store",
                            "ROS_AWS_REGION": "us-west-2"
                        }
                    }
                }
            ],
            "tags": {},
            "vpcConfig": {
                "subnets": [
                    "subnet-716dd52a",
                    "subnet-43c22325",
                    "subnet-3f526976"
                ],
                "securityGroups": [
                    "sg-3fb40545"
                ],
                "vpcId": "vpc-99895eff",
                "assignPublicIp": true
            }
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
p0cpdrrwng2n",
            "status": "Completed",
            "lastUpdatedAt": 1548168817.0,
            "failureBehavior": "Continue",
            "clientRequestToken": "e4a23e75-f9a7-411d-835f-21881c82c58b",
            "outputLocation": {
```

```
                "s3Bucket": "awsrobomakercloudwatch-111111111111-
bundlesbucket-14e5s9jvwtmv7",
                "s3Prefix": "output"
            },
            "maxJobDurationInSeconds": 3600,
            "simulationTimeMillis": 0,
            "iamRole": "arn:aws:iam::111111111111:role/
AWSRoboMakerCloudWatch-154766341-SimulationJobRole-G0OBWTQ8YBG6",
            "robotApplications": [
                {
                    "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/AWSRoboMakerCloudWatch-1547663411642_NZbpqEJ3T/1547663517377",
                    "applicationVersion": "$LATEST",
                    "launchConfig": {
                        "packageName": "cloudwatch_robot",
                        "launchFile": "await_commands.launch",
                        "environmentVariables": {
                            "LAUNCH_ID": "1548168752173",
                            "ROS_AWS_REGION": "us-west-2"
                        }
                    }
                }
            ],
            "simulationApplications": [
                {
                    "application": "arn:aws:robomaker:us-
west-2:111111111111:simulation-application/
AWSRoboMakerCloudWatch-1547663411642_0LIt6D1h6/1547663521470",
                    "applicationVersion": "$LATEST",
                    "launchConfig": {
                        "packageName": "cloudwatch_simulation",
                        "launchFile": "bookstore_turtlebot_navigation.launch",
                        "environmentVariables": {
                            "LAUNCH_ID": "1548168752173",
                            "ROS_AWS_REGION": "us-west-2",
                            "TURTLEBOT3_MODEL": "waffle_pi"
                        }
                    }
                }
            ],
            "tags": {},
            "vpcConfig": {
                "subnets": [
                    "subnet-716dd52a",
```

```
                    "subnet-43c22325",
                    "subnet-3f526976"
                ],
                "securityGroups": [
                    "sg-3fb40545"
                ],
                "vpcId": "vpc-99895eff",
                "assignPublicIp": true
            }
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
g8h6tglmblgw",
            "status": "Canceled",
            "lastUpdatedAt": 1546543442.0,
            "failureBehavior": "Fail",
            "clientRequestToken": "d796bbb4-2a2c-1abc-f2a9-0d9e547d853f",
            "outputLocation": {
                "s3Bucket": "sample-bucket",
                "s3Prefix": "SimulationLog_115490482698"
            },
            "maxJobDurationInSeconds": 28800,
            "simulationTimeMillis": 0,
            "iamRole": "arn:aws:iam::111111111111:role/RoboMakerSampleTheFirst",
            "robotApplications": [
                {
                    "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/RoboMakerHelloWorldRobot/1546541208251",
                    "applicationVersion": "$LATEST",
                    "launchConfig": {
                        "packageName": "hello_world_robot",
                        "launchFile": "rotate.launch"
                    }
                }
            ],
            "simulationApplications": [
                {
                    "application": "arn:aws:robomaker:us-
west-2:111111111111:simulation-application/
RoboMakerHelloWorldSimulation/1546541198985",
                    "applicationVersion": "$LATEST",
                    "launchConfig": {
                        "packageName": "hello_world_simulation",
                        "launchFile": "empty_world.launch"
```

```
                    }
                }
            ],
            "tags": {}
        }
    ],
    "unprocessedJobs": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDescribeSimulationJob。

cancel-simulation-job

以下程式碼範例顯示如何使用 cancel-simulation-job。

AWS CLI

取消模擬任務

下列cancel-simulation-job範例會取消指定的模擬任務。

```
aws robomaker cancel-simulation-job \
    --job arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-66bbb3gpxm8x
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelSimulationJob。

create-deployment-job

以下程式碼範例顯示如何使用 create-deployment-job。

AWS CLI

建立部署任務

此範例會建立機群 MyFleet 的部署任務。它包含名為「ENVIRONMENT」的環境變數。它也會連接名為 "Region" 的標籤。

命令：

```
aws robomaker create-deployment-job --deployment-
config concurrentDeploymentPercentage=20,failureThresholdPercentage=25
```

```
 --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
Trek/1539894765711 --tags Region=West --deployment-application-
configs application=arn:aws:robomaker:us-west-2:111111111111:robot-application/
RoboMakerVoiceInteractionRobot/1546537110575,applicationVersion=1,launchConfig={environmentV
```

輸出：

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/sim-0974h36s4v0t",
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1539894765711",
  "status": "Pending",
  "deploymentApplicationConfigs": [
      {
          "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/RoboMakerVoiceInteractionRobot/1546537110575",
          "applicationVersion": "1",
          "launchConfig": {
              "packageName": "voice_interaction_robot",
              "launchFile": "await_commands.launch",
              "environmentVariables": {
                  "ENVIRONMENT": "Beta"
              }
          }
      }
  ],
  "createdAt": 1550770236.0,
  "deploymentConfig": {
      "concurrentDeploymentPercentage": 20,
      "failureThresholdPercentage": 25
  },
  "tags": {
      "Region": "West"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDeploymentJob。

create-fleet

以下程式碼範例顯示如何使用 create-fleet。

AWS CLI

建立機群

此範例會建立機群。它會連接名為 區域的標籤。

命令：

```
aws robomaker create-fleet --name MyFleet --tags Region=East
```

輸出：

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyOtherFleet/1550771394395",
  "name": "MyFleet",
  "createdAt": 1550771394.0,
  "tags": {
      "Region": "East"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFleet。

create-robot-application-version

以下程式碼範例顯示如何使用 create-robot-application-version。

AWS CLI

建立機器人應用程式版本

此範例會建立機器人應用程式版本。

命令：

```
aws robomaker create-robot-application-version --application arn:aws:robomaker:us-
west-2:111111111111:robot-application/MyRobotApplication/1551201873931
```

輸出：

```
{
```

```
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
MyRobotApplication/1551201873931",
  "name": "MyRobotApplication",
  "version": "1",
  "sources": [
      {
          "s3Bucket": "amzn-s3-demo-bucket",
          "s3Key": "my-robot-application.tar.gz",
          "etag": "f8cf5526f1c6e7b3a72c3ed3f79c5493-70",
          "architecture": "ARMHF"
      }
  ],
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
  },
  "lastUpdatedAt": 1551201873.0,
  "revisionId": "9986bb8d-a695-4ab4-8810-9f4a74d1aa00"
  "tags": {}
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRobotApplicationVersion。

create-robot-application

以下程式碼範例顯示如何使用 create-robot-application。

AWS CLI

建立機器人應用程式

此範例會建立機器人應用程式。

命令：

```
aws robomaker create-robot-application --name MyRobotApplication
 --sources s3Bucket=amzn-s3-demo-bucket,s3Key=my-robot-
application.tar.gz,architecture=X86_64 --robot-software-
suite name=ROS,version=Kinetic
```

輸出：

```
{
```

```
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
MyRobotApplication/1551201873931",
  "name": "MyRobotApplication",
  "version": "$LATEST",
  "sources": [
      {
          "s3Bucket": "amzn-s3-demo-bucket",
          "s3Key": "my-robot-application.tar.gz",
          "architecture": "ARMHF"
      }
  ],
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
  },
  "lastUpdatedAt": 1551201873.0,
  "revisionId": "1f3cb539-9239-4841-a656-d3efcffa07e1",
  "tags": {}
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRobotApplication。

create-robot

以下程式碼範例顯示如何使用 create-robot。

AWS CLI

建立機器人

此範例會建立機器人。它使用 ARMHF 架構。它也會連接名為 區域的標籤。

命令：

```
aws robomaker create-robot --name MyRobot --architecture ARMHF --greengrass-group-
id 0f728a3c-7dbf-4a3e-976d-d16a8360caba --tags Region=East
```

輸出：

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398",
  "name": "MyRobot",
```

```
  "createdAt": 1550772325.0,
  "greengrassGroupId": "0f728a3c-7dbf-4a3e-976d-d16a8360caba",
  "architecture": "ARMHF",
  "tags": {
      "Region": "East"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRobot。

create-simulation-application-version

以下程式碼範例顯示如何使用 create-simulation-application-version。

AWS CLI

建立模擬應用程式版本

此範例會建立機器人應用程式版本。

命令：

```
aws robomaker create-simulation-application-version --
application arn:aws:robomaker:us-west-2:111111111111:robot-application/
MySimulationApplication/1551203427605
```

輸出：

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
MyRobotApplication/1551203427605",
  "name": "MyRobotApplication",
  "version": "1",
  "sources": [
      {
          "s3Bucket": "amzn-s3-demo-bucket",
          "s3Key": "my-simulation-application.tar.gz",
          "etag": "00d8a94ff113856688c4fce618ae0f45-94",
          "architecture": "X86_64"
      }
  ],
  "simulationSoftwareSuite": {
```

```
        "name": "Gazebo",
        "version": "7"
    },
    "robotSoftwareSuite": {
        "name": "ROS",
        "version": "Kinetic"
    },
    "renderingEngine": {
        "name": "OGRE",
        "version": "1.x"
    },
   "lastUpdatedAt": 1551203853.0,
    "revisionId": "ee753e53-519c-4d37-895d-65e79bcd1914",
    "tags": {}
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSimulationApplicationVersion。

create-simulation-application

以下程式碼範例顯示如何使用 create-simulation-application。

AWS CLI

建立模擬應用程式

此範例會建立模擬應用程式。

命令：

```
aws robomaker create-simulation-application  --name MyRobotApplication
 --sources s3Bucket=amzn-s3-demo-bucket,s3Key=my-simulation-
application.tar.gz,architecture=ARMHF --robot-software-
suite name=ROS,version=Kinetic --simulation-software-suite name=Gazebo,version=7 --
rendering-engine name=OGRE,version=1.x
```

輸出：

```
{
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
MyRobotApplication/1551203301792",
    "name": "MyRobotApplication",
```

```
  "version": "$LATEST",
  "sources": [
      {
          "s3Bucket": "amzn-s3-demo-bucket",
          "s3Key": "my-simulation-application.tar.gz",
          "architecture": "X86_64"
      }
  ],
  "simulationSoftwareSuite": {
      "name": "Gazebo",
      "version": "7"
  },
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
  },
  "renderingEngine": {
      "name": "OGRE",
      "version": "1.x"
  },
  "lastUpdatedAt": 1551203301.0,
  "revisionId": "ee753e53-519c-4d37-895d-65e79bcd1914",
  "tags": {}
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSimulationApplication。

create-simulation-job

以下程式碼範例顯示如何使用 create-simulation-job。

AWS CLI

建立模擬任務

此範例會建立模擬任務。它使用機器人應用程式和模擬應用程式。

命令：

```
aws robomaker create-simulation-job --max-job-duration-
in-seconds 3600 --iam-role arn:aws:iam::111111111111:role/
AWSRoboMakerCloudWatch-154766341-SimulationJobRole-G0OBWTQ8YBG6 --robot-
applications application=arn:aws:robomaker:us-west-2:111111111111:robot-application/
```

```
MyRobotApplication/1551203485821,launchConfig={packageName=hello_world_robot,launchFile=rota
 --simulation-applications application=arn:aws:robomaker:us-
west-2:111111111111:simulation-application/
MySimulationApplication/1551203427605,launchConfig={packageName=hello_world_simulation,launc
 --tags Region=North
```

輸出：

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-w7m68wpr05h8",
  "status": "Pending",
  "lastUpdatedAt": 1551213837.0,
  "failureBehavior": "Fail",
  "clientRequestToken": "b283ccce-e468-43ee-8642-be76a9d69f15",
  "maxJobDurationInSeconds": 3600,
  "simulationTimeMillis": 0,
  "iamRole": "arn:aws:iam::111111111111:role/MySimulationRole",
  "robotApplications": [
      {
          "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/MyRobotApplication/1551203485821",
          "applicationVersion": "$LATEST",
          "launchConfig": {
              "packageName": "hello_world_robot",
              "launchFile": "rotate.launch"
          }
      }
  ],
  "simulationApplications": [
      {
          "application": "arn:aws:robomaker:us-west-2:111111111111:simulation-
application/MySimulationApplication/1551203427605",
          "applicationVersion": "$LATEST",
          "launchConfig": {
              "packageName": "hello_world_simulation",
              "launchFile": "empty_world.launch"
          }
      }
  ],
  "tags": {
      "Region": "North"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSimulationJob。

delete-fleet

以下程式碼範例顯示如何使用 delete-fleet。

AWS CLI

刪除機群

此範例會刪除機群。

命令：

```
aws robomaker delete-fleet --fleet arn:aws:robomaker:us-
west-2:111111111111:deployment-fleet/MyFleet/1550771394395
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFleet。

delete-robot-application

以下程式碼範例顯示如何使用 delete-robot-application。

AWS CLI

刪除機器人應用程式

此範例會刪除機器人應用程式。

命令：

```
aws robomaker delete-robot-application --application arn:aws:robomaker:us-
west-2:111111111111:robot-application/MyRobotApplication/1551203485821
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRobotApplication。

delete-robot

以下程式碼範例顯示如何使用 delete-robot。

AWS CLI

刪除機器人

此範例會刪除機器人。

命令：

```
aws robomaker delete-robot --robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1540829698778
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRobot。

delete-simulation-application

以下程式碼範例顯示如何使用 delete-simulation-application。

AWS CLI

刪除模擬應用程式

此範例會刪除模擬應用程式。

命令：

```
aws robomaker delete-simulation-application --application arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSimulationApplication。

deregister-robot

以下程式碼範例顯示如何使用 deregister-robot。

AWS CLI

從機群取消註冊機器人

此範例會從機群取消註冊機器人。

命令：

```
aws robomaker deregister-robot --fleet arn:aws:robomaker:us-
west-2:111111111111:deployment-fleet/MyFleet/1550771358907 --
robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398
```

輸出：

```
{
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1550771358907",
  "robot": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterRobot。

describe-deployment-job

以下程式碼範例顯示如何使用 describe-deployment-job。

AWS CLI

描述部署任務

下列describe-deployment-job範例會擷取指定部署任務的詳細資訊。

```
aws robomaker describe-deployment-job \
    --job arn:aws:robomaker:us-west-2:111111111111:deployment-job/deployment-
xl8qssl6pbcn
```

輸出：

```
{
    "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/deployment-
xl8qssl6pbcn",
    "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
Trek/1539894765711",
    "status": "InProgress",
    "deploymentConfig": {
        "concurrentDeploymentPercentage": 20,
        "failureThresholdPercentage": 25
    },
    "deploymentApplicationConfigs": [
```

```
        {
            "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/RoboMakerHelloWorldRobot/1546541208251",
            "applicationVersion": "1",
            "launchConfig": {
                "packageName": "hello_world_robot",
                "launchFile": "rotate.launch"
            }
        }
    ],
    "createdAt": 1551218369.0,
    "robotDeploymentSummary": [
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
MyRobot/1540834232469",
            "deploymentStartTime": 1551218376.0,
            "status": "Deploying",
            "progressDetail": {}
        }
    ],
    "tags": {}
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDeploymentJob。

describe-fleet

以下程式碼範例顯示如何使用 describe-fleet。

AWS CLI

描述機群

下列describe-fleet範例會擷取指定機群的詳細資訊。

```
aws robomaker describe-fleet \
    --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1550771358907
```

輸出：

```
{
```

```
    "name": "MyFleet",
    "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1539894765711",
    "robots": [
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
MyRobot/1540834232469",
            "createdAt": 1540834232.0
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
MyOtherRobot/1540829698778",
            "createdAt": 1540829698.0
        }
    ],
    "createdAt": 1539894765.0,
    "lastDeploymentStatus": "Succeeded",
    "lastDeploymentJob": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/
deployment-xl8qssl6pbcn",
    "lastDeploymentTime": 1551218369.0,
    "tags": {}
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFleet。

describe-robot-application

以下程式碼範例顯示如何使用 describe-robot-application。

AWS CLI

描述機器人應用程式

此範例說明機器人應用程式。

命令：

```
aws robomaker describe-robot-application --application arn:aws:robomaker:us-
west-2:111111111111:robot-application/MyRobotApplication/1551203485821
```

輸出：

```
{
```

```
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
MyRobotApplication/1551203485821",
  "name": "MyRobotApplication",
  "version": "$LATEST",
  "sources": [
      {
          "s3Bucket": "amzn-s3-demo-bucket",
          "s3Key": "my-robot-application.tar.gz",
          "architecture": "X86_64"
      }
  ],
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
  },
  "revisionId": "e72efe0d-f44f-4333-b604-f6fa5c6bb50b",
  "lastUpdatedAt": 1551203485.0,
  "tags": {}
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRobotApplication。

describe-robot

以下程式碼範例顯示如何使用 describe-robot。

AWS CLI

描述機器人

此範例說明機器人。

命令：

```
aws robomaker describe-robot --robot arn:aws:robomaker:us-west-2:111111111111:robot/
MyRobot/1550772324398
```

輸出：

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398",
  "name": "MyRobot",
```

```
  "status": "Available",
  "greengrassGroupId": "0f728a3c-7dbf-4a3e-976d-d16a8360caba",
  "createdAt": 1550772325.0,
  "architecture": "ARMHF",
  "tags": {
      "Region": "East"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeRobot。

describe-simulation-application

以下程式碼範例顯示如何使用 describe-simulation-application。

AWS CLI

描述模擬應用程式

此範例說明模擬應用程式。

命令：

```
aws robomaker describe-simulation-application --application arn:aws:robomaker:us-
west-2:111111111111:simulation-application/MySimulationApplication/1551203427605
```

輸出：

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
MySimulationApplication/1551203427605",
  "name": "MySimulationApplication",
  "version": "$LATEST",
  "sources": [
      {
          "s3Bucket": "amzn-s3-demo-bucket",
          "s3Key": "my-simulation-application.tar.gz",
          "architecture": "X86_64"
      }
  ],
  "simulationSoftwareSuite": {
```

```
        "name": "Gazebo",
        "version": "7"
    },
    "robotSoftwareSuite": {
        "name": "ROS",
        "version": "Kinetic"
    },
    "renderingEngine": {
        "name": "OGRE",
        "version": "1.x"
    },
    "revisionId": "783674ab-b7b8-42d9-b01f-9373907987e5",
    "lastUpdatedAt": 1551203427.0,
    "tags": {}
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSimulationApplication。

describe-simulation-job

以下程式碼範例顯示如何使用 describe-simulation-job。

AWS CLI

描述模擬任務

此範例說明模擬任務。

命令：

```
aws robomaker describe-simulation-job --job arn:aws:robomaker:us-
west-2:111111111111:simulation-job/sim-pql32v7pfjy6
```

輸出：

```
{
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-pql32v7pfjy6",
    "status": "Running",
    "lastUpdatedAt": 1551219349.0,
    "failureBehavior": "Continue",
    "clientRequestToken": "a19ec4b5-e50d-3591-33da-c2e593c60615",
```

```
  "outputLocation": {
      "s3Bucket": "my-output-bucket",
      "s3Prefix": "output"
  },
  "maxJobDurationInSeconds": 3600,
  "simulationTimeMillis": 0,
  "iamRole": "arn:aws:iam::111111111111:role/MySimulationRole",
  "robotApplications": [
      {
          "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/MyRobotApplication/1551206341136",
          "applicationVersion": "$LATEST",
          "launchConfig": {
              "packageName": "hello_world_robot",
              "launchFile": "rotate.launch"
          }
      }
  ],
  "simulationApplications": [
      {
          "application": "arn:aws:robomaker:us-west-2:111111111111:simulation-
application/MySimulationApplication/1551206347967",
          "applicationVersion": "$LATEST",
          "launchConfig": {
              "packageName": "hello_world_simulation",
              "launchFile": "empty_world.launch"
          }
      }
  ],
  "tags": {}
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSimulationJob。

`list-deployment-jobs`

以下程式碼範例顯示如何使用 `list-deployment-jobs`。

AWS CLI

列出部署任務

下列`list-deployment-jobs`範例會擷取部署任務的清單。

```
aws robomaker list-deployment-jobs
```

輸出：

```
{
    "deploymentJobs": [
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/
sim-6293szzm56rv",
            "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1539894765711",
            "status": "InProgress",
            "deploymentApplicationConfigs": [
                {
                    "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/HelloWorldRobot/1546537110575",
                    "applicationVersion": "1",
                    "launchConfig": {
                        "packageName": "hello_world_robot",
                        "launchFile": "rotate.launch",
                        "environmentVariables": {
                            "ENVIRONMENT": "Desert"
                        }
                    }
                }
            ],
            "deploymentConfig": {
                "concurrentDeploymentPercentage": 20,
                "failureThresholdPercentage": 25
            },
            "createdAt": 1550689373.0
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/
deployment-4w4g69p25zdb",
            "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1539894765711",
            "status": "Pending",
            "deploymentApplicationConfigs": [
                {
                    "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/AWSRoboMakerHelloWorld-1544562726923_YGHM_sh5M/1544562822877",
                    "applicationVersion": "1",
```

```
                    "launchConfig": {
                        "packageName": "fail",
                        "launchFile": "fail"
                    }
                }
            ],
            "deploymentConfig": {
                "concurrentDeploymentPercentage": 20,
                "failureThresholdPercentage": 25
            },
            "failureReason": "",
            "failureCode": "",
            "createdAt": 1544719763.0
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDeploymentJobs。

list-fleets

以下程式碼範例顯示如何使用 `list-fleets`。

AWS CLI

列出機群

此範例列出機群。最多將傳回 20 個機群。

命令：

```
aws robomaker list-fleets --max-items 20
```

輸出：

```
{
  "fleetDetails": [
      {
          "name": "Trek",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1539894765711",
          "createdAt": 1539894765.0,
```

```
            "lastDeploymentStatus": "Failed",
            "lastDeploymentJob": "arn:aws:robomaker:us-west-2:111111111111:deployment-
job/deployment-4w4g69p25zdb",
            "lastDeploymentTime": 1544719763.0
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFleets。

list-robot-applications

以下程式碼範例顯示如何使用 list-robot-applications。

AWS CLI

列出機器人應用程式

此範例列出機器人應用程式。結果限制為 20 個機器人應用程式。

命令：

```
aws robomaker list-robot-applications --max-results 20
```

輸出：

```
{
  "robotApplicationSummaries": [
      {
          "name": "MyRobot",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
MyRobot/1546537110575",
          "version": "$LATEST",
          "lastUpdatedAt": 1546540372.0
      },
      {
          "name": "AnotherRobot",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
AnotherRobot/1546541208251",
          "version": "$LATEST",
          "lastUpdatedAt": 1546541208.0
      },
```

```
      {
          "name": "MySuperRobot",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
MySuperRobot/1547663517377",
          "version": "$LATEST",
          "lastUpdatedAt": 1547663517.0
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRobotApplications。

`list-robots`

以下程式碼範例顯示如何使用 `list-robots`。

AWS CLI

列出機器人

此範例列出機器人。最多將傳回 20 個機器人。

命令：

```
aws robomaker list-robots --max-results 20
```

輸出：

```
{
  "robots": [
      {
          "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
Robot100/1544035373264",
          "name": "Robot100",
          "status": "Available",
          "createdAt": 1544035373.0,
          "architecture": "X86_64"
      },
      {
          "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
Robot101/1542146976587",
          "name": "Robot101",
```

```
            "status": "Available",
            "createdAt": 1542146976.0,
            "architecture": "X86_64"
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
Robot102/1540834232469",
            "name": "Robot102",
            "fleetArn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
Trek/1539894765711",
            "status": "Available",
            "createdAt": 1540834232.0,
            "architecture": "X86_64",
            "lastDeploymentJob": "arn:aws:robomaker:us-west-2:111111111111:deployment-
job/deployment-jb007b75gl5f",
            "lastDeploymentTime": 1550689533.0
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
MyRobot/1540829698778",
            "name": "MyRobot",
            "status": "Registered",
            "createdAt": 1540829698.0,
            "architecture": "X86_64"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRobots。

`list-simulation-applications`

以下程式碼範例顯示如何使用 `list-simulation-applications`。

AWS CLI

列出模擬應用程式

此範例列出模擬應用程式。最多將傳回 20 個模擬應用程式。

命令：

```
aws robomaker list-simulation-applications --max-results 20
```

輸出：

```
{
  "simulationApplicationSummaries": [
      {
          "name": "AWSRoboMakerObjectTracker-1548959046124_NPvyfcatq",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
AWSRoboMakerObjectTracker-1548959046124_NPvyfcatq/1548959170096",
          "version": "$LATEST",
          "lastUpdatedAt": 1548959170.0
      },
      {
          "name": "RoboMakerHelloWorldSimulation",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
RoboMakerHelloWorldSimulation/1546541198985",
          "version": "$LATEST",
          "lastUpdatedAt": 1546541198.0
      },
      {
          "name": "RoboMakerObjectTrackerSimulation",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
RoboMakerObjectTrackerSimulation/1545846795615",
          "version": "$LATEST",
          "lastUpdatedAt": 1545847405.0
      },
      {
          "name": "RoboMakerVoiceInteractionSimulation",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
RoboMakerVoiceInteractionSimulation/1546537100507",
          "version": "$LATEST",
          "lastUpdatedAt": 1546540352.0
      },
      {
          "name": "AWSRoboMakerCloudWatch-1547663411642_0LIt6D1h6",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
AWSRoboMakerCloudWatch-1547663411642_0LIt6D1h6/1547663521470",
          "version": "$LATEST",
          "lastUpdatedAt": 1547663521.0
      },
      {
          "name": "AWSRoboMakerDeepRacer-1545848257672_1YZCaieQ-",
          "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
AWSRoboMakerDeepRacer-1545848257672_1YZCaieQ-/1545848370525",
          "version": "$LATEST",
```

```
            "lastUpdatedAt": 1545848370.0
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSimulationApplications。

list-simulation-jobs

以下程式碼範例顯示如何使用 list-simulation-jobs。

AWS CLI

列出模擬任務

此範例列出模擬任務。

命令：

```
aws robomaker list-simulation-jobs
```

輸出：

```
{
  "simulationJobSummaries": [
      {
          "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/
sim-66bbb3gpxm8x",
          "lastUpdatedAt": 1548959178.0,
          "status": "Completed",
          "simulationApplicationNames": [
              "AWSRoboMakerObjectTracker-1548959046124_NPvyfcatq"
          ],
          "robotApplicationNames": [
              null
          ]
      },
      {
          "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
b27c4rkrtzcw",
          "lastUpdatedAt": 1543514088.0,
          "status": "Canceled",
          "simulationApplicationNames": [
```

```
                "AWSRoboMakerPersonDetection-1543513948280_T8rHW2_lu"
            ],
            "robotApplicationNames": [
                "AWSRoboMakerPersonDetection-1543513948280_EYaMT0mYb"
            ]
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/
sim-51vxjbzy4q8t",
            "lastUpdatedAt": 1543508858.0,
            "status": "Canceled",
            "simulationApplicationNames": [
                "AWSRoboMakerCloudWatch-1543504747391_lFF9ZQyx6"
            ],
            "robotApplicationNames": [
                "AWSRoboMakerCloudWatch-1543504747391_axbYa3S3K"
            ]
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
kgf1fqxflqbx",
            "lastUpdatedAt": 1543504862.0,
            "status": "Completed",
            "simulationApplicationNames": [
                "AWSRoboMakerCloudWatch-1543504747391_lFF9ZQyx6"
            ],
            "robotApplicationNames": [
                "AWSRoboMakerCloudWatch-1543504747391_axbYa3S3K"
            ]
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
vw8lvh061nqt",
            "lastUpdatedAt": 1543441430.0,
            "status": "Completed",
            "simulationApplicationNames": [
                "AWSRoboMakerHelloWorld-1543437372341__yb_Jg96l"
            ],
            "robotApplicationNames": [
                "AWSRoboMakerHelloWorld-1543437372341_lNbmKHvs9"
            ]
        },
        {
```

```
            "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
txy5ypxmhz84",
            "lastUpdatedAt": 1543437488.0,
            "status": "Completed",
            "simulationApplicationNames": [
                "AWSRoboMakerHelloWorld-1543437372341__yb_Jg96l"
            ],
            "robotApplicationNames": [
                "AWSRoboMakerHelloWorld-1543437372341_lNbmKHvs9"
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSimulationJobs。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出資源的標籤

此範例列出 an AWS RoboMaker 資源的標籤。

命令：

```
aws robomaker list-tags-for-resource --resource-arn "arn:aws:robomaker:us-
west-2:111111111111:robot/Robby_the_Robot/1544035373264"
```

輸出：

```
{
    "tags": {
        "Region": "North",
        "Stage": "Initial"
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

register-robot

以下程式碼範例顯示如何使用 register-robot。

AWS CLI

註冊機器人

此範例會將機器人註冊到機群。

命令：

```
aws robomaker register-robot --fleet arn:aws:robomaker:us-
west-2:111111111111:deployment-fleet/MyFleet/1550771358907 --
robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398
```

輸出：

```
{
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1550771358907",
  "robot": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterRobot。

restart-simulation-job

以下程式碼範例顯示如何使用 restart-simulation-job。

AWS CLI

重新啟動模擬

此範例會重新啟動模擬。

命令：

```
aws robomaker restart-simulation-job --job arn:aws:robomaker:us-
west-2:111111111111:simulation-job/sim-t6rdgt70mftr
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestartSimulationJob。

sync-deployment-job

以下程式碼範例顯示如何使用 sync-deployment-job。

AWS CLI

同步部署任務

此範例會同步部署任務。

命令：

```
aws robomaker sync-deployment-job --fleet arn:aws:robomaker:us-
west-2:111111111111:deployment-fleet/Trek/1539894765711
```

輸出：

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/
deployment-09ccxs3tlfms",
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1539894765711",
  "status": "Pending",
  "deploymentConfig": {
      "concurrentDeploymentPercentage": 20,
      "failureThresholdPercentage": 25
  },
  "deploymentApplicationConfigs": [
      {
          "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/MyRobotApplication/1546541208251",
          "applicationVersion": "1",
          "launchConfig": {
              "packageName": "hello_world_simulation",
              "launchFile": "empty_world.launch"
          }
      }
  ],
  "createdAt": 1551286954.0
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SyncDeploymentJob。

`tag-resource`

以下程式碼範例顯示如何使用 `tag-resource`。

AWS CLI

標記資源

此範例會標記資源。它連接兩個標籤：區域和階段。

命令：

```
aws robomaker tag-resource --resource-arn "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1544035373264" --tags Region=North,Stage=Initial
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

`untag-resource`

以下程式碼範例顯示如何使用 `untag-resource`。

AWS CLI

取消標記資源

此範例會從資源移除標籤。它會移除區域標籤。

命令：

```
aws robomaker untag-resource --resource-arn "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1544035373264" --tag-keys Region
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

`update-robot-application`

以下程式碼範例顯示如何使用 `update-robot-application`。

AWS CLI

更新機器人應用程式

此範例會更新機器人應用程式。

命令：

```
aws robomaker update-robot-application --application arn:aws:robomaker:us-
west-2:111111111111:robot-application/MyRobotApplication/1551203485821
 --sources s3Bucket=amzn-s3-demo-bucket,s3Key=my-robot-
application.tar.gz,architecture=X86_64 --robot-software-
suite name=ROS,version=Kinetic
```

輸出：

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
MyRobotApplication/1551203485821",
  "name": "MyRobotApplication",
  "version": "$LATEST",
  "sources": [
      {
          "s3Bucket": "amzn-s3-demo-bucket",
          "s3Key": "my-robot-application.tar.gz",
          "architecture": "X86_64"
      }
  ],
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
  },
  "lastUpdatedAt": 1551287993.0,
  "revisionId": "20b5e331-24fd-4504-8b8c-531afe5f4c94"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRobotApplication。

update-simulation-application

以下程式碼範例顯示如何使用 update-simulation-application。

AWS CLI

更新模擬應用程式

此範例會更新模擬應用程式。

命令：

```
aws robomaker update-simulation-application --application arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605 --sources s3Bucket=amzn-s3-demo-bucket,s3Key=my-simulation-application.tar.gz,architecture=X86_64 --robot-software-suite name=ROS,version=Kinetic --simulation-software-suite name=Gazebo,version=7 --rendering-engine name=OGRE,version=1.x
```

輸出：

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605",
  "name": "MySimulationApplication",
  "version": "$LATEST",
  "sources": [
      {
          "s3Bucket": "amzn-s3-demo-bucket",
          "s3Key": "my-simulation-application.tar.gz",
          "architecture": "X86_64"
      }
  ],
  "simulationSoftwareSuite": {
      "name": "Gazebo",
      "version": "7"
  },
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
  },
  "renderingEngine": {
      "name": "OGRE",
      "version": "1.x"
  },
  "lastUpdatedAt": 1551289361.0,
  "revisionId": "4a22cb5d-93c5-4cef-9311-52bdd119b79e"
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSimulationApplication。

使用 的 Route 53 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Route 53 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`change-resource-record-sets`

以下程式碼範例顯示如何使用 `change-resource-record-sets`。

AWS CLI

建立、更新或刪除資源紀錄集

下列`change-resource-record-sets`命令會使用 檔案中的 `hosted-zone-id`Z1R8UBAEXAMPLE和 JSON 格式組態來建立資源紀錄集`C:\awscli\route53\change-resource-record-sets.json`：

```
aws route53 change-resource-record-sets --hosted-zone-id Z1R8UBAEXAMPLE --change-batch file://C:\awscli\route53\change-resource-record-sets.json
```

如需詳細資訊，請參閱《Amazon Route 53 API 參考》中的 POST ChangeResourceRecordSets。

JSON 檔案中的組態取決於您要建立的資源紀錄集類型：

BasicWeightedAliasWeighted AliasLatencyLatency AliasFailoverFailover 別名

基本語法：

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "TTL": time to live in seconds,
        "ResourceRecords": [
          {
            "Value": "applicable value for the record type"
          },
          {...}
        ]
      }
    },
    {...}
  ]
}
```

加權語法：

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Weight": value between 0 and 255,
        "TTL": time to live in seconds,
        "ResourceRecords": [
          {
            "Value": "applicable value for the record type"
          },
          {...}
        ],
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
```

```
    {...}
  ]
}
```

別名語法：

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "AliasTarget": {
          "HostedZoneId": "hosted zone ID for your CloudFront distribution, Amazon
 S3 bucket, Elastic Load Balancing load balancer, or Amazon Route 53 hosted zone",
          "DNSName": "DNS domain name for your CloudFront distribution, Amazon S3
 bucket, Elastic Load Balancing load balancer, or another resource record set in
 this hosted zone",
          "EvaluateTargetHealth": true|false
        },
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}
```

加權別名語法：

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Weight": value between 0 and 255,
        "AliasTarget": {
```

```
          "HostedZoneId": "hosted zone ID for your CloudFront distribution, Amazon
 S3 bucket, Elastic Load Balancing load balancer, or Amazon Route 53 hosted zone",
          "DNSName": "DNS domain name for your CloudFront distribution, Amazon S3
 bucket, Elastic Load Balancing load balancer, or another resource record set in
 this hosted zone",
          "EvaluateTargetHealth": true|false
        },
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}
```

延遲語法：

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Region": "Amazon EC2 region name",
        "TTL": time to live in seconds,
        "ResourceRecords": [
          {
            "Value": "applicable value for the record type"
          },
          {...}
        ],
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}
```

延遲別名語法：

```
{
```

```
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Region": "Amazon EC2 region name",
        "AliasTarget": {
          "HostedZoneId": "hosted zone ID for your CloudFront distribution, Amazon
 S3 bucket, Elastic Load Balancing load balancer, or Amazon Route 53 hosted zone",
          "DNSName": "DNS domain name for your CloudFront distribution, Amazon S3
 bucket, Elastic Load Balancing load balancer, or another resource record set in
 this hosted zone",
          "EvaluateTargetHealth": true|false
        },
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}
```

容錯移轉語法：

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Failover": "PRIMARY" | "SECONDARY",
        "TTL": time to live in seconds,
        "ResourceRecords": [
          {
            "Value": "applicable value for the record type"
          },
          {...}
        ],
```

```
        "HealthCheckId": "ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}
```

容錯移轉別名語法：

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Failover": "PRIMARY" | "SECONDARY",
        "AliasTarget": {
          "HostedZoneId": "hosted zone ID for your CloudFront distribution, Amazon
 S3 bucket, Elastic Load Balancing load balancer, or Amazon Route 53 hosted zone",
          "DNSName": "DNS domain name for your CloudFront distribution, Amazon S3
 bucket, Elastic Load Balancing load balancer, or another resource record set in
 this hosted zone",
          "EvaluateTargetHealth": true|false
        },
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ChangeResourceRecordSets。

change-tags-for-resource

以下程式碼範例顯示如何使用 change-tags-for-resource。

AWS CLI

下列命令owner會將名為 的標籤新增至 ID 指定的運作狀態檢查資源：

```
aws route53 change-tags-for-resource --resource-type healthcheck --resource-
id 6233434j-18c1-34433-ba8e-3443434 --add-tags Key=owner,Value=myboss
```

下列命令owner會從 ID 指定的託管區域資源中移除名為 的標籤：

```
aws route53 change-tags-for-resource --resource-type hostedzone --resource-
id Z1523434445 --remove-tag-keys owner
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ChangeTagsForResource。

create-health-check

以下程式碼範例顯示如何使用 create-health-check。

AWS CLI

建立運作狀態檢查

下列create-health-check命令會使用 檔案中的呼叫者參考2014-04-01-18:47和 JSON 格式組態來建立運作狀態檢查C:\awscli\route53\create-health-check.json：

```
aws route53 create-health-check --caller-reference 2014-04-01-18:47 --health-check-
config file://C:\awscli\route53\create-health-check.json
```

JSON 語法：

```
{
  "IPAddress": "IP address of the endpoint to check",
  "Port": port on the endpoint to check--required when Type is "TCP",
  "Type": "HTTP"|"HTTPS"|"HTTP_STR_MATCH"|"HTTPS_STR_MATCH"|"TCP",
  "ResourcePath": "path of the file that you want Amazon Route 53 to request--all
 Types except TCP",
  "FullyQualifiedDomainName": "domain name of the endpoint to check--all Types
 except TCP",
  "SearchString": "if Type is HTTP_STR_MATCH or HTTPS_STR_MATCH, the string to
 search for in the response body from the specified resource",
  "RequestInterval": 10 | 30,
  "FailureThreshold": integer between 1 and 10
}
```

若要將運作狀態檢查新增至 Route 53 資源記錄集，請使用 change-resource-record-sets命令。

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的 Amazon Route 53 運作狀態檢查和 DNS 容錯移轉。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateHealthCheck。

create-hosted-zone

以下程式碼範例顯示如何使用 create-hosted-zone。

AWS CLI

若要建立託管區域

下列create-hosted-zone命令example.com會使用呼叫者參考 新增名為 的託管區域2014-04-01-18:47。選用的註解包含空格，因此必須以引號括住：

```
aws route53 create-hosted-zone --name example.com --caller-
reference 2014-04-01-18:47 --hosted-zone-config Comment="command-line version"
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的使用託管區域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateHostedZone。

delete-health-check

以下程式碼範例顯示如何使用 delete-health-check。

AWS CLI

刪除運作狀態檢查

下列delete-health-check命令會使用 health-check-id的 刪除運作狀態檢查e75b48d9-547a-4c3d-88a5-ae4002397608：

```
aws route53 delete-health-check --health-check-id e75b48d9-547a-4c3d-88a5-
ae4002397608
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteHealthCheck。

delete-hosted-zone

以下程式碼範例顯示如何使用 delete-hosted-zone。

AWS CLI

刪除託管區域

下列delete-hosted-zone命令會使用 id的 刪除託管區域Z36KTIQEXAMPLE：

```
aws route53 delete-hosted-zone --id Z36KTIQEXAMPLE
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteHostedZone。

get-change

以下程式碼範例顯示如何使用 get-change。

AWS CLI

取得資源紀錄集變更的狀態

下列get-change命令會取得具有 Id之 change-resource-record-sets請求的狀態和其他資訊/change/CWPIK4URU2I5S：

```
aws route53 get-change --id /change/CWPIK4URU2I5S
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetChange。

get-health-check

以下程式碼範例顯示如何使用 get-health-check。

AWS CLI

取得運作狀態檢查的相關資訊

下列get-health-check命令會取得具有 health-check-id之 的運作狀態檢查相關資訊02ec8401-9879-4259-91fa-04e66d094674：

```
aws route53 get-health-check --health-check-id 02ec8401-9879-4259-91fa-04e66d094674
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetHealthCheck。

get-hosted-zone

以下程式碼範例顯示如何使用 get-hosted-zone。

AWS CLI

取得託管區域的相關資訊

下列get-hosted-zone命令會使用 id的 取得託管區域的相關資訊Z1R8UBAEXAMPLE：

```
aws route53 get-hosted-zone --id Z1R8UBAEXAMPLE
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetHostedZone。

list-health-checks

以下程式碼範例顯示如何使用 list-health-checks。

AWS CLI

列出與目前 AWS 帳戶相關聯的運作狀態檢查

下列list-health-checks命令會列出與目前 AWS 帳戶相關聯的前 100 個運作狀態檢查的詳細資訊：

```
aws route53 list-health-checks
```

如果您有超過 100 個運作狀態檢查，或想要在小於 100 的群組中列出它們，請包含 --maxitems 參數。例如，若要一次列出一個運作狀態檢查，請使用下列命令：

```
aws route53 list-health-checks --max-items 1
```

若要檢視下一個運作狀態檢查，請從對上一個命令的NextToken回應中取得 的值，並將其包含在 --starting-token 參數中，例如：

```
aws route53 list-health-checks --max-items 1 --starting-token Z3M3LMPEXAMPLE
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListHealthChecks。

list-hosted-zones-by-name

以下程式碼範例顯示如何使用 list-hosted-zones-by-name。

AWS CLI

下列命令會列出最多 100 個依網域名稱排序的託管區域：

```
aws route53 list-hosted-zones-by-name
```

輸出：

```
{
  "HostedZones": [
      {
          "ResourceRecordSetCount": 2,
          "CallerReference": "test20150527-2",
          "Config": {
              "Comment": "test2",
              "PrivateZone": false
          },
          "Id": "/hostedzone/Z119WBBTVP5WFX",
          "Name": "2.example.com."
      },
      {
          "ResourceRecordSetCount": 2,
          "CallerReference": "test20150527-1",
          "Config": {
              "Comment": "test",
              "PrivateZone": false
          },
          "Id": "/hostedzone/Z3P5QSUBK4POTI",
          "Name": "www.example.com."
      }
  ],
  "IsTruncated": false,
  "MaxItems": "100"
```

```
}
```

下列命令會列出依名稱排序的託管區域，開頭為 www.example.com：

```
aws route53 list-hosted-zones-by-name --dns-name www.example.com
```

輸出：

```
{
  "HostedZones": [
      {
          "ResourceRecordSetCount": 2,
          "CallerReference": "mwunderl20150527-1",
          "Config": {
              "Comment": "test",
              "PrivateZone": false
          },
          "Id": "/hostedzone/Z3P5QSUBK4POTI",
          "Name": "www.example.com."
      }
  ],
  "DNSName": "www.example.com",
  "IsTruncated": false,
  "MaxItems": "100"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListHostedZonesByName。

list-hosted-zones

以下程式碼範例顯示如何使用 list-hosted-zones。

AWS CLI

列出與目前 AWS 帳戶相關聯的託管區域

下列list-hosted-zones命令會列出與目前 AWS 帳戶相關聯的前 100 個託管區域的摘要資訊：

```
aws route53 list-hosted-zones
```

如果您具有 100 個以上的託管區域，或者要以小於 100 的群組列示它們，請包括 --max-items 參數。例如，若要一次列出一個託管區域，請使用下列命令：

```
aws route53 list-hosted-zones --max-items 1
```

若要檢視有關下一個託管區域的資訊，請從上一個命令的回應中取得 NextToken 值，並將其包含在 --starting-token 參數中，例如：

```
aws route53 list-hosted-zones --max-items 1 --starting-token Z3M3LMPEXAMPLE
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListHostedZones。

list-query-logging-configs

以下程式碼範例顯示如何使用 list-query-logging-configs。

AWS CLI

列出查詢記錄組態

下列list-query-logging-configs範例會針對託管區域 列出您 AWS 帳戶中前 100 個查詢記錄組態的相關資訊Z1OX3WQEXAMPLE。

```
aws route53 list-query-logging-configs \
    --hosted-zone-id Z1OX3WQEXAMPLE
```

輸出：

```
{
    "QueryLoggingConfigs": [
        {
            "Id": "964ff34e-ae03-4f06-80a2-9683cexample",
            "HostedZoneId": "Z1OX3WQEXAMPLE",
            "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:111122223333:log-
group:/aws/route53/example.com:*"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的記錄 DNS 查詢。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListQueryLoggingConfigs。

list-resource-record-sets

以下程式碼範例顯示如何使用 list-resource-record-sets。

AWS CLI

列出託管區域中的資源紀錄集

下列list-resource-record-sets命令列出指定託管區域中前 100 個資源紀錄集的摘要資訊：

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE
```

如果託管區域包含超過 100 個資源紀錄集，或者如果您想要在小於 100 的群組中列出它們，請包含 --maxitems 參數。例如，若要一次列出一個資源記錄集，請使用下列命令：

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --max-items 1
```

若要檢視託管區域中下一個資源紀錄集的相關資訊，請從對上一個命令的NextToken回應取得 的值，並將其包含在 --starting-token 參數中，例如：

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --max-items 1
 --starting-token Z3M3LMPEXAMPLE
```

若要檢視特定名稱的所有資源紀錄集，請使用 --query 參數來篩選掉它們。例如：

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --
query "ResourceRecordSets[?Name == 'example.domain.']"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourceRecordSets。

使用 的 Route 53 網域註冊範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 具有 Route 53 網域註冊的 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`check-domain-availability`

以下程式碼範例顯示如何使用 `check-domain-availability`。

AWS CLI

判斷您是否可以使用 Route 53 註冊網域名稱

下列`check-domain-availability`命令會傳回有關是否可以使用 Route 53 `example.com`註冊網域名稱的資訊。

此命令僅在 `us-east-1`區域中執行。如果您的預設區域設定為 `us-east-1`，您可以省略 `region` 參數。

```
aws route53domains check-domain-availability \
    --region us-east-1 \
    --domain-name example.com
```

輸出：

```
{
    "Availability": "UNAVAILABLE"
}
```

Route 53 支援大量的頂層網域 (TLDs)，例如 `.com`和 `.jp`，但我們不支援所有可用的 TLDs。如果您檢查網域的可用性，且 Route 53 不支援 TLD，則 會`check-domain-availability`傳回下列訊息。

```
An error occurred (UnsupportedTLD) when calling the CheckDomainAvailability
 operation: <top-level domain> tld is not supported.
```

如需向 Route 53 註冊網域時可使用TLDs 清單，請參閱《Amazon Route 53 開發人員指南》中的向 Amazon Route 53 註冊的網域。 Amazon Route 53 如需使用 Amazon Route 53 註冊網域的詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的註冊新網域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CheckDomainAvailability。

check-domain-transferability

以下程式碼範例顯示如何使用 check-domain-transferability。

AWS CLI

判斷網域是否可以轉移到 Route 53

下列check-domain-transferability命令會傳回有關是否可以將網域名稱轉移至 example.com Route 53 的資訊。

此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains check-domain-transferability \
    --region us-east-1 \
    --domain-name example.com
```

輸出：

```
{
    "Transferability": {
        "Transferable": "UNTRANSFERABLE"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的將網域註冊轉移至 Amazon Route 53。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CheckDomainTransferability。

delete-tags-for-domain

以下程式碼範例顯示如何使用 delete-tags-for-domain。

AWS CLI

刪除網域的標籤

下列delete-tags-for-domain命令會從指定的網域刪除三個標籤。請注意，您只指定標籤索引鍵，而不是標籤值。

此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains delete-tags-for-domain \
    --region us-east-1 \
    --domain-name example.com \
    --tags-to-delete accounting-key hr-key engineering-key
```

此命令不會產生輸出。

若要確認標籤已刪除，您可以執行 list-tags-for-domain 。如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的標記 Amazon Route 53 資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTagsForDomain。

disable-domain-auto-renew

以下程式碼範例顯示如何使用 disable-domain-auto-renew。

AWS CLI

停用網域的自動續約

下列disable-domain-auto-renew命令會將 Route 53 設定為在網域註冊過期example.com之前不會自動續約網域。

此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains disable-domain-auto-renew \
    --region us-east-1 \
    --domain-name example.com
```

此命令不會產生輸出。

若要確認設定已變更，您可以執行 get-domain-detail 。如果停用自動續約，則 的值AutoRenew為 False。如需自動續約的詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的續約網域註冊 <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-renew.html。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableDomainAutoRenew。

disable-domain-transfer-lock

以下程式碼範例顯示如何使用 disable-domain-transfer-lock。

AWS CLI

停用網域上的轉移鎖定

下列disable-domain-transfer-lock命令會移除網域上的傳輸鎖定，example.com以便將網域轉移到另一個註冊商。此命令會變更clientTransferProhibited狀態。

此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains disable-domain-transfer-lock \
    --region us-east-1 \
    --domain-name example.com
```

輸出：

```
{
    "OperationId": "3f28e0ac-126a-4113-9048-cc930example"
}
```

若要確認轉移鎖定已變更，您可以執行 get-domain-detail 。停用傳輸鎖定時， 的值StatusList不包含 clientTransferProhibited。

如需轉移程序的詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的將網域從 Amazon Route 53 轉移至另一個註冊商。 Amazon Route 53

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableDomainTransferLock。

enable-domain-auto-renew

以下程式碼範例顯示如何使用 enable-domain-auto-renew。

AWS CLI

啟用網域的自動續約

下列enable-domain-auto-renew命令會將 Route 53 設定為在網域註冊過期example.com之前自動續約網域。

此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains enable-domain-auto-renew \
    --region us-east-1 \
    --domain-name example.com
```

此命令不會產生輸出。若要確認設定已變更，您可以執行 get-domain-detail 。如果啟用自動續約，則 的值AutoRenew為 True。

如需自動續約的詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的續約網域註冊 <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-renew.html。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableDomainAutoRenew。

enable-domain-transfer-lock

以下程式碼範例顯示如何使用 enable-domain-transfer-lock。

AWS CLI

在網域上啟用轉移鎖定

下列enable-domain-transfer-lock命令會鎖定指定的網域，使其無法傳輸至其他註冊商。此命令會變更clientTransferProhibited狀態。

此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains enable-domain-transfer-lock \
    --region us-east-1 \
    --domain-name example.com
```

輸出：

```
{
    "OperationId": "3f28e0ac-126a-4113-9048-cc930example"
}
```

若要確認轉移鎖定已變更，您可以執行 get-domain-detail 。啟用傳輸鎖定時， 的值StatusList會包含 clientTransferProhibited。

如需轉移程序的詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的將網域從 Amazon Route 53 轉移至另一個註冊商。 Amazon Route 53

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableDomainTransferLock。

get-contact-reachability-status

以下程式碼範例顯示如何使用 get-contact-reachability-status。

AWS CLI

判斷註冊聯絡人是否已回應確認電子郵件

下列get-contact-reachability-status命令會傳回指定網域的註冊聯絡人是否已回應確認電子郵件的相關資訊。

此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains get-contact-reachability-status \
    --region us-east-1 \
    --domain-name example.com
```

輸出：

```
{
    "domainName": "example.com",
    "status": "DONE"
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的重新傳送授權和確認電子郵件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetContactReachabilityStatus。

get-domain-detail

以下程式碼範例顯示如何使用 get-domain-detail。

AWS CLI

取得指定網域的詳細資訊

下列get-domain-detail命令會顯示指定網域的詳細資訊。

此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains get-domain-detail \
    --region us-east-1 \
    --domain-name example.com
```

輸出：

```
{
    "DomainName": "example.com",
    "Nameservers": [
        {
            "Name": "ns-2048.awsdns-64.com",
            "GlueIps": []
        },
        {
            "Name": "ns-2049.awsdns-65.net",
            "GlueIps": []
        },
        {
            "Name": "ns-2050.awsdns-66.org",
            "GlueIps": []
        },
        {
            "Name": "ns-2051.awsdns-67.co.uk",
            "GlueIps": []
        }
    ],
    "AutoRenew": true,
    "AdminContact": {
        "FirstName": "Saanvi",
        "LastName": "Sarkar",
        "ContactType": "COMPANY",
        "OrganizationName": "Example",
        "AddressLine1": "123 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "ssarkar@example.com",
        "ExtraParams": []
```

```
    },
    "RegistrantContact": {
        "FirstName": "Alejandro",
        "LastName": "Rosalez",
        "ContactType": "COMPANY",
        "OrganizationName": "Example",
        "AddressLine1": "123 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "arosalez@example.com",
        "ExtraParams": []
    },
    "TechContact": {
        "FirstName": "Wang",
        "LastName": "Xiulan",
        "ContactType": "COMPANY",
        "OrganizationName": "Example",
        "AddressLine1": "123 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "wxiulan@example.com",
        "ExtraParams": []
    },
    "AdminPrivacy": true,
    "RegistrantPrivacy": true,
    "TechPrivacy": true,
    "RegistrarName": "Amazon Registrar, Inc.",
    "WhoIsServer": "whois.registrar.amazon",
    "RegistrarUrl": "http://registrar.amazon.com",
    "AbuseContactEmail": "abuse@registrar.amazon.com",
    "AbuseContactPhone": "+1.2062661000",
    "CreationDate": 1444934889.601,
    "ExpirationDate": 1602787689.0,
    "StatusList": [
        "clientTransferProhibited"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDomainDetail。

get-domain-suggestions

以下程式碼範例顯示如何使用 get-domain-suggestions。

AWS CLI

若要取得建議的網域名稱清單

下列get-domain-suggestions命令會根據網域名稱 顯示建議的網域名稱清單example.com。回應僅包含可用的網域名稱。此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains get-domain-suggestions \
    --region us-east-1 \
    --domain-name example.com \
    --suggestion-count 10 \
    --only-available
```

輸出：

```
{
    "SuggestionsList": [
        {
            "DomainName": "egzaampal.com",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "examplelaw.com",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "examplehouse.net",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "homeexample.net",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "examplelist.com",
```

```
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "examplenews.net",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "officeexample.com",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "exampleworld.com",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "exampleart.com",
            "Availability": "AVAILABLE"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDomainSuggestions。

get-operation-detail

以下程式碼範例顯示如何使用 get-operation-detail。

AWS CLI

取得 操作的目前狀態

有些網域註冊操作會以非同步方式運作，並在完成之前傳回回應。這些操作會傳回操作 ID，您可用來取得目前的狀態。下列get-operation-detail命令會傳回指定操作的狀態。

此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains get-operation-detail \
    --region us-east-1 \
    --operation-id edbd8d63-7fe7-4343-9bc5-54033example
```

輸出：

```
{
    "OperationId": "edbd8d63-7fe7-4343-9bc5-54033example",
    "Status": "SUCCESSFUL",
    "DomainName": "example.com",
    "Type": "DOMAIN_LOCK",
    "SubmittedDate": 1573749367.864
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetOperationDetail。

`list-domains`

以下程式碼範例顯示如何使用 `list-domains`。

AWS CLI

列出已向目前 AWS 帳戶註冊的網域

下列`list-domains`命令會列出已向目前 AWS 帳戶註冊之網域的摘要資訊。

此命令僅在 `us-east-1`區域中執行。如果您的預設區域設定為 `us-east-1`，您可以省略 `region` 參數。

```
aws route53domains list-domains
    --region us-east-1
```

輸出：

```
{
    "Domains": [
        {
            "DomainName": "example.com",
            "AutoRenew": true,
            "TransferLock": true,
            "Expiry": 1602712345.0
        },
        {
            "DomainName": "example.net",
            "AutoRenew": true,
            "TransferLock": true,
            "Expiry": 1602723456.0
        },
```

```
        {
            "DomainName": "example.org",
            "AutoRenew": true,
            "TransferLock": true,
            "Expiry": 1602734567.0
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDomains。

`list-operations`

以下程式碼範例顯示如何使用 `list-operations`。

AWS CLI

列出傳回操作 ID 的操作狀態

有些網域註冊操作會以非同步方式執行，並在完成之前傳回回應。這些操作會傳回操作 ID，您可用來取得目前的狀態。下列`list-operations`命令會列出目前網域註冊操作的摘要資訊，包括 狀態。

此命令僅在 `us-east-1`區域中執行。如果您的預設區域設定為 `us-east-1`，您可以省略 `region` 參數。

```
aws route53domains list-operations
    --region us-east-1
```

輸出：

```
{
    "Operations": [
        {
            "OperationId": "aab9822f-1da0-4bf3-8a15-fd4e0example",
            "Status": "SUCCESSFUL",
            "Type": "DOMAIN_LOCK",
            "SubmittedDate": 1455321739.986
        },
        {
            "OperationId": "c24379ed-76be-42f8-bdad-9379bexample",
            "Status": "SUCCESSFUL",
```

```
            "Type": "UPDATE_NAMESERVER",
            "SubmittedDate": 1468960475.109
        },
        {
            "OperationId": "f47e1297-ef9e-4c2b-ae1e-a5fcbexample",
            "Status": "SUCCESSFUL",
            "Type": "RENEW_DOMAIN",
            "SubmittedDate": 1473561835.943
        },
        {
            "OperationId": "75584f23-b15f-459e-aed7-dc6f5example",
            "Status": "SUCCESSFUL",
            "Type": "UPDATE_DOMAIN_CONTACT",
            "SubmittedDate": 1547501003.41
        }
    ]
}
```

輸出包含傳回 操作 ID 的所有操作，以及您對使用目前 AWS 帳戶註冊的所有網域執行的操作。如果只想取得指定日期之後提交的操作，您可以包含 `submitted-since` 參數，並以 Unix 格式和國際標準時間 (UTC) 指定日期。下列命令會取得 2020 年 1 月 1 日 UTC 上午 12：00 之後提交之所有操作的狀態。

```
aws route53domains list-operations \
    --submitted-since 1577836800
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListOperations。

`list-tags-for-domain`

以下程式碼範例顯示如何使用 `list-tags-for-domain`。

AWS CLI

列出網域的標籤

下列`list-tags-for-domain`命令會列出目前與指定網域相關聯的標籤。

此命令僅在 `us-east-1`區域中執行。如果您的預設區域設定為 `us-east-1`，您可以省略 `region` 參數。

```
aws route53domains list-tags-for-domain \
```

```
    --region us-east-1 \
    --domain-name example.com
```

輸出：

```
{
    "TagList": [
        {
            "Key": "key1",
            "Value": "value1"
        },
        {
            "Key": "key2",
            "Value": "value2"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的標記 Amazon Route 53 資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTagsForDomain。

register-domain

以下程式碼範例顯示如何使用 register-domain。

AWS CLI

註冊網域

下列register-domain命令會註冊網域，從 JSON 格式的檔案擷取所有參數值。

此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains register-domain \
    --region us-east-1 \
    --cli-input-json file://register-domain.json
```

register-domain.json 的內容：

```
{
    "DomainName": "example.com",
```

```
    "DurationInYears": 1,
    "AutoRenew": true,
    "AdminContact": {
        "FirstName": "Martha",
        "LastName": "Rivera",
        "ContactType": "PERSON",
        "OrganizationName": "Example",
        "AddressLine1": "1 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "mrivera@example.com"
    },
    "RegistrantContact": {
        "FirstName": "Li",
        "LastName": "Juan",
        "ContactType": "PERSON",
        "OrganizationName": "Example",
        "AddressLine1": "1 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "ljuan@example.com"
    },
    "TechContact": {
        "FirstName": "Mateo",
        "LastName": "Jackson",
        "ContactType": "PERSON",
        "OrganizationName": "Example",
        "AddressLine1": "1 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "mjackson@example.com"
    },
    "PrivacyProtectAdminContact": true,
    "PrivacyProtectRegistrantContact": true,
    "PrivacyProtectTechContact": true
```

```
}
```

輸出：

```
{
    "OperationId": "b114c44a-9330-47d1-a6e8-a0b11example"
}
```

若要確認操作成功，您可以執行 get-operation-detail。如需詳細資訊，請參閱 get-operation-detail 。

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的註冊新網域。

如需哪些頂層網域 (TLDs) 需要 的值ExtraParams和有效值的相關資訊，請參閱《Amazon Route 53 API 參考》中的 ExtraParam。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterDomain。

renew-domain

以下程式碼範例顯示如何使用 renew-domain。

AWS CLI

續約網域

下列renew-domain命令會將指定的網域續約五年。若要取得 的值current-expiry-year，請使用 get-domain-detail命令，然後從 ExpirationDate Unix 格式轉換 的值。

此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains renew-domain \
    --region us-east-1 \
    --domain-name example.com \
    --duration-in-years 5 \
    --current-expiry-year 2020
```

輸出：

```
{
    "OperationId": "3f28e0ac-126a-4113-9048-cc930example"
```

```
}
```

若要確認操作成功，您可以執行 get-operation-detail。如需詳細資訊，請參閱 get-operation-detail。

每個頂層網域 (TLD) 的登錄檔，例如 .com 或 .org，會控制您可以續約網域的年數上限。若要取得網域的最長續約期間，請參閱《Amazon Route 53 開發人員指南》中的網域中 TLD Amazon Route 53的「註冊和續約期間」一節。

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的續約網域的註冊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RenewDomain。

resend-contact-reachability-email

以下程式碼範例顯示如何使用 resend-contact-reachability-email。

AWS CLI

將確認電子郵件重新傳送至註冊聯絡人的目前電子郵件地址

下列resend-contact-reachability-email命令會將確認電子郵件重新傳送至 example.com 網域註冊聯絡人的目前電子郵件地址。

此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains resend-contact-reachability-email \
    --region us-east-1 \
    --domain-name example.com
```

輸出：

```
{
    "domainName": "example.com",
    "emailAddress": "moliveira@example.com",
    "isAlreadyVerified": true
}
```

如果 的值isAlreadyVerified為 true，如本範例所示，註冊聯絡人已確認指定的電子郵件地址可以連線。

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的重新傳送授權和確認電子郵件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResendContactReachabilityEmail。

`retrieve-domain-auth-code`

以下程式碼範例顯示如何使用 `retrieve-domain-auth-code`。

AWS CLI

取得網域的授權碼，以便您可以將網域轉移到其他註冊商

下列`retrieve-domain-auth-code`命令會取得 example.com 網域的目前授權碼。當您想要將網域轉移到該註冊商時，請將此值提供給另一個網域註冊商。

此命令僅在 `us-east-1`區域中執行。如果您的預設區域設定為 `us-east-1`，您可以省略 `region` 參數。

```
aws route53domains retrieve-domain-auth-code \
    --region us-east-1 \
    --domain-name example.com
```

輸出：

```
{
    "AuthCode": ")o!v3dJeXampLe"
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的將網域從 Amazon Route 53 轉移到另一個註冊商。 Amazon Route 53

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RetrieveDomainAuthCode。

`transfer-domain`

以下程式碼範例顯示如何使用 `transfer-domain`。

AWS CLI

將網域轉移到 Amazon Route 53

下列transfer-domain命令會使用 JSON 格式檔案 提供的參數，將網域傳輸到 Route 53C:\temp\transfer-domain.json。

此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains transfer-domain \
    --region us-east-1 \
    --cli-input-json file://C:\temp\transfer-domain.json
```

transfer-domain.json 的內容：

```
{
    "DomainName": "example.com",
    "DurationInYears": 1,
    "Nameservers": [
        {
            "Name": "ns-2048.awsdns-64.com"
        },
        {
            "Name": "ns-2049.awsdns-65.net"
        },
        {
            "Name": "ns-2050.awsdns-66.org"
        },
        {
            "Name": "ns-2051.awsdns-67.co.uk"
        }
    ],
    "AuthCode": ")o!v3dJeXampLe",
    "AutoRenew": true,
    "AdminContact": {
        "FirstName": "Martha",
        "LastName": "Rivera",
        "ContactType": "PERSON",
        "OrganizationName": "Example",
        "AddressLine1": "1 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "mrivera@example.com"
```

```
    },
    "RegistrantContact": {
        "FirstName": "Li",
        "LastName": "Juan",
        "ContactType": "PERSON",
        "OrganizationName": "Example",
        "AddressLine1": "1 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "ljuan@example.com"
    },
    "TechContact": {
        "FirstName": "Mateo",
        "LastName": "Jackson",
        "ContactType": "PERSON",
        "OrganizationName": "Example",
        "AddressLine1": "1 Main Street",
        "City": "Anytown",
        "State": "WA",
        "CountryCode": "US",
        "ZipCode": "98101",
        "PhoneNumber": "+1.8005551212",
        "Email": "mjackson@example.com"
    },
    "PrivacyProtectAdminContact": true,
    "PrivacyProtectRegistrantContact": true,
    "PrivacyProtectTechContact": true
}
```

輸出：

```
{
    "OperationId": "b114c44a-9330-47d1-a6e8-a0b11example"
}
```

若要確認操作成功，您可以執行 `get-operation-detail`。如需詳細資訊，請參閱 get-operation-detail 。

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的將網域註冊轉移至 Amazon Route 53。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TransferDomain。

update-domain-contact-privacy

以下程式碼範例顯示如何使用 update-domain-contact-privacy。

AWS CLI

更新網域聯絡人的隱私權設定

下列update-domain-contact-privacy命令會關閉 example.com 網域管理聯絡人的隱私權保護。此命令僅在 us-east-1區域中執行。

如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains update-domain-contact-privacy \
    --region us-east-1 \
    --domain-name example.com \
    --no-admin-privacy
```

輸出：

```
{
    "OperationId": "b3a219e9-d801-4244-b533-b7256example"
}
```

若要確認操作成功，您可以執行 get-operation-detail。如需詳細資訊，請參閱 get-operation-detail 。

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的啟用或停用網域聯絡資訊的隱私權保護。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDomainContactPrivacy。

update-domain-contact

以下程式碼範例顯示如何使用 update-domain-contact。

AWS CLI

更新網域的聯絡資訊

下列update-domain-contact命令會更新網域的聯絡資訊，從 JSON 格式檔案 取得參數C:\temp\update-domain-contact.json。

此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains update-domain-contact \
    --region us-east-1 \
    --cli-input-json file://C:\temp\update-domain-contact.json
```

update-domain-contact.json 的內容：

```
{
    "AdminContact": {
        "AddressLine1": "101 Main Street",
        "AddressLine2": "Suite 1a",
        "City": "Seattle",
        "ContactType": "COMPANY",
        "CountryCode": "US",
        "Email": "w.xiulan@example.com",
        "FirstName": "Wang",
        "LastName": "Xiulan",
        "OrganizationName": "Example",
        "PhoneNumber": "+1.8005551212",
        "State": "WA",
        "ZipCode": "98101"
    },
    "DomainName": "example.com",
    "RegistrantContact": {
        "AddressLine1": "101 Main Street",
        "AddressLine2": "Suite 1a",
        "City": "Seattle",
        "ContactType": "COMPANY",
        "CountryCode": "US",
        "Email": "w.xiulan@example.com",
        "FirstName": "Wang",
        "LastName": "Xiulan",
        "OrganizationName": "Example",
        "PhoneNumber": "+1.8005551212",
        "State": "WA",
        "ZipCode": "98101"
    },
    "TechContact": {
```

```
        "AddressLine1": "101 Main Street",
        "AddressLine2": "Suite 1a",
        "City": "Seattle",
        "ContactType": "COMPANY",
        "CountryCode": "US",
        "Email": "w.xiulan@example.com",
        "FirstName": "Wang",
        "LastName": "Xiulan",
        "OrganizationName": "Example",
        "PhoneNumber": "+1.8005551212",
        "State": "WA",
        "ZipCode": "98101"
    }
}
```

輸出：

```
{
    "OperationId": "b3a219e9-d801-4244-b533-b7256example"
}
```

若要確認操作成功，您可以執行 get-domain-detail 。如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的更新網域的聯絡資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDomainContact。

update-domain-nameservers

以下程式碼範例顯示如何使用 update-domain-nameservers。

AWS CLI

更新網域的名稱伺服器

下列update-domain-nameservers命令會更新網域的名稱伺服器。

此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains update-domain-nameservers \
    --region us-east-1 \
```

```
    --domain-name example.com \
    --
nameservers Name=ns-1.awsdns-01.org Name=ns-2.awsdns-02.co.uk Name=ns-3.awsdns-03.net Name=n
```

輸出：

```
{
    "OperationId": "f1691ec4-0e7a-489e-82e0-b19d3example"
}
```

若要確認操作成功，您可以執行 get-domain-detail 。

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的新增或變更網域的名稱伺服器和 Glue 記錄。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDomainNameservers。

update-tags-for-domain

以下程式碼範例顯示如何使用 update-tags-for-domain。

AWS CLI

新增或更新網域的標籤

下列update-tags-for-domain命令會新增或更新 example.com 網域的兩個金鑰和對應值。若要更新索引鍵的值，只需包含索引鍵和新值。您一次只能在一個網域中新增或更新標籤。

此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains update-tags-for-domain \
    --region us-east-1 \
    --domain-name example.com \
    --tags-to-update "Key=key1,Value=value1" "Key=key2,Value=value2"
```

此命令不會產生輸出。若要確認標籤已新增或更新，您可以執行 list-tags-for-domain 。

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的標記 Amazon Route 53 資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateTagsForDomain。

view-billing

以下程式碼範例顯示如何使用 view-billing。

AWS CLI

取得目前 AWS 帳戶的網域註冊費用帳單資訊

下列view-billing命令會傳回目前帳戶在 Unix 時間 2018 年 1 月 1 日 (1514764800 和 Unix 時間 2019 年 12 月 31 日 (1577836800 午夜期間的所有網域相關帳單記錄）。

此命令僅在 us-east-1區域中執行。如果您的預設區域設定為 us-east-1，您可以省略 region 參數。

```
aws route53domains view-billing \
    --region us-east-1 \
    --start-time 1514764800 \
    --end-time 1577836800
```

輸出：

```
{
    "BillingRecords": [
        {
            "DomainName": "example.com",
            "Operation": "RENEW_DOMAIN",
            "InvoiceId": "149962827",
            "BillDate": 1536618063.181,
            "Price": 12.0
        },
        {
            "DomainName": "example.com",
            "Operation": "RENEW_DOMAIN",
            "InvoiceId": "290913289",
            "BillDate": 1568162630.884,
            "Price": 12.0
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Route 53 API 參考》中的 ViewBilling。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ViewBilling。

使用 的 Route 53 Profiles 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Route 53 Profiles 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-profile`

以下程式碼範例顯示如何使用 `associate-profile`。

AWS CLI

建立設定檔的關聯

下列`associate-profile`範例會將設定檔與 VPC 建立關聯。

```
aws route53profiles associate-profile \
    --name test-association \
    --profile-id rp-4987774726example \
    --resource-id vpc-0af3b96b3example
```

輸出：

```
{
    "ProfileAssociation": {
        "CreationTime": 1710851336.527,
        "Id": "rpassoc-489ce212fexample",
        "ModificationTime": 1710851336.527,
        "Name": "test-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceId": "vpc-0af3b96b3example",
        "Status": "CREATING",
```

```
        "StatusMessage": "Creating Profile Association"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的使用設定檔。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateProfile。

associate-resource-to-profile

以下程式碼範例顯示如何使用 associate-resource-to-profile。

AWS CLI

將資源與設定檔建立關聯

下列associate-resource-to-profile範例會將優先順序為 102 的 DNS 防火牆規則群組與設定檔建立關聯。

```
aws route53profiles associate-resource-to-profile \
    --name test-resource-association \
    --profile-id rp-4987774726example \
    --resource-arn arn:aws:route53resolver:us-east-1:123456789012:firewall-rule-
group/rslvr-frg-cfe7f72example \
    --resource-properties "{\"priority\": 102}"
```

輸出：

```
{
    "ProfileResourceAssociation": {
        "CreationTime": 1710851216.613,
        "Id": "rpr-001913120a7example",
        "ModificationTime": 1710851216.613,
        "Name": "test-resource-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-
rule-group/rslvr-frg-cfe7f72example",
        "ResourceProperties": "{\"priority\":102}",
        "ResourceType": "FIREWALL_RULE_GROUP",
        "Status": "UPDATING",
        "StatusMessage": "Updating the Profile to DNS Firewall rule group
 association"
```

```
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateResourceToProfile。

create-profile

以下程式碼範例顯示如何使用 create-profile。

AWS CLI

建立設定檔

下列create-profile範例會建立設定檔。

```
aws route53profiles create-profile \
    --name test
```

輸出：

```
{
    "Profile": {
        "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-6ffe47d5example",
        "ClientToken": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
        "CreationTime": 1710850903.578,
        "Id": "rp-6ffe47d5example",
        "ModificationTime": 1710850903.578,
        "Name": "test",
        "OwnerId": "123456789012",
        "ShareStatus": "NOT_SHARED",
        "Status": "COMPLETE",
        "StatusMessage": "Created Profile"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateProfile。

delete-profile

以下程式碼範例顯示如何使用 delete-profile。

AWS CLI

刪除設定檔

下列delete-profile範例會刪除設定檔。

```
aws route53profiles delete-profile \
    --profile-id rp-6ffe47d5example
```

輸出：

```
{
    "Profile": {
        "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-6ffe47d5example",
        "ClientToken": "0a15fec0-05d9-4f78-bec0-EXAMPLE11111",
        "CreationTime": 1710850903.578,
        "Id": "rp-6ffe47d5example",
        "ModificationTime": 1710850903.578,
        "Name": "test",
        "OwnerId": "123456789012",
        "ShareStatus": "NOT_SHARED",
        "Status": "DELETED",
        "StatusMessage": "Deleted Profile"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteProfile。

disassociate-profile

以下程式碼範例顯示如何使用 disassociate-profile。

AWS CLI

取消設定檔的關聯

下列disassociate-profile範例會取消設定檔與 VPC 的關聯。

```
aws route53profiles disassociate-profile \
    --profile-id rp-4987774726example \
    --resource-id vpc-0af3b96b3example
```

輸出：

```
{
    "ProfileAssociation": {
        "CreationTime": 1710851336.527,
        "Id": "rpassoc-489ce212fexample",
        "ModificationTime": 1710851401.362,
        "Name": "test-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceId": "vpc-0af3b96b3example",
        "Status": "DELETING",
        "StatusMessage": "Deleting Profile Association"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateProfile。

disassociate-resource-from-profile

以下程式碼範例顯示如何使用 disassociate-resource-from-profile。

AWS CLI

取消資源與設定檔的關聯

下列disassociate-resource-from-profile範例會取消 DNS 防火牆規則群組與設定檔的關聯。

```
aws route53profiles disassociate-resource-from-profile \
    --profile-id rp-4987774726example \
    --resource-arn arn:aws:route53resolver:us-east-1:123456789012:firewall-rule-group/rslvr-frg-cfe7f72example
```

輸出：

```
{
    "ProfileResourceAssociation": {
        "CreationTime": 1710851216.613,
        "Id": "rpr-001913120a7example",
        "ModificationTime": 1710852624.36,
```

```
        "Name": "test-resource-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-
rule-group/rslvr-frg-cfe7f72example",
        "ResourceProperties": "{\"priority\":105}",
        "ResourceType": "FIREWALL_RULE_GROUP",
        "Status": "DELETING",
        "StatusMessage": "Deleting the Profile to DNS Firewall rule group
 association"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateResourceFromProfile。

get-profile-association

以下程式碼範例顯示如何使用 get-profile-association。

AWS CLI

取得設定檔關聯的相關資訊

下列get-profile-association會傳回指定設定檔關聯的相關資訊。

```
aws route53profiles get-profile-association \
    --profile-association-id rpassoc-489ce212fexample
```

輸出：

```
{
    "ProfileAssociation": {
        "CreationTime": 1709338817.148,
        "Id": "rrpassoc-489ce212fexample",
        "ModificationTime": 1709338974.772,
        "Name": "test-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceId": "vpc-0af3b96b3example",
        "Status": "COMPLETE",
        "StatusMessage": "Created Profile Association"
    }
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetProfileAssociation。

get-profile-resource-association

以下程式碼範例顯示如何使用 get-profile-resource-association。

AWS CLI

取得與設定檔相關聯之資源的相關資訊

以下內容會將指定資源關聯的相關資訊get-profile-resource-association傳回至設定檔。

```
aws route53profiles get-profile-resource-association \
    --profile-resource-association-id rpr-001913120a7example
```

輸出：

```
{
    "ProfileResourceAssociation": {
        "CreationTime": 1710851216.613,
        "Id": "rpr-001913120a7example",
        "ModificationTime": 1710852303.798,
        "Name": "test-resource-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-
rule-group/rslvr-frg-cfe7f72example",
        "ResourceProperties": "{\"priority\":105}",
        "ResourceType": "FIREWALL_RULE_GROUP",
        "Status": "COMPLETE",
        "StatusMessage": "Completed creation of Profile to DNS Firewall rule group
 association"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetProfileResourceAssociation。

get-profile

以下程式碼範例顯示如何使用 get-profile。

AWS CLI

取得設定檔的相關資訊

下列get-profile會傳回指定設定檔的相關資訊。

```
aws route53profiles get-profile \
    --profile-id rp-4987774726example
```

輸出：

```
{
    "Profile": {
        "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-4987774726example",
        "ClientToken": "0cbc5ae7-4921-4204-bea9-EXAMPLE11111",
        "CreationTime": 1710851044.288,
        "Id": "rp-4987774726example",
        "ModificationTime": 1710851044.288,
        "Name": "test",
        "OwnerId": "123456789012",
        "ShareStatus": "NOT_SHARED",
        "Status": "COMPLETE",
        "StatusMessage": "Created Profile"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetProfile。

list-profile-associations

以下程式碼範例顯示如何使用 list-profile-associations。

AWS CLI

列出設定檔關聯

下列list-profile-associations列出您 AWS 帳戶中的設定檔關聯。

```
aws route53profiles list-profile-associations
```

輸出：

```
{
    "ProfileAssociations": [
        {
            "CreationTime": 1709338817.148,
            "Id": "rpassoc-489ce212fexample",
            "ModificationTime": 1709338974.772,
            "Name": "test-association",
            "OwnerId": "123456789012",
            "ProfileId": "rp-4987774726example",
            "ResourceId": "vpc-0af3b96b3example",
            "Status": "COMPLETE",
            "StatusMessage": "Created Profile Association"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListProfileAssociations。

list-profile-resource-associations

以下程式碼範例顯示如何使用 list-profile-resource-associations。

AWS CLI

列出設定檔資源關聯

以下list-profile-resource-associations列出指定設定檔的設定檔資源關聯。

```
aws route53profiles list-profile-resource-associations \
    --profile-id rp-4987774726example
```

輸出：

```
{
     "ProfileResourceAssociations": [
         {
             "CreationTime": 1710851216.613,
             "Id": "rpr-001913120a7example",
             "ModificationTime": 1710851216.613,
```

```
            "Name": "test-resource-association",
            "OwnerId": "123456789012",
            "ProfileId": "rp-4987774726example",
            "ResourceArn": "arn:aws:route53resolver:us-
east-1:123456789012:firewall-rule-group/rslvr-frg-cfe7f72example",
            "ResourceProperties": "{\"priority\":102}",
            "ResourceType": "FIREWALL_RULE_GROUP",
            "Status": "COMPLETE",
            "StatusMessage": "Completed creation of Profile to DNS Firewall rule
 group association"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListProfileResourceAssociations。

list-profiles

以下程式碼範例顯示如何使用 list-profiles。

AWS CLI

列出設定檔

以下list-profiles列出您 AWS 帳戶中的設定檔，並顯示這些設定檔的其他資訊。

```
aws route53profiles list-profiles
```

輸出：

```
{
    "ProfileSummaries": [
        {
            "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-4987774726example",
            "Id": "rp-4987774726example",
            "Name": "test",
            "ShareStatus": "NOT_SHARED"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListProfiles。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出資源的標籤

下列list-tags-for-resource列出指定資源的標籤。

```
aws route53profiles list-tags-for-resource \
    --resource-arn arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-4987774726example
```

輸出：

```
{
    "Tags": {
        "my-key-2": "my-value-2",
        "my-key-1": "my-value-1"
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

update-profile-resource-association

以下程式碼範例顯示如何使用 update-profile-resource-association。

AWS CLI

更新與設定檔相關聯的資源

以下update-profile-resource-association更新與 設定檔相關聯的 DNS 防火牆規則群組的優先順序。

```
aws route53profiles update-profile-resource-association \
    --profile-resource-association-id rpr-001913120a7example \
    --resource-properties "{\"priority\": 105}"
```

輸出：

```
{
    "ProfileResourceAssociation": {
        "CreationTime": 1710851216.613,
        "Id": "rpr-001913120a7example",
        "ModificationTime": 1710852303.798,
        "Name": "test-resource-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-
rule-group/rslvr-frg-cfe7f72example",
        "ResourceProperties": "{\"priority\":105}",
        "ResourceType": "FIREWALL_RULE_GROUP",
        "Status": "UPDATING",
        "StatusMessage": "Updating the Profile to DNS Firewall rule group
 association"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateProfileResourceAssociation。

使用 的 Route 53 Resolver 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Route 53 Resolver 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-firewall-rule-group`

以下程式碼範例顯示如何使用 `associate-firewall-rule-group`。

AWS CLI

將防火牆規則群組與 VPC 建立關聯

下列associate-firewall-rule-group範例會將 DNS 防火牆規則群組與 Amazon VPC 建立關聯。

```
aws route53resolver associate-firewall-rule-group \
    --name test-association \
    --firewall-rule-group-id rslvr-frg-47f93271fexample \
    --vpc-id vpc-31e92222 \
    --priority 101
```

輸出：

```
{
    "FirewallRuleGroupAssociation": {
        "Id": "rslvr-frgassoc-57e8873d7example",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-
association/rslvr-frgassoc-57e8873d7example",
        "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
        "VpcId": "vpc-31e92222",
        "Name": "test-association",
        "Priority": 101,
        "MutationProtection": "DISABLED",
        "Status": "UPDATING",
        "StatusMessage": "Creating Firewall Rule Group Association",
        "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
        "CreationTime": "2021-05-25T21:47:48.755768Z",
        "ModificationTime": "2021-05-25T21:47:48.755768Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理 VPC 和 Route 53 Resolver DNS 防火牆規則群組之間的關聯。 Amazon Route 53

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateFirewallRuleGroup。

associate-resolver-endpoint-ip-address

以下程式碼範例顯示如何使用 associate-resolver-endpoint-ip-address。

AWS CLI

將另一個 IP 地址與解析程式端點建立關聯

下列associate-resolver-endpoint-ip-address範例會將另一個 IP 地址與傳入解析程式端點建立關聯。如果您只指定子網路 ID 並從 --ip-address 參數省略 IP 地址，Resolver 會從指定子網路中的可用 IP 地址中為您選擇 IP 地址。

```
aws route53resolver associate-resolver-endpoint-ip-address \
    --resolver-endpoint-id rslvr-in-497098ad5example \
    --ip-address="SubnetId=subnet-12d8exam,Ip=192.0.2.118"
```

輸出：

```
{
    "ResolverEndpoint": {
        "Id": "rslvr-in-497098ad5example",
        "CreatorRequestId": "AWSConsole.25.0123456789",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-in-497098ad5example",
        "Name": "my-inbound-endpoint",
        "SecurityGroupIds": [
            "sg-05cd7b25d6example"
        ],
        "Direction": "INBOUND",
        "IpAddressCount": 3,
        "HostVPCId": "vpc-304bexam",
        "Status": "UPDATING",
        "StatusMessage": "Updating the Resolver Endpoint",
        "CreationTime": "2020-01-02T23:25:45.538Z",
        "ModificationTime": "2020-01-02T23:25:45.538Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的建立或編輯傳入端點時指定的值。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateResolverEndpointIpAddress。

associate-resolver-rule

以下程式碼範例顯示如何使用 associate-resolver-rule。

AWS CLI

將解析程式規則與 VPC 建立關聯

下列associate-resolver-rule範例會將解析程式規則與 Amazon VPC 建立關聯。在您執行命令之後，Resolver 會根據規則中的設定，開始將 DNS 查詢轉送到您的網路，例如轉送的查詢網域名稱。

```
aws route53resolver associate-resolver-rule \
    --name my-resolver-rule-association \
    --resolver-rule-id rslvr-rr-42b60677c0example \
    --vpc-id vpc-304bexam
```

輸出：

```
{
    "ResolverRuleAssociation": {
        "Id": "rslvr-rrassoc-d61cbb2c8bexample",
        "ResolverRuleId": "rslvr-rr-42b60677c0example",
        "Name": "my-resolver-rule-association",
        "VPCId": "vpc-304bexam",
        "Status": "CREATING",
        "StatusMessage": "[Trace id: 1-5dc5a8fa-ec2cc480d2ef07617example] Creating
 the association."
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的轉送傳出 DNS 查詢到您的網路。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateResolverRule。

create-firewall-domain-list

以下程式碼範例顯示如何使用 create-firewall-domain-list。

AWS CLI

建立 Route 53 Resolver DNS Firewall 網域清單

下列create-firewall-domain-list範例會在您的帳戶中建立名為 test 的 Route 53 Resolver DNS Firewall 網域清單 AWS 。

```
aws route53resolver create-firewall-domain-list \
    --creator-request-id my-request-id \
    --name test
```

輸出：

```
{
    "FirewallDomainList": {
        "Id": "rslvr-fdl-d61cbb2cbexample",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-domain-list/
rslvr-fdl-d61cbb2cbexample",
        "Name": "test",
        "DomainCount": 0,
        "Status": "COMPLETE",
        "StatusMessage": "Created Firewall Domain List",
        "CreatorRequestId": "my-request-id",
        "CreationTime": "2021-05-25T15:55:51.115365Z",
        "ModificationTime": "2021-05-25T15:55:51.115365Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理您自己的網域清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFirewallDomainList。

create-firewall-rule-group

以下程式碼範例顯示如何使用 create-firewall-rule-group。

AWS CLI

建立防火牆規則群組

下列create-firewall-rule-group範例會建立 DNS 防火牆規則群組。

```
aws route53resolver create-firewall-rule-group \
    --creator-request-id my-request-id \
    --name test
```

輸出：

```
{
```

```
    "FirewallRuleGroup": {
        "Id": "rslvr-frg-47f93271fexample",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group/
rslvr-frg-47f93271fexample",
        "Name": "test",
        "RuleCount": 0,
        "Status": "COMPLETE",
        "StatusMessage": "Created Firewall Rule Group",
        "OwnerId": "123456789012",
        "CreatorRequestId": "my-request-id",
        "ShareStatus": "NOT_SHARED",
        "CreationTime": "2021-05-25T18:59:26.490017Z",
        "ModificationTime": "2021-05-25T18:59:26.490017Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理 DNS 防火牆中的規則群組和規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFirewallRuleGroup。

create-firewall-rule

以下程式碼範例顯示如何使用 create-firewall-rule。

AWS CLI

建立防火牆規則

下列create-firewall-rule範例會在 DNS 防火牆規則中為 DNS 防火牆網域清單中列出的網域建立防火牆規則。

```
aws route53resolver create-firewall-rule \
    --name allow-rule \
    --firewall-rule-group-id rslvr-frg-47f93271fexample \
    --firewall-domain-list-id rslvr-fdl-9e956e9ffexample \
    --priority 101 \
    --action ALLOW
```

輸出：

```
{
```

```
    "FirewallRule": {
        "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
        "FirewallDomainListId": "rslvr-fdl-9e956e9ffexample",
        "Name": "allow-rule",
        "Priority": 101,
        "Action": "ALLOW",
        "CreatorRequestId": "d81e3fb7-020b-415e-939f-EXAMPLE11111",
        "CreationTime": "2021-05-25T21:44:00.346093Z",
        "ModificationTime": "2021-05-25T21:44:00.346093Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理 DNS 防火牆中的規則群組和規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFirewallRule。

`create-resolver-endpoint`

以下程式碼範例顯示如何使用 `create-resolver-endpoint`。

AWS CLI

建立傳入解析程式端點

下列`create-resolver-endpoint`範例會建立傳入 Resolver 端點。您可以使用相同的命令來建立傳入和傳出端點。

aws route53resolver create-resolver-endpoint --name my-inbound-endpoint --creator-request-id 2020-01-01-18：47 --security-group-ids "sg-f62bexam" --direction INBOUND --ip-addresses SubnetId=subnet-ba47exam，Ip=192.0.2.255 SubnetId=subnet-12d8exam，Ip=192.0.2.254

輸出：

```
{
    "ResolverEndpoint": {
        "Id": "rslvr-in-f9ab8a03f1example",
        "CreatorRequestId": "2020-01-01-18:47",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-in-f9ab8a03f1example",
        "Name": "my-inbound-endpoint",
        "SecurityGroupIds": [
            "sg-f62bexam"
```

```
        ],
        "Direction": "INBOUND",
        "IpAddressCount": 2,
        "HostVPCId": "vpc-304examp",
        "Status": "CREATING",
        "StatusMessage": "[Trace id: 1-5dc1ff84-f3477826e4a190025example] Creating
 the Resolver Endpoint",
        "CreationTime": "2020-01-01T23:02:29.583Z",
        "ModificationTime": "2020-01-01T23:02:29.583Z"
    }
}
```

建立傳出解析程式端點

下列create-resolver-endpoint範例會使用 JSON 格式文件 中的值來建立傳出解析程式端點create-outbound-resolver-endpoint.json。

```
aws route53resolver create-resolver-endpoint \
    --cli-input-json file://c:\temp\create-outbound-resolver-endpoint.json
```

create-outbound-resolver-endpoint.json 的內容：

```
{
   "CreatorRequestId": "2020-01-01-18:47",
   "Direction": "OUTBOUND",
   "IpAddresses": [
      {
         "Ip": "192.0.2.255",
         "SubnetId": "subnet-ba47exam"
      },
      {
         "Ip": "192.0.2.254",
         "SubnetId": "subnet-12d8exam"
      }
   ],
   "Name": "my-outbound-endpoint",
   "SecurityGroupIds": [ "sg-05cd7b25d6example" ],
   "Tags": [
      {
         "Key": "my-key-name",
         "Value": "my-key-value"
      }
   ]
```

```
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的在 VPC 和網路之間解析 DNS 查詢。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateResolverEndpoint。

create-resolver-rule

以下程式碼範例顯示如何使用 create-resolver-rule。

AWS CLI

建立解析程式規則

下列create-resolver-rule範例會建立解析程式轉送規則。此規則使用傳出端點 rslvr-out-d5e5920e37example 將 的 DNS 查詢轉送example.com至 IP 地址 10.24.8.75 和 10.24.8.156。

```
aws route53resolver create-resolver-rule \
    --creator-request-id 2020-01-02-18:47 \
    --domain-name example.com \
    --name my-rule \
    --resolver-endpoint-id rslvr-out-d5e5920e37example \
    --rule-type FORWARD \
    --target-ips "Ip=10.24.8.75" "Ip=10.24.8.156"
```

輸出：

```
{
    "ResolverRule": {
        "Status": "COMPLETE",
        "RuleType": "FORWARD",
        "ResolverEndpointId": "rslvr-out-d5e5920e37example",
        "Name": "my-rule",
        "DomainName": "example.com.",
        "CreationTime": "2022-05-10T21:35:30.923187Z",
        "TargetIps": [
            {
                "Ip": "10.24.8.75",
                "Port": 53
            },
            {
```

```
                "Ip": "10.24.8.156",
                "Port": 53
            }
        ],
        "CreatorRequestId": "2022-05-10-16:33",
        "ModificationTime": "2022-05-10T21:35:30.923187Z",
        "ShareStatus": "NOT_SHARED",
        "Arn": "arn:aws:route53resolver:us-east-1:111117012054:resolver-rule/rslvr-
rr-b1e0b905e93611111",
        "OwnerId": "111111111111",
        "Id": "rslvr-rr-rslvr-rr-b1e0b905e93611111",
        "StatusMessage": "[Trace id: 1-22222222-3e56afcc71a3724664f22e24]
 Successfully created Resolver Rule."
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateResolverRule。

delete-firewall-domain-list

以下程式碼範例顯示如何使用 delete-firewall-domain-list。

AWS CLI

刪除 Route 53 Resolver DNS Firewall 網域清單

下列delete-firewall-domain-list範例會刪除您 AWS 帳戶中名為 test 的 Route 53 Resolver DNS Firewall 網域清單。

```
aws route53resolver delete-firewall-domain-list \
    --firewall-domain-list-id rslvr-fdl-9e956e9ffexample
```

輸出：

```
{
    "FirewallDomainList": {
        "Id": "rslvr-fdl-9e956e9ffexample",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-domain-list/
rslvr-fdl-9e956e9ffexample",
        "Name": "test",
        "DomainCount": 6,
        "Status": "DELETING",
```

```
        "StatusMessage": "Deleting the Firewall Domain List",
        "CreatorRequestId": "my-request-id",
        "CreationTime": "2021-05-25T15:55:51.115365Z",
        "ModificationTime": "2021-05-25T18:58:05.588024Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理您自己的網域清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFirewallDomainList。

delete-firewall-rule-group

以下程式碼範例顯示如何使用 delete-firewall-rule-group。

AWS CLI

刪除防火牆規則群組

下列delete-firewall-rule-group範例會刪除防火牆規則群組。

```
aws route53resolver delete-firewall-rule-group \
    --firewall-rule-group-id rslvr-frg-47f93271fexample
```

輸出：

```
{
    "FirewallRuleGroup": {
        "Id": "rslvr-frg-47f93271fexample",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group/
rslvr-frg-47f93271fexample",
        "Name": "test",
        "RuleCount": 0,
        "Status": "UPDATING",
        "StatusMessage": "Updating Firewall Rule Group",
        "OwnerId": "123456789012",
        "CreatorRequestId": "my-request-id",
        "ShareStatus": "NOT_SHARED",
        "CreationTime": "2021-05-25T18:59:26.490017Z",
        "ModificationTime": "2021-05-25T21:51:53.028688Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理 DNS 防火牆中的規則群組和規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFirewallRuleGroup。

delete-firewall-rule

以下程式碼範例顯示如何使用 delete-firewall-rule。

AWS CLI

刪除防火牆規則

下列delete-firewall-rule範例會刪除指定的防火牆規則。

```
aws route53resolver delete-firewall-rule \
    --firewall-rule-group-id rslvr-frg-47f93271fexample \
    --firewall-domain-list-id rslvr-fdl-9e956e9ffexample
```

輸出：

```
{
    "FirewallRule": {
        "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
        "FirewallDomainListId": "rslvr-fdl-9e956e9ffexample",
        "Name": "allow-rule",
        "Priority": 102,
        "Action": "ALLOW",
        "CreatorRequestId": "d81e3fb7-020b-415e-939f-EXAMPLE11111",
        "CreationTime": "2021-05-25T21:44:00.346093Z",
        "ModificationTime": "2021-05-25T21:45:59.611600Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理 DNS 防火牆中的規則群組和規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFirewallRule。

delete-resolver-endpoint

以下程式碼範例顯示如何使用 delete-resolver-endpoint。

AWS CLI

刪除解析程式端點

下列delete-resolver-endpoint範例會刪除指定的端點。

重要：如果您刪除傳入端點，來自您網路的 DNS 查詢不會再轉送至端點中指定之 VPC 中的解析程式。如果您刪除傳出端點，則針對指定已刪除傳出端點的那些規則，Resolver 會停止將 DNS 查詢從您的 VPC 轉送到您的網路。

```
aws route53resolver delete-resolver-endpoint \
    --resolver-endpoint-id rslvr-in-497098ad59example
```

輸出：

```
{
    "ResolverEndpoint": {
        "Id": "rslvr-in-497098ad59example",
        "CreatorRequestId": "AWSConsole.25.157290example",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-in-497098ad59example",
        "Name": "my-inbound-endpoint",
        "SecurityGroupIds": [
            "sg-05cd7b25d6example"
        ],
        "Direction": "INBOUND",
        "IpAddressCount": 5,
        "HostVPCId": "vpc-304bexam",
        "Status": "DELETING",
        "StatusMessage": "[Trace id: 1-5dc5b658-811b5be0922bbc382example] Deleting
 ResolverEndpoint.",
        "CreationTime": "2020-01-01T23:25:45.538Z",
        "ModificationTime": "2020-01-02T23:25:45.538Z"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteResolverEndpoint。

delete-resolver-rule

以下程式碼範例顯示如何使用 delete-resolver-rule。

AWS CLI

刪除解析程式規則

下列delete-resolver-rule範例會刪除指定的規則。

請注意，如果規則與任何 VPCs相關聯，您必須先取消規則與 VPCs關聯，才能將其刪除。

```
aws route53resolver delete-resolver-rule \
    --resolver-rule-id rslvr-rr-5b3809426bexample
```

輸出：

```
{
    "ResolverRule": {
        "Id": "rslvr-rr-5b3809426bexample",
        "CreatorRequestId": "2020-01-03-18:47",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/rslvr-
rr-5b3809426bexample",
        "DomainName": "zenith.example.com.",
        "Status": "DELETING",
        "StatusMessage": "[Trace id: 1-5dc5e05b-602e67b052cb74f05example] Deleting
 Resolver Rule.",
        "RuleType": "FORWARD",
        "Name": "my-resolver-rule",
        "TargetIps": [
            {
                "Ip": "192.0.2.50",
                "Port": 53
            }
        ],
        "ResolverEndpointId": "rslvr-out-d5e5920e3example",
        "OwnerId": "111122223333",
        "ShareStatus": "NOT_SHARED"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteResolverRule。

disassociate-firewall-rule-group

以下程式碼範例顯示如何使用 disassociate-firewall-rule-group。

AWS CLI

取消防火牆規則群組與 VPC 的關聯

下列disassociate-firewall-rule-group範例會取消 DNS 防火牆規則群組與 Amazon VPC 的關聯。

```
aws route53resolver disassociate-firewall-rule-group \
    --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example
```

輸出：

```
{
    "FirewallRuleGroupAssociation": {
        "Id": "rslvr-frgassoc-57e8873d7example",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-
association/rslvr-frgassoc-57e8873d7example",
        "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
        "VpcId": "vpc-31e92222",
        "Name": "test-association",
        "Priority": 103,
        "MutationProtection": "DISABLED",
        "Status": "DELETING",
        "StatusMessage": "Deleting the Firewall Rule Group Association",
        "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
        "CreationTime": "2021-05-25T21:47:48.755768Z",
        "ModificationTime": "2021-05-25T21:51:02.377887Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理 VPC 和 Route 53 Resolver DNS 防火牆規則群組之間的關聯。 Amazon Route 53

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateFirewallRuleGroup。

disassociate-resolver-endpoint-ip-address

以下程式碼範例顯示如何使用 disassociate-resolver-endpoint-ip-address。

AWS CLI

取消 IP 地址與解析程式端點的關聯

下列disassociate-resolver-endpoint-ip-address範例會從指定的解析程式傳入或傳出端點移除 IP 地址。

注意 端點必須至少有兩個 IP 地址。如果端點目前只有兩個 IP 地址，而且您想要將一個地址取代為另一個地址，您必須先使用 associate-resolver-endpoint-ip-address 來關聯新的 IP 地址。然後，您可以取消其中一個原始 IP 地址與端點的關聯。

```
aws route53resolver disassociate-resolver-endpoint-ip-address \
    --resolver-endpoint-id rslvr-in-f9ab8a03f1example \
    --ip-address="SubnetId=subnet-12d8a459,Ip=172.31.40.121"
```

輸出：

```
{
    "ResolverEndpoint": {
        "Id": "rslvr-in-f9ab8a03f1example",
        "CreatorRequestId": "2020-01-01-18:47",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-in-f9ab8a03f1example",
        "Name": "my-inbound-endpoint",
        "SecurityGroupIds": [
            "sg-f62bexam"
        ],
        "Direction": "INBOUND",
        "IpAddressCount": 3,
        "HostVPCId": "vpc-304bexam",
        "Status": "UPDATING",
        "StatusMessage": "Updating the Resolver Endpoint",
        "CreationTime": "2020-01-01T23:02:29.583Z",
        "ModificationTime": "2020-01-05T23:02:29.583Z"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateResolverEndpointIpAddress。

disassociate-resolver-rule

以下程式碼範例顯示如何使用 disassociate-resolver-rule。

AWS CLI

取消解析程式規則與 Amazon VPC 的關聯

下列disassociate-resolver-rule範例會移除指定解析程式規則與指定 VPC 之間的關聯。在下列情況下，您可以取消規則與 VPC 的關聯：

對於源自此 VPC 的 DNS 查詢，您希望 Resolver 停止將查詢轉送至您的網路，以取得規則中指定的網域名稱。您想要刪除轉送規則。如果規則目前與一或多個 VPC 相關聯，您必須先取消規則與所有 VPC 的關聯，才能刪除規則。

```
aws route53resolver disassociate-resolver-rule \
    --resolver-rule-id rslvr-rr-4955cb98ceexample \
    --vpc-id vpc-304bexam
```

輸出：

```
{
    "ResolverRuleAssociation": {
        "Id": "rslvr-rrassoc-322f4e8b9cexample",
        "ResolverRuleId": "rslvr-rr-4955cb98ceexample",
        "Name": "my-resolver-rule-association",
        "VPCId": "vpc-304bexam",
        "Status": "DELETING",
        "StatusMessage": "[Trace id: 1-5dc5ffa2-a26c38004c1f94006example] Deleting
 Association"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateResolverRule。

get-firewall-config

以下程式碼範例顯示如何使用 get-firewall-config。

AWS CLI

取得 VPC 的防火牆組態

下列get-firewall-config範例會擷取指定 VPC 的 DNS 防火牆行為。

```
aws route53resolver get-firewall-config \
```

```
    --resource-id vpc-31e92222
```

輸出：

```
{
    "FirewallConfig": {
        "Id": "rslvr-fc-86016850cexample",
        "ResourceId": "vpc-31e9222",
        "OwnerId": "123456789012",
        "FirewallFailOpen": "DISABLED"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的 DNS 防火牆 VPC 組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFirewallConfig。

get-firewall-domain-list

以下程式碼範例顯示如何使用 get-firewall-domain-list。

AWS CLI

若要取得 Route 53 Resolver DNS Firewall 網域清單

下列get-firewall-domain-list範例會使用您指定的 ID 擷取網域清單。

```
aws route53resolver get-firewall-domain-list \
    --firewall-domain-list-id rslvr-fdl-42b60677cexample
```

輸出：

```
{
    "FirewallDomainList": {
        "Id": "rslvr-fdl-9e956e9ffexample",
        "Arn": "arn:aws:route53resolver:us-west-2:123457689012:firewall-domain-list/
rslvr-fdl-42b60677cexample",
        "Name": "test",
        "DomainCount": 0,
        "Status": "COMPLETE",
        "StatusMessage": "Created Firewall Domain List",
        "CreatorRequestId": "my-request-id",
        "CreationTime": "2021-05-25T15:55:51.115365Z",
```

```
        "ModificationTime": "2021-05-25T15:55:51.115365Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理您自己的網域清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFirewallDomainList。

get-firewall-rule-group-association

以下程式碼範例顯示如何使用 get-firewall-rule-group-association。

AWS CLI

取得防火牆規則群組關聯

下列get-firewall-rule-group-association範例會擷取防火牆規則群組關聯。

```
aws route53resolver get-firewall-rule-group-association \
    --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example
```

輸出：

```
{
    "FirewallRuleGroupAssociation": {
        "Id": "rslvr-frgassoc-57e8873d7example",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-
association/rslvr-frgassoc-57e8873d7example",
        "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
        "VpcId": "vpc-31e92222",
        "Name": "test-association",
        "Priority": 101,
        "MutationProtection": "DISABLED",
        "Status": "COMPLETE",
        "StatusMessage": "Finished rule group association update",
        "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
        "CreationTime": "2021-05-25T21:47:48.755768Z",
        "ModificationTime": "2021-05-25T21:47:48.755768Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理 VPC 與 Route 53 Resolver DNS 防火牆規則群組之間的關聯。 Amazon Route 53

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFirewallRuleGroupAssociation。

get-firewall-rule-group-policy

以下程式碼範例顯示如何使用 get-firewall-rule-group-policy。

AWS CLI

取得 AWS IAM 政策

下列get-firewall-rule-group-policy範例取得用於共用指定規則群組的 AWS Identity and Access Management (AWS IAM) 政策。

```
aws route53resolver get-firewall-rule-group-policy \
    --arn arn:aws:route53resolver:us-west-2:AWS_ACCOUNT_ID:firewall-rule-group/
rslvr-frg-47f93271fexample
```

輸出：

```
{
    "FirewallRuleGroupPolicy": "{\"Version\":\"2012-10-17\",
\"Statement\":[{\"Sid\":\"test\",\"Effect\":\"Allow\",\"Principal
\":{\"AWS\":\"arn:aws:iam::AWS_ACCOUNT_ID:root\"},\"Action\":
[\"route53resolver:GetFirewallRuleGroup\",\"route53resolver:ListFirewallRuleGroups
\"],\"Resource\":\"arn:aws:route53resolver:us-east-1:AWS_ACCOUNT_ID:firewall-rule-
group/rslvr-frg-47f93271fexample\"}]}"
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理 DNS 防火牆中的規則群組和規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFirewallRuleGroupPolicy。

get-firewall-rule-group

以下程式碼範例顯示如何使用 get-firewall-rule-group。

AWS CLI

取得防火牆規則群組

下列get-firewall-rule-group範例會使用您提供的 ID 擷取 DNS 防火牆規則群組的相關資訊。

```
aws route53resolver get-firewall-rule-group \
    --firewall-rule-group-id rslvr-frg-47f93271fexample
```

輸出：

```
{
    "FirewallRuleGroup": {
        "Id": "rslvr-frg-47f93271fexample",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group/
rslvr-frg-47f93271fexample",
        "Name": "test",
        "RuleCount": 0,
        "Status": "COMPLETE",
        "StatusMessage": "Created Firewall Rule Group",
        "OwnerId": "123456789012",
        "CreatorRequestId": "my-request-id",
        "ShareStatus": "NOT_SHARED",
        "CreationTime": "2021-05-25T18:59:26.490017Z",
        "ModificationTime": "2021-05-25T18:59:26.490017Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理 DNS 防火牆中的規則群組和規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFirewallRuleGroup。

get-resolver-endpoint

以下程式碼範例顯示如何使用 get-resolver-endpoint。

AWS CLI

取得解析程式端點的相關資訊

下列get-resolver-endpoint範例顯示傳出指定端點的詳細資訊。您可以透過指定適用的端點 ID，將 get-resolver-endpoint用於傳入和傳出端點。

```
aws route53resolver get-resolver-endpoint \
```

```
    --resolver-endpoint-id rslvr-out-d5e5920e37example
```

輸出：

```
{
    "ResolverEndpoint": {
        "Id": "rslvr-out-d5e5920e37example",
        "CreatorRequestId": "2020-01-01-18:47",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-out-d5e5920e37example",
        "Name": "my-outbound-endpoint",
        "SecurityGroupIds": [
            "sg-05cd7b25d6example"
        ],
        "Direction": "OUTBOUND",
        "IpAddressCount": 2,
        "HostVPCId": "vpc-304bexam",
        "Status": "OPERATIONAL",
        "StatusMessage": "This Resolver Endpoint is operational.",
        "CreationTime": "2020-01-01T23:50:50.979Z",
        "ModificationTime": "2020-01-02T23:50:50.979Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的您在建立或編輯傳入端點時指定的值。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResolverEndpoint。

get-resolver-rule-association

以下程式碼範例顯示如何使用 get-resolver-rule-association。

AWS CLI

取得解析程式規則與 VPC 之間關聯的相關資訊

下列get-resolver-rule-association範例顯示指定解析程式規則與 VPC 之間關聯的詳細資訊。您可以使用 associate-resolver-rule 建立解析程式規則和 VPC 的關聯。

```
aws route53resolver get-resolver-rule-association \
```

```
    --resolver-rule-association-id rslvr-rrassoc-d61cbb2c8bexample
```

輸出：

```
{
    "ResolverRuleAssociation": {
        "Id": "rslvr-rrassoc-d61cbb2c8bexample",
        "ResolverRuleId": "rslvr-rr-42b60677c0example",
        "Name": "my-resolver-rule-association",
        "VPCId": "vpc-304bexam",
        "Status": "COMPLETE",
        "StatusMessage": ""
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResolverRuleAssociation。

get-resolver-rule

以下程式碼範例顯示如何使用 get-resolver-rule。

AWS CLI

取得解析程式規則的相關資訊

下列get-resolver-rule範例顯示指定 Resolver 規則的詳細資訊，例如規則轉送 DNS 查詢的網域名稱，以及規則相關聯的傳出解析程式端點 ID。

```
aws route53resolver get-resolver-rule \
    --resolver-rule-id rslvr-rr-42b60677c0example
```

輸出：

```
{
    "ResolverRule": {
        "Id": "rslvr-rr-42b60677c0example",
        "CreatorRequestId": "2020-01-01-18:47",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/rslvr-
rr-42b60677c0example",
        "DomainName": "example.com.",
        "Status": "COMPLETE",
```

```
        "StatusMessage": "[Trace id: 1-5dc4b177-ff1d9d001a0f80005example]
 Successfully created Resolver Rule.",
        "RuleType": "FORWARD",
        "Name": "my-rule",
        "TargetIps": [
            {
                "Ip": "192.0.2.45",
                "Port": 53
            }
        ],
        "ResolverEndpointId": "rslvr-out-d5e5920e37example",
        "OwnerId": "111122223333",
        "ShareStatus": "NOT_SHARED"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的您在建立或編輯規則時指定的值。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResolverRule。

import-firewall-domains

以下程式碼範例顯示如何使用 import-firewall-domains。

AWS CLI

將網域匯入網域清單

下列import-firewall-domains範例會將一組網域從檔案匯入您指定的 DNS 防火牆網域清單。

```
aws route53resolver import-firewall-domains \
    --firewall-domain-list-id rslvr-fdl-d61cbb2cbexample \
    --operation REPLACE \
    --domain-file-url s3://PATH/TO/YOUR/FILE
```

輸出：

```
{
    "Id": "rslvr-fdl-d61cbb2cbexample",
    "Name": "test",
    "Status": "IMPORTING",
    "StatusMessage": "Importing domains from provided file."
```

```
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理您自己的網域清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ImportFirewallDomains。

list-firewall-configs

以下程式碼範例顯示如何使用 `list-firewall-configs`。

AWS CLI

列出防火牆組態

下列`list-firewall-configs`範例列出您的 DNS 防火牆組態。

```
aws route53resolver list-firewall-configs
```

輸出：

```
{
    "FirewallConfigs": [
        {
            "Id": "rslvr-fc-86016850cexample",
            "ResourceId": "vpc-31e92222",
            "OwnerId": "123456789012",
            "FirewallFailOpen": "DISABLED"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的 DNS 防火牆 VPC 組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFirewallConfigs。

list-firewall-domain-lists

以下程式碼範例顯示如何使用 `list-firewall-domain-lists`。

AWS CLI

列出所有 Route 53 Resolver DNS Firewall 網域清單

下列list-firewall-domain-lists範例列出所有網域清單。

```
aws route53resolver list-firewall-domain-lists
```

輸出：

```
{
    "FirewallDomainLists": [
        {
            "Id": "rslvr-fdl-2c46f2ecfexample",
            "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-domain-
list/rslvr-fdl-2c46f2ecfexample",
            "Name": "AWSManagedDomainsMalwareDomainList",
            "CreatorRequestId": "AWSManagedDomainsMalwareDomainList",
            "ManagedOwnerName": "Route 53 Resolver DNS Firewall"
        },
        {
            "Id": "rslvr-fdl-aa970e9e1example",
            "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-domain-
list/rslvr-fdl-aa970e9e1example",
            "Name": "AWSManagedDomainsBotnetCommandandControl",
            "CreatorRequestId": "AWSManagedDomainsBotnetCommandandControl",
            "ManagedOwnerName": "Route 53 Resolver DNS Firewall"
        },
        {
            "Id": "rslvr-fdl-42b60677cexample",
            "Arn": "arn:aws:route53resolver:us-west-2:123456789111:firewall-domain-
list/rslvr-fdl-42b60677cexample",
            "Name": "test",
            "CreatorRequestId": "my-request-id"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的 Route 53 Resolver DNS 防火牆網域清單。 Amazon Route 53

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFirewallDomainLists。

list-firewall-domains

以下程式碼範例顯示如何使用 list-firewall-domains。

AWS CLI

在網域清單中列出網域

下列`list-firewall-domains`範例列出您指定的 DNS 防火牆網域清單中的網域。

```
aws route53resolver list-firewall-domains \
    --firewall-domain-list-id rslvr-fdl-d61cbb2cbexample
```

輸出：

```
{
    "Domains": [
        "test1.com.",
        "test2.com.",
        "test3.com."
    ]
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理您自己的網域清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFirewallDomains。

`list-firewall-rule-group-associations`

以下程式碼範例顯示如何使用 `list-firewall-rule-group-associations`。

AWS CLI

列出 DNS 防火牆規則群組關聯

下列`list-firewall-rule-group-associations`範例列出您與 Amazon VPCs DNS 防火牆規則群組關聯。

```
aws route53resolver list-firewall-rule-group-associations
```

輸出：

```
{
    "FirewallRuleGroupAssociations": [
        {
```

```
            "Id": "rslvr-frgassoc-57e8873d7example",
            "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-
group-association/rslvr-frgassoc-57e8873d7example",
            "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
            "VpcId": "vpc-31e92222",
            "Name": "test-association",
            "Priority": 101,
            "MutationProtection": "DISABLED",
            "Status": "UPDATING",
            "StatusMessage": "Creating Firewall Rule Group Association",
            "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
            "CreationTime": "2021-05-25T21:47:48.755768Z",
            "ModificationTime": "2021-05-25T21:47:48.755768Z"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理 VPC 與 Route 53 Resolver DNS 防火牆規則群組之間的關聯。 Amazon Route 53

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFirewallRuleGroupAssociations。

`list-firewall-rule-groups`

以下程式碼範例顯示如何使用 `list-firewall-rule-groups`。

AWS CLI

取得防火牆規則群組的清單

下列`list-firewall-rule-groups`範例列出您的 DNS 防火牆規則群組。

```
aws route53resolver list-firewall-rule-groups
```

輸出：

```
{
    "FirewallRuleGroups": [
        {
            "Id": "rslvr-frg-47f93271fexample",
            "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-
group/rslvr-frg-47f93271fexample",
```

```
            "Name": "test",
            "OwnerId": "123456789012",
            "CreatorRequestId": "my-request-id",
            "ShareStatus": "NOT_SHARED"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理 DNS 防火牆中的規則群組和規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFirewallRuleGroups。

`list-firewall-rules`

以下程式碼範例顯示如何使用 `list-firewall-rules`。

AWS CLI

列出防火牆規則

下列`list-firewall-rules`範例列出防火牆規則群組中的所有 DNS 防火牆規則。

```
aws route53resolver list-firewall-rules \
    --firewall-rule-group-id rslvr-frg-47f93271fexample
```

輸出：

```
{
    "FirewallRules": [
        {
            "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
            "FirewallDomainListId": "rslvr-fdl-9e956e9ffexample",
            "Name": "allow-rule",
            "Priority": 101,
            "Action": "ALLOW",
            "CreatorRequestId": "d81e3fb7-020b-415e-939f-EXAMPLE11111",
            "CreationTime": "2021-05-25T21:44:00.346093Z",
            "ModificationTime": "2021-05-25T21:44:00.346093Z"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理 DNS 防火牆中的規則群組和規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFirewallRules。

list-resolver-endpoint-ip-addresses

以下程式碼範例顯示如何使用 list-resolver-endpoint-ip-addresses。

AWS CLI

列出指定傳入或傳出端點的 IP 地址

下列list-resolver-endpoint-ip-addresses範例列出與傳入端點 相關聯的 IP 地址資訊rslvr-in-f9ab8a03f1example。您也可以指定適用的端點 ID，將 list-resolver-endpoint-ip-addresses用於傳出端點。

```
aws route53resolver list-resolver-endpoint-ip-addresses \
    --resolver-endpoint-id rslvr-in-f9ab8a03f1example
```

輸出：

```
{
    "MaxResults": 10,
    "IpAddresses": [
        {
            "IpId": "rni-1de60cdbfeexample",
            "SubnetId": "subnet-ba47exam",
            "Ip": "192.0.2.44",
            "Status": "ATTACHED",
            "StatusMessage": "This IP address is operational.",
            "CreationTime": "2020-01-03T23:02:29.587Z",
            "ModificationTime": "2020-01-03T23:03:05.555Z"
        },
        {
            "IpId": "rni-aac7085e38example",
            "SubnetId": "subnet-12d8exam",
            "Ip": "192.0.2.45",
            "Status": "ATTACHED",
            "StatusMessage": "This IP address is operational.",
            "CreationTime": "2020-01-03T23:02:29.593Z",
            "ModificationTime": "2020-01-03T23:02:55.060Z"
        }
```

```
    ]
}
```

如需輸出中值的詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的建立或編輯傳入端點時指定的值，以及建立或編輯傳出端點時指定的值。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResolverEndpointIpAddresses。

`list-resolver-endpoints`

以下程式碼範例顯示如何使用 `list-resolver-endpoints`。

AWS CLI

列出 AWS 區域中的解析程式端點

下列`list-resolver-endpoints`範例列出目前帳戶中存在的傳入和傳出解析程式端點。

```
aws route53resolver list-resolver-endpoints
```

輸出：

```
{
    "MaxResults": 10,
    "ResolverEndpoints": [
        {
            "Id": "rslvr-in-497098ad59example",
            "CreatorRequestId": "2020-01-01-18:47",
            "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-
endpoint/rslvr-in-497098ad59example",
            "Name": "my-inbound-endpoint",
            "SecurityGroupIds": [
                "sg-05cd7b25d6example"
            ],
            "Direction": "INBOUND",
            "IpAddressCount": 2,
            "HostVPCId": "vpc-304bexam",
            "Status": "OPERATIONAL",
            "StatusMessage": "This Resolver Endpoint is operational.",
            "CreationTime": "2020-01-01T23:25:45.538Z",
            "ModificationTime": "2020-01-01T23:25:45.538Z"
        },
        {
```

```
            "Id": "rslvr-out-d5e5920e37example",
            "CreatorRequestId": "2020-01-01-18:48",
            "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-
endpoint/rslvr-out-d5e5920e37example",
            "Name": "my-outbound-endpoint",
            "SecurityGroupIds": [
                "sg-05cd7b25d6example"
            ],
            "Direction": "OUTBOUND",
            "IpAddressCount": 2,
            "HostVPCId": "vpc-304bexam",
            "Status": "OPERATIONAL",
            "StatusMessage": "This Resolver Endpoint is operational.",
            "CreationTime": "2020-01-01T23:50:50.979Z",
            "ModificationTime": "2020-01-01T23:50:50.979Z"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResolverEndpoints。

list-resolver-rule-associations

以下程式碼範例顯示如何使用 list-resolver-rule-associations。

AWS CLI

列出解析程式規則與 VPCs之間的關聯

下列list-resolver-rule-associations範例列出解析程式規則與目前 AWS 帳戶中 VPCs之間的關聯。

```
aws route53resolver list-resolver-rule-associations
```

輸出：

```
{
    "MaxResults": 30,
    "ResolverRuleAssociations": [
        {
            "Id": "rslvr-autodefined-assoc-vpc-304bexam-internet-resolver",
            "ResolverRuleId": "rslvr-autodefined-rr-internet-resolver",
```

```
            "Name": "System Rule Association",
            "VPCId": "vpc-304bexam",
            "Status": "COMPLETE",
            "StatusMessage": ""
        },
        {
            "Id": "rslvr-rrassoc-d61cbb2c8bexample",
            "ResolverRuleId": "rslvr-rr-42b60677c0example",
            "Name": "my-resolver-rule-association",
            "VPCId": "vpc-304bexam",
            "Status": "COMPLETE",
            "StatusMessage": ""
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的 Route 53 Resolver 如何將 DNS 查詢從您的 VPCs 轉送到您的網路。 Amazon Route 53

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResolverRuleAssociations。

list-resolver-rules

以下程式碼範例顯示如何使用 `list-resolver-rules`。

AWS CLI

列出解析程式規則

下列`list-resolver-rules`範例列出目前 AWS 帳戶中所有解析程式規則。

```
aws route53resolver list-resolver-rules
```

輸出：

```
{
    "MaxResults": 30,
    "ResolverRules": [
        {
            "Id": "rslvr-autodefined-rr-internet-resolver",
            "CreatorRequestId": "",
            "Arn": "arn:aws:route53resolver:us-west-2::autodefined-rule/rslvr-
autodefined-rr-internet-resolver",
```

```
            "DomainName": ".",
            "Status": "COMPLETE",
            "RuleType": "RECURSIVE",
            "Name": "Internet Resolver",
            "OwnerId": "Route 53 Resolver",
            "ShareStatus": "NOT_SHARED"
        },
        {
            "Id": "rslvr-rr-42b60677c0example",
            "CreatorRequestId": "2020-01-01-18:47",
            "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/
rslvr-rr-42b60677c0bc4e299",
            "DomainName": "example.com.",
            "Status": "COMPLETE",
            "StatusMessage": "[Trace id: 1-5dc4b177-ff1d9d001a0f80005example]
 Successfully created Resolver Rule.",
            "RuleType": "FORWARD",
            "Name": "my-rule",
            "TargetIps": [
                {
                    "Ip": "192.0.2.45",
                    "Port": 53
                }
            ],
            "ResolverEndpointId": "rslvr-out-d5e5920e37example",
            "OwnerId": "111122223333",
            "ShareStatus": "NOT_SHARED"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的 Route 53 Resolver 如何將 DNS 查詢從您的 VPCs 轉送到您的網路。 Amazon Route 53

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResolverRules。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出解析程式資源的標籤

下列list-tags-for-resource範例列出指派給指定解析程式規則的標籤。

```
aws route53resolver list-tags-for-resource \
    --resource-arn "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/
rslvr-rr-42b60677c0example"
```

輸出：

```
{
    "Tags": [
        {
            "Key": "my-key-1",
            "Value": "my-value-1"
        },
        {
            "Key": "my-key-2",
            "Value": "my-value-2"
        }
    ]
}
```

如需使用標籤進行成本分配的詳細資訊，請參閱AWS 《帳單和成本管理使用者指南》中的使用成本分配標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

put-firewall-rule-group-policy

以下程式碼範例顯示如何使用 put-firewall-rule-group-policy。

AWS CLI

連接 AWS IAM 政策以共用防火牆規則群組政策

下列put-firewall-rule-group-policy範例會連接 AWS Identity and Access Management (AWS IAM) 政策來共用規則群組。

```
aws route53resolver put-firewall-rule-group-policy \
    --firewall-rule-group-policy "{\"Version\":\"2012-10-17\",
\"Statement\":[{\"Sid\":\"test\",\"Effect\":\"Allow\",\"Principal
\":{\"AWS\":\"arn:aws:iam::AWS_ACCOUNT_ID:root\"},\"Action\":
```

```
[\"route53resolver:GetFirewallRuleGroup\",\"route53resolver:ListFirewallRuleGroups
\"],\"Resource\":\"arn:aws:route53resolver:us-east-1:AWS_ACCOUNT_ID:firewall-rule-
group/rslvr-frg-47f93271fexample\"}]}"
```

輸出：

```
{
    "ReturnValue": true
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理 DNS 防火牆中的規則群組和規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutFirewallRuleGroupPolicy。

put-resolver-rule-policy

以下程式碼範例顯示如何使用 put-resolver-rule-policy。

AWS CLI

與其他 AWS 帳戶共用解析程式規則

下列put-resolver-rule-policy範例會指定您要與其他 AWS 帳戶共用的 Resolver 規則、您要共用規則的帳戶，以及您希望帳戶能夠在規則上執行的規則相關操作。

注意 您必須使用建立規則之相同帳戶的登入資料來執行此命令。

```
aws route53resolver put-resolver-rule-policy \
    --region us-east-1 \
    --arn "arn:aws:route53resolver:us-east-1:111122223333:resolver-rule/rslvr-
rr-42b60677c0example" \
    --resolver-rule-policy "{\"Version\": \"2012-10-17\", \
        \"Statement\": [ { \
        \"Effect\" : \"Allow\", \
        \"Principal\" : {\"AWS\" : \"444455556666\" }, \
        \"Action\" : [ \
            \"route53resolver:GetResolverRule\", \
            \"route53resolver:AssociateResolverRule\", \
            \"route53resolver:DisassociateResolverRule\", \
            \"route53resolver:ListResolverRules\", \
            \"route53resolver:ListResolverRuleAssociations\" ], \
```

```
        \"Resource\" : [ \"arn:aws:route53resolver:us-east-1:111122223333:resolver-
rule/rslvr-rr-42b60677c0example\" ] } ] }"
```

輸出：

```
{
    "ReturnValue": true
}
```

執行 之後put-resolver-rule-policy，您可以執行下列兩個 Resource Access Manager (RAM) 命令。您必須使用要與之共用規則的帳戶：

get-resource-share-invitations 傳回值 resourceShareInvitationArn。您需要此值才能接受使用共用規則的邀請。accept-resource-share-invitation 接受使用共用規則的邀請。

如需詳細資訊，請參閱下列 文件：

《Amazon Route 53 開發人員指南》中的 get-resource-share-invitationsaccept-resource-share-invitations與其他 AWS 帳戶共享轉送規則和使用共用規則

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutResolverRulePolicy。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤與解析程式資源建立關聯

下列tag-resource範例會將兩個標籤鍵/值對與指定的解析程式規則建立關聯。

```
aws route53resolver tag-resource \
    --resource-arn "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/
rslvr-rr-42b60677c0example" \
    --tags "Key=my-key-1,Value=my-value-1" "Key=my-key-2,Value=my-value-2"
```

此命令不會產生輸出。

如需使用標籤進行成本分配的詳細資訊，請參閱AWS 《帳單和成本管理使用者指南》中的使用成本分配標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從解析程式資源移除標籤

下列untag-resource範例會從指定的解析程式規則中移除兩個標籤。

```
aws route53resolver untag-resource \
    --resource-arn "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/
rslvr-rr-42b60677c0example" \
    --tag-keys my-key-1 my-key-2
```

此命令不會產生輸出。若要確認標籤已移除，您可以使用 list-tags-for-resource。

如需使用標籤進行成本分配的詳細資訊，請參閱AWS 《帳單和成本管理使用者指南》中的使用成本分配標籤。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-firewall-config

以下程式碼範例顯示如何使用 update-firewall-config。

AWS CLI

更新防火牆組態

下列update-firewall-config範例會更新 DNS 防火牆組態。

```
aws route53resolver update-firewall-config \
    --resource-id vpc-31e92222 \
    --firewall-fail-open DISABLED
```

輸出：

```
{
    "FirewallConfig": {
```

```
        "Id": "rslvr-fc-86016850cexample",
        "ResourceId": "vpc-31e92222",
        "OwnerId": "123456789012",
        "FirewallFailOpen": "DISABLED"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的 DNS 防火牆 VPC 組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateFirewallConfig。

update-firewall-domains

以下程式碼範例顯示如何使用 update-firewall-domains。

AWS CLI

更新網域清單

下列update-firewall-domains範例會使用您提供的 ID，將網域新增至網域清單。

```
aws route53resolver update-firewall-domains \
    --firewall-domain-list-id rslvr-fdl-42b60677cexampleb \
    --operation ADD \
    --domains test1.com test2.com test3.com
```

輸出：

```
{
    "Id": "rslvr-fdl-42b60677cexample",
    "Name": "test",
    "Status": "UPDATING",
    "StatusMessage": "Updating the Firewall Domain List"
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理您自己的網域清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateFirewallDomains。

update-firewall-rule-group-association

以下程式碼範例顯示如何使用 update-firewall-rule-group-association。

AWS CLI

更新防火牆規則群組關聯

下列update-firewall-rule-group-association範例會更新防火牆規則群組關聯。

```
aws route53resolver update-firewall-rule-group-association \
    --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example \
    --priority 103
```

輸出：

```
{
    "FirewallRuleGroupAssociation": {
        "Id": "rslvr-frgassoc-57e8873d7example",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-
association/rslvr-frgassoc-57e8873d7example",
        "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
        "VpcId": "vpc-31e92222",
        "Name": "test-association",
        "Priority": 103,
        "MutationProtection": "DISABLED",
        "Status": "UPDATING",
        "StatusMessage": "Updating the Firewall Rule Group Association Attributes",
        "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
        "CreationTime": "2021-05-25T21:47:48.755768Z",
        "ModificationTime": "2021-05-25T21:50:09.272569Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理 VPC 與 Route 53 Resolver DNS 防火牆規則群組之間的關聯。 Amazon Route 53

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateFirewallRuleGroupAssociation。

update-firewall-rule

以下程式碼範例顯示如何使用 update-firewall-rule。

AWS CLI

更新防火牆規則

下列update-firewall-rule範例會使用您指定的參數更新防火牆規則。

```
aws route53resolver update-firewall-rule \
    --firewall-rule-group-id rslvr-frg-47f93271fexample \
    --firewall-domain-list-id rslvr-fdl-9e956e9ffexample \
    --priority 102
```

輸出：

```
{
    "FirewallRule": {
        "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
        "FirewallDomainListId": "rslvr-fdl-9e956e9ffexample",
        "Name": "allow-rule",
        "Priority": 102,
        "Action": "ALLOW",
        "CreatorRequestId": "d81e3fb7-020b-415e-939f-EXAMPLE11111",
        "CreationTime": "2021-05-25T21:44:00.346093Z",
        "ModificationTime": "2021-05-25T21:45:59.611600Z"
    }
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的管理 DNS 防火牆中的規則群組和規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateFirewallRule。

update-resolver-endpoint

以下程式碼範例顯示如何使用 update-resolver-endpoint。

AWS CLI

更新解析程式端點的名稱

下列update-resolver-endpoint範例會更新解析程式端點的名稱。不支援更新其他值。

```
aws route53resolver update-resolver-endpoint \
    --resolver-endpoint-id rslvr-in-b5d45e32bdc445f09 \
    --name my-renamed-inbound-endpoint
```

輸出：

```
{
    "ResolverEndpoint": {
        "Id": "rslvr-in-b5d45e32bdexample",
        "CreatorRequestId": "2020-01-02-18:48",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-in-b5d45e32bdexample",
        "Name": "my-renamed-inbound-endpoint",
        "SecurityGroupIds": [
            "sg-f62bexam"
        ],
        "Direction": "INBOUND",
        "IpAddressCount": 2,
        "HostVPCId": "vpc-304bexam",
        "Status": "OPERATIONAL",
        "StatusMessage": "This Resolver Endpoint is operational.",
        "CreationTime": "2020-01-01T18:33:59.265Z",
        "ModificationTime": "2020-01-08T18:33:59.265Z"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateResolverEndpoint。

update-resolver-rule

以下程式碼範例顯示如何使用 update-resolver-rule。

AWS CLI

範例 1：更新設定解析程式端點

下列update-resolver-rule範例會更新規則的名稱、DNS 查詢轉送至的內部部署網路上的 IP 地址，以及您用來將查詢轉送至網路的傳出解析程式端點 ID。

請注意， 的現有值TargetIps會遭到覆寫，因此您必須指定更新後您希望規則擁有的所有 IP 地址。

```
aws route53resolver update-resolver-rule \
    --resolver-rule-id rslvr-rr-1247fa64f3example \
    --config Name="my-2nd-rule",TargetIps=[{Ip=192.0.2.45,Port=53},
{Ip=192.0.2.46,Port=53}],ResolverEndpointId=rslvr-out-7b89ed0d25example
```

輸出：

```
{
    "ResolverRule": {
        "Id": "rslvr-rr-1247fa64f3example",
        "CreatorRequestId": "2020-01-02-18:47",
        "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/rslvr-
rr-1247fa64f3example",
        "DomainName": "www.example.com.",
        "Status": "COMPLETE",
        "StatusMessage": "[Trace id: 1-5dcc90b9-8a8ee860aba1ebd89example]
 Successfully updated Resolver Rule.",
        "RuleType": "FORWARD",
        "Name": "my-2nd-rule",
        "TargetIps": [
            {
                "Ip": "192.0.2.45",
                "Port": 53
            },
            {
                "Ip": "192.0.2.46",
                "Port": 53
            }
        ],
        "ResolverEndpointId": "rslvr-out-7b89ed0d25example",
        "OwnerId": "111122223333",
        "ShareStatus": "NOT_SHARED"
    }
}
```

範例 2：使用 ``config`` 設定的檔案更新 Resolver 端點設定

或者，您可以在 JSON 檔案中包含config設定，然後在呼叫 時指定該檔案update-resolver-rule。

```
aws route53resolver update-resolver-rule \
    --resolver-rule-id rslvr-rr-1247fa64f3example \
    --config file://c:\temp\update-resolver-rule.json
```

update-resolver-rule.json 的內容。

```
{
    "Name": "my-2nd-rule",
    "TargetIps": [
```

```
        {
            "Ip": "192.0.2.45",
            "Port": 53
        },
        {
            "Ip": "192.0.2.46",
            "Port": 53
        }
    ],
    "ResolverEndpointId": "rslvr-out-7b89ed0d25example"
}
```

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的您在建立或編輯規則時指定的值。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateResolverRule。

使用 的 Amazon S3 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon S3 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`abort-multipart-upload`

以下程式碼範例顯示如何使用 `abort-multipart-upload`。

AWS CLI

中止指定的分段上傳

下列`abort-multipart-upload`命令會中止儲存貯體 `multipart/01` 中金鑰的分段上傳`amzn-s3-demo-bucket`。

```
aws s3api abort-multipart-upload \
    --bucket amzn-s3-demo-bucket \
    --key multipart/01 \
    --upload-
id dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

此命令所需的上傳 ID 由 輸出，create-multipart-upload也可以使用 擷取list-multipart-uploads。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AbortMultipartUpload。

complete-multipart-upload

以下程式碼範例顯示如何使用 complete-multipart-upload。

AWS CLI

下列命令會完成儲存貯體 multipart/01 中金鑰 的分段上傳amzn-s3-demo-bucket：

```
aws s3api complete-multipart-upload --multipart-upload file://
mpustruct --bucket amzn-s3-demo-bucket --key 'multipart/01' --upload-
id dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

此命令所需的上傳 ID 由 輸出，create-multipart-upload也可以使用 擷取list-multipart-uploads。

上述命令中的分段上傳選項採用 JSON 結構，描述應該重新組合為完整檔案的分段上傳部分。在此範例中，file:// 字首用於從本機資料夾中名為 mpustruct 的檔案載入 JSON 結構。

mpustruct：

```
{
  "Parts": [
    {
      "ETag": "e868e0f4719e394144ef36531ee6824c",
      "PartNumber": 1
    },
    {
      "ETag": "6bb2b12753d66fe86da4998aa33fffb0",
      "PartNumber": 2
    },
```

```
    {
      "ETag": "d0a0112e841abec9c9ec83406f0159c8",
      "PartNumber": 3
    }
  ]
}
```

每次使用 upload-part命令上傳組件時，都會輸出每個組件的 ETag 值，也可以透過呼叫list-parts或計算每個組件的 MD5 檢查總和來擷取。

輸出：

```
{
    "ETag": "\"3944a9f7a4faab7f78788ff6210f63f0-3\"",
    "Bucket": "amzn-s3-demo-bucket",
    "Location": "https://amzn-s3-demo-bucket.s3.amazonaws.com/multipart%2F01",
    "Key": "multipart/01"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CompleteMultipartUpload。

copy-object

以下程式碼範例顯示如何使用 copy-object。

AWS CLI

下列命令會將物件從 複製到 bucket-1 bucket-2：

```
aws s3api copy-object --copy-source bucket-1/test.txt --key test.txt --
bucket bucket-2
```

輸出：

```
{
    "CopyObjectResult": {
        "LastModified": "2015-11-10T01:07:25.000Z",
        "ETag": "\"589c8b79c230a6ecd5a7e1d040a9a030\""
    },
    "VersionId": "YdnYvTCVDqRRFA.NFJjy36p0hxifMlkA"
```

```
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CopyObject。

cp

以下程式碼範例顯示如何使用 cp。

AWS CLI

範例 1：將本機檔案複製到 S3

下列cp命令會將單一檔案複製到指定的儲存貯體和金鑰：

```
aws s3 cp test.txt s3://amzn-s3-demo-bucket/test2.txt
```

輸出：

```
upload: test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

範例 2：將本機檔案複製到具有過期日期的 S3

下列cp命令會將單一檔案複製到指定的儲存貯體和金鑰，並在指定的 ISO 8601 時間戳記過期：

```
aws s3 cp test.txt s3://amzn-s3-demo-bucket/test2.txt \
    --expires 2014-10-01T20:30:00Z
```

輸出：

```
upload: test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

範例 3：將檔案從 S3 複製到 S3

下列cp命令會將單一 s3 物件複製到指定的儲存貯體和金鑰：

```
aws s3 cp s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket/test2.txt
```

輸出：

```
copy: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

範例 4：將 S3 物件複製到本機檔案

下列cp命令會將單一物件複製到本機指定的檔案：

```
aws s3 cp s3://amzn-s3-demo-bucket/test.txt test2.txt
```

輸出：

```
download: s3://amzn-s3-demo-bucket/test.txt to test2.txt
```

範例 5：將 S3 物件從一個儲存貯體複製到另一個儲存貯體

下列cp命令會將單一物件複製到指定的儲存貯體，同時保留其原始名稱：

```
aws s3 cp s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket2/
```

輸出：

```
copy: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket2/test.txt
```

範例 6：遞迴複製 S3 物件至本機目錄

使用 參數 傳遞時`--recursive`，下列cp命令會遞迴地將所有指定字首和儲存貯體下的物件複製到指定的目錄。在此範例中，儲存貯體`amzn-s3-demo-bucket`具有物件 `test1.txt`和 `test2.txt`：

```
aws s3 cp s3://amzn-s3-demo-bucket . \
    --recursive
```

輸出：

```
download: s3://amzn-s3-demo-bucket/test1.txt to test1.txt
download: s3://amzn-s3-demo-bucket/test2.txt to test2.txt
```

範例 7：遞迴複製本機檔案至 S3

使用 參數 傳遞時--recursive，下列cp命令會遞迴地將指定目錄下的所有檔案複製到指定的儲存貯體和字首，同時使用 --exclude 參數排除某些檔案。在此範例中， 目錄myDir具有 檔案test1.txt和 test2.jpg：

```
aws s3 cp myDir s3://amzn-s3-demo-bucket/ \
    --recursive \
    --exclude "*.jpg"
```

輸出：

```
upload: myDir/test1.txt to s3://amzn-s3-demo-bucket/test1.txt
```

範例 8：遞迴複製 S3 物件到另一個儲存貯體

使用 參數 傳遞時--recursive，下列cp命令會遞迴地將指定儲存貯體下的所有物件複製到另一個儲存貯體，同時使用 --exclude 參數排除某些物件。在此範例中，儲存貯體amzn-s3-demo-bucket具有 物件 test1.txt和 another/test1.txt：

```
aws s3 cp s3://amzn-s3-demo-bucket/ s3://amzn-s3-demo-bucket2/ \
    --recursive \
    --exclude "another/*"
```

輸出：

```
copy: s3://amzn-s3-demo-bucket/test1.txt to s3://amzn-s3-demo-bucket2/test1.txt
```

您可以結合 --exclude和 --include選項，只複製符合模式的物件，不包括所有其他物件：

```
aws s3 cp s3://amzn-s3-demo-bucket/logs/ s3://amzn-s3-demo-bucket2/logs/ \
    --recursive \
    --exclude "*" \
    --include "*.log"
```

輸出：

```
copy: s3://amzn-s3-demo-bucket/logs/test/test.log to s3://amzn-s3-demo-bucket2/logs/
test/test.log
copy: s3://amzn-s3-demo-bucket/logs/test3.log to s3://amzn-s3-demo-bucket2/logs/
test3.log
```

範例 9：複製 S3 物件時設定存取控制清單 (ACL)

將 ACL 設定為 時，下列cp命令會將單一物件複製到指定的儲存貯體和金鑰public-read-write：

```
aws s3 cp s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket/test2.txt \
    --acl public-read-write
```

輸出：

```
copy: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

請注意，如果您使用 --acl選項，請確保任何相關聯的 IAM 政策都包含 "s3:PutObjectAcl"動作：

```
aws iam get-user-policy \
    --user-name myuser \
    --policy-name mypolicy
```

輸出：

```
{
    "UserName": "myuser",
    "PolicyName": "mypolicy",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": [
                    "s3:PutObject",
                    "s3:PutObjectAcl"
                ],
                "Resource": [
                    "arn:aws:s3:::amzn-s3-demo-bucket/*"
                ],
                "Effect": "Allow",
                "Sid": "Stmt1234567891234"
            }
        ]
    }
}
```

範例 10：授予 S3 物件的許可

下列cp命令說明使用 --grants選項，將讀取存取權授予 URI 識別的所有使用者，以及將完全控制權授予正式 ID 識別的特定使用者：

```
aws s3 cp file.txt s3://amzn-s3-demo-bucket/ --grants read=uri=http://
acs.amazonaws.com/groups/global/
AllUsers full=id=79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

輸出：

```
upload: file.txt to s3://amzn-s3-demo-bucket/file.txt
```

範例 11：將本機檔案串流上傳至 S3

PowerShell 可能會變更 的編碼，或將 CRLF 新增至管道輸入。

下列cp命令會將本機檔案串流從標準輸入上傳至指定的儲存貯體和金鑰：

```
aws s3 cp - s3://amzn-s3-demo-bucket/stream.txt
```

範例 12：將大於 50GB 的本機檔案串流上傳至 S3

下列cp命令會將 51GB 本機檔案串流從標準輸入上傳至指定的儲存貯體和金鑰。必須提供 --expected-size選項，否則當達到預設部分限制 10，000 時，上傳可能會失敗：

```
aws s3 cp - s3://amzn-s3-demo-bucket/stream.txt --expected-size 54760833024
```

範例 13：下載 S3 物件做為本機檔案串流

PowerShell 可能會變更 的編碼，或將 CRLF 新增至管道或重新導向的輸出。

下列cp命令會在本機將 S3 物件下載為標準輸出的串流。下載為串流目前與 --recursive 參數不相容：

```
aws s3 cp s3://amzn-s3-demo-bucket/stream.txt -
```

範例 14：上傳至 S3 存取點

下列cp命令會將單一檔案 (mydoc.txt) 上傳至金鑰 () 的存取點 (myaccesspoint)mykey：

```
aws s3 cp mydoc.txt s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/mykey
```

輸出：

```
upload: mydoc.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/mykey
```

範例 15：從 S3 存取點下載

下列cp命令會將單一物件 (mykey) 從存取點 (myaccesspoint) 下載至本機檔案 ()mydoc.txt：

```
aws s3 cp s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
mykey mydoc.txt
```

輸出：

```
download: s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey to
 mydoc.txt
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 Cp。

create-bucket

以下程式碼範例顯示如何使用 create-bucket。

AWS CLI

範例 1：建立儲存貯體

下列create-bucket範例會建立名為 的儲存貯體amzn-s3-demo-bucket：

```
aws s3api create-bucket \
    --bucket amzn-s3-demo-bucket \
    --region us-east-1
```

輸出：

```
{
```

```
    "Location": "/amzn-s3-demo-bucket"
}
```

如需詳細資訊，請參閱《Amazon S3 使用者指南》中的建立儲存貯體。

範例 2：在擁有者強制執行的情況下建立儲存貯體

下列create-bucket範例會建立名為 的儲存貯體amzn-s3-demo-bucket，該儲存貯體使用 S3 物件擁有權的儲存貯體擁有者強制執行設定。

```
aws s3api create-bucket \
    --bucket amzn-s3-demo-bucket \
    --region us-east-1 \
    --object-ownership BucketOwnerEnforced
```

輸出：

```
{
    "Location": "/amzn-s3-demo-bucket"
}
```

如需詳細資訊，請參閱《Amazon S3 使用者指南》中的控制物件的所有權並停用 ACL。

範例 3：在 ``us-east-1`` 區域外建立儲存貯體

下列create-bucket範例會在 eu-west-1 區域中建立名為 的儲存貯amzn-s3-demo-bucket體。以外的區域us-east-1需要LocationConstraint指定適當的 ，才能在所需區域中建立儲存貯體。

```
aws s3api create-bucket \
    --bucket amzn-s3-demo-bucket \
    --region eu-west-1 \
    --create-bucket-configuration LocationConstraint=eu-west-1
```

輸出：

```
{
    "Location": "http://amzn-s3-demo-bucket.s3.amazonaws.com/"
}
```

如需詳細資訊，請參閱《Amazon S3 使用者指南》中的建立儲存貯體。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateBucket。

create-multipart-upload

以下程式碼範例顯示如何使用 create-multipart-upload。

AWS CLI

下列命令會使用amzn-s3-demo-bucket金鑰 在 儲存貯體中建立分段上傳multipart/01：

```
aws s3api create-multipart-upload --bucket amzn-s3-demo-bucket --key 'multipart/01'
```

輸出：

```
{
    "Bucket": "amzn-s3-demo-bucket",
    "UploadId":
 "dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3URC
    "Key": "multipart/01"
}
```

完成的檔案會在儲存貯01體 中名為 multipart的資料夾中命名amzn-s3-demo-bucket。儲存要與 upload-part命令搭配使用的上傳 ID、金鑰和儲存貯體名稱。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateMultipartUpload。

delete-bucket-analytics-configuration

以下程式碼範例顯示如何使用 delete-bucket-analytics-configuration。

AWS CLI

刪除儲存貯體的分析組態

下列delete-bucket-analytics-configuration範例會移除指定儲存貯體和 ID 的分析組態。

```
aws s3api delete-bucket-analytics-configuration \
    --bucket amzn-s3-demo-bucket \
```

```
    --id 1
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBucketAnalyticsConfiguration。

delete-bucket-cors

以下程式碼範例顯示如何使用 delete-bucket-cors。

AWS CLI

下列命令會從名為 的儲存貯體中刪除跨來源資源共用組態amzn-s3-demo-bucket：

```
aws s3api delete-bucket-cors --bucket amzn-s3-demo-bucket
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBucketCors。

delete-bucket-encryption

以下程式碼範例顯示如何使用 delete-bucket-encryption。

AWS CLI

刪除儲存貯體的伺服器端加密組態

下列delete-bucket-encryption範例會刪除指定儲存貯體的伺服器端加密組態。

```
aws s3api delete-bucket-encryption \
    --bucket amzn-s3-demo-bucket
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBucketEncryption。

delete-bucket-intelligent-tiering-configuration

以下程式碼範例顯示如何使用 delete-bucket-intelligent-tiering-configuration。

AWS CLI

移除儲存貯體上的 S3 Intelligent-Tiering 組態

下列delete-bucket-intelligent-tiering-configuration範例會移除儲存貯體上名為 ExampleConfig 的 S3 Intelligent-Tiering 組態。

```
aws s3api delete-bucket-intelligent-tiering-configuration \
    --bucket amzn-s3-demo-bucket \
    --id ExampleConfig
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon S3 使用者指南》中的使用 S3 Intelligent-Tiering。 Amazon S3

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBucketIntelligentTieringConfiguration。

delete-bucket-inventory-configuration

以下程式碼範例顯示如何使用 delete-bucket-inventory-configuration。

AWS CLI

刪除儲存貯體的庫存組態

下列delete-bucket-inventory-configuration範例會刪除具有指定儲存貯1體 ID 的庫存組態。

```
aws s3api delete-bucket-inventory-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBucketInventoryConfiguration。

delete-bucket-lifecycle

以下程式碼範例顯示如何使用 delete-bucket-lifecycle。

AWS CLI

下列命令會從名為 的儲存貯體中刪除生命週期組態amzn-s3-demo-bucket：

```
aws s3api delete-bucket-lifecycle --bucket amzn-s3-demo-bucket
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBucketLifecycle。

delete-bucket-metrics-configuration

以下程式碼範例顯示如何使用 delete-bucket-metrics-configuration。

AWS CLI

刪除儲存貯體的指標組態

下列delete-bucket-metrics-configuration範例會移除指定儲存貯體和 ID 的指標組態。

```
aws s3api delete-bucket-metrics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 123
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBucketMetricsConfiguration。

delete-bucket-ownership-controls

以下程式碼範例顯示如何使用 delete-bucket-ownership-controls。

AWS CLI

移除儲存貯體的儲存貯體擁有權設定

下列delete-bucket-ownership-controls範例會移除儲存貯體的儲存貯體擁有權設定。

```
aws s3api delete-bucket-ownership-controls \
    --bucket amzn-s3-demo-bucket
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon S3 使用者指南》中的在現有儲存貯體上設定物件擁有權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBucketOwnershipControls。

delete-bucket-policy

以下程式碼範例顯示如何使用 delete-bucket-policy。

AWS CLI

下列命令會從名為 的儲存貯體中刪除儲存貯體政策amzn-s3-demo-bucket：

```
aws s3api delete-bucket-policy --bucket amzn-s3-demo-bucket
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBucketPolicy。

delete-bucket-replication

以下程式碼範例顯示如何使用 delete-bucket-replication。

AWS CLI

下列命令會從名為 的儲存貯體中刪除複寫組態amzn-s3-demo-bucket：

```
aws s3api delete-bucket-replication --bucket amzn-s3-demo-bucket
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBucketReplication。

delete-bucket-tagging

以下程式碼範例顯示如何使用 delete-bucket-tagging。

AWS CLI

下列命令會從名為 的儲存貯體中刪除標記組態amzn-s3-demo-bucket：

```
aws s3api delete-bucket-tagging --bucket amzn-s3-demo-bucket
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBucketTagging。

delete-bucket-website

以下程式碼範例顯示如何使用 delete-bucket-website。

AWS CLI

下列命令會從名為 的儲存貯體中刪除網站組態amzn-s3-demo-bucket：

```
aws s3api delete-bucket-website --bucket amzn-s3-demo-bucket
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteBucketWebsite。

delete-bucket

以下程式碼範例顯示如何使用 delete-bucket。

AWS CLI

下列命令會刪除名為 的儲存貯體amzn-s3-demo-bucket：

```
aws s3api delete-bucket --bucket amzn-s3-demo-bucket --region us-east-1
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteBucket。

delete-object-tagging

以下程式碼範例顯示如何使用 delete-object-tagging。

AWS CLI

刪除物件的標籤集

下列delete-object-tagging範例會從物件 刪除具有指定金鑰的標籤doc1.rtf。

```
aws s3api delete-object-tagging \
    --bucket amzn-s3-demo-bucket \
    --key doc1.rtf
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteObjectTagging。

delete-object

以下程式碼範例顯示如何使用 delete-object。

AWS CLI

下列命令test.txt會從名為 的儲存貯體中刪除名為 的物件amzn-s3-demo-bucket：

```
aws s3api delete-object --bucket amzn-s3-demo-bucket --key test.txt
```

如果已啟用儲存貯體版本控制，輸出將包含刪除標記的版本 ID：

```
{
  "VersionId": "9_gKg5vG56F.TTEUdwkxGpJ3tNDlWlGq",
  "DeleteMarker": true
}
```

如需刪除物件的詳細資訊，請參閱《Amazon S3 開發人員指南》中的刪除物件。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteObject。

delete-objects

以下程式碼範例顯示如何使用 delete-objects。

AWS CLI

下列命令會從名為 的儲存貯體中刪除物件amzn-s3-demo-bucket：

```
aws s3api delete-objects --bucket amzn-s3-demo-bucket --delete file://delete.json
```

delete.json 是目前目錄中的 JSON 文件，指定要刪除的物件：

```
{
  "Objects": [
    {
      "Key": "test1.txt"
    }
  ],
  "Quiet": false
}
```

輸出：

```
{
```

```
    "Deleted": [
        {
            "DeleteMarkerVersionId": "mYAT5Mc6F7aeUL8SS7FAAqUPO1koHwzU",
            "Key": "test1.txt",
            "DeleteMarker": true
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteObjects。

delete-public-access-block

以下程式碼範例顯示如何使用 delete-public-access-block。

AWS CLI

刪除儲存貯體的封鎖公開存取組態

下列delete-public-access-block範例會移除指定儲存貯體上的封鎖公開存取組態。

```
aws s3api delete-public-access-block \
    --bucket amzn-s3-demo-bucket
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePublicAccessBlock。

get-bucket-accelerate-configuration

以下程式碼範例顯示如何使用 get-bucket-accelerate-configuration。

AWS CLI

擷取儲存貯體的加速組態

下列get-bucket-accelerate-configuration範例會擷取指定儲存貯體的加速組態。

```
aws s3api get-bucket-accelerate-configuration \
    --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "Status": "Enabled"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketAccelerateConfiguration。

get-bucket-acl

以下程式碼範例顯示如何使用 get-bucket-acl。

AWS CLI

下列命令會擷取名為 之儲存貯體的存取控制清單amzn-s3-demo-bucket：

```
aws s3api get-bucket-acl --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "Owner": {
        "DisplayName": "my-username",
        "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "my-username",
                "ID":
 "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
            },
            "Permission": "FULL_CONTROL"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketAcl。

get-bucket-analytics-configuration

以下程式碼範例顯示如何使用 get-bucket-analytics-configuration。

AWS CLI

擷取具有特定 ID 之儲存貯體的分析組態

下列get-bucket-analytics-configuration範例顯示指定儲存貯體和 ID 的分析組態。

```
aws s3api get-bucket-analytics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1
```

輸出：

```
{
    "AnalyticsConfiguration": {
        "StorageClassAnalysis": {},
        "Id": "1"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketAnalyticsConfiguration。

get-bucket-cors

以下程式碼範例顯示如何使用 get-bucket-cors。

AWS CLI

下列命令會擷取名為 之儲存貯體的跨來源資源共用組態amzn-s3-demo-bucket：

```
aws s3api get-bucket-cors --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "CORSRules": [
        {
            "AllowedHeaders": [
                "*"
            ],
            "ExposeHeaders": [
```

```
                "x-amz-server-side-encryption"
            ],
            "AllowedMethods": [
                "PUT",
                "POST",
                "DELETE"
            ],
            "MaxAgeSeconds": 3000,
            "AllowedOrigins": [
                "http://www.example.com"
            ]
        },
        {
            "AllowedHeaders": [
                "Authorization"
            ],
            "MaxAgeSeconds": 3000,
            "AllowedMethods": [
                "GET"
            ],
            "AllowedOrigins": [
                "*"
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketCors。

get-bucket-encryption

以下程式碼範例顯示如何使用 get-bucket-encryption。

AWS CLI

擷取儲存貯體的伺服器端加密組態

下列get-bucket-encryption範例會擷取儲存貯體 的伺服器端加密組態amzn-s3-demo-bucket。

```
aws s3api get-bucket-encryption \
    --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "ServerSideEncryptionConfiguration": {
        "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "AES256"
                }
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketEncryption。

get-bucket-intelligent-tiering-configuration

以下程式碼範例顯示如何使用 get-bucket-intelligent-tiering-configuration。

AWS CLI

擷取儲存貯體上的 S3 Intelligent-Tiering 組態

下列get-bucket-intelligent-tiering-configuration範例會在儲存貯體上擷取名為 ExampleConfig 的 S3 Intelligent-Tiering 組態。

```
aws s3api get-bucket-intelligent-tiering-configuration \
    --bucket amzn-s3-demo-bucket \
    --id ExampleConfig
```

輸出：

```
{
    "IntelligentTieringConfiguration": {
        "Id": "ExampleConfig2",
        "Filter": {
            "Prefix": "images"
        },
        "Status": "Enabled",
        "Tierings": [
```

```
            {
                "Days": 90,
                "AccessTier": "ARCHIVE_ACCESS"
            },
            {
                "Days": 180,
                "AccessTier": "DEEP_ARCHIVE_ACCESS"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon S3 使用者指南》中的使用 S3 Intelligent-Tiering。 Amazon S3

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketIntelligentTieringConfiguration。

get-bucket-inventory-configuration

以下程式碼範例顯示如何使用 get-bucket-inventory-configuration。

AWS CLI

擷取儲存貯體的庫存組態

下列get-bucket-inventory-configuration範例會擷取 ID 為 之指定儲存貯體的庫存組態1。

```
aws s3api get-bucket-inventory-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1
```

輸出：

```
{
    "InventoryConfiguration": {
        "IsEnabled": true,
        "Destination": {
            "S3BucketDestination": {
                "Format": "ORC",
                "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
                "AccountId": "123456789012"
```

```
            }
        },
        "IncludedObjectVersions": "Current",
        "Id": "1",
        "Schedule": {
            "Frequency": "Weekly"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketInventoryConfiguration。

get-bucket-lifecycle-configuration

以下程式碼範例顯示如何使用 get-bucket-lifecycle-configuration。

AWS CLI

下列命令會擷取名為 之儲存貯體的生命週期組態amzn-s3-demo-bucket：

```
aws s3api get-bucket-lifecycle-configuration --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "Rules": [
        {
            "ID": "Move rotated logs to Glacier",
            "Prefix": "rotated/",
            "Status": "Enabled",
            "Transitions": [
                {
                    "Date": "2015-11-10T00:00:00.000Z",
                    "StorageClass": "GLACIER"
                }
            ]
        },
        {
            "Status": "Enabled",
            "Prefix": "",
            "NoncurrentVersionTransitions": [
                {
```

```
                    "NoncurrentDays": 0,
                    "StorageClass": "GLACIER"
                }
            ],
            "ID": "Move old versions to Glacier"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketLifecycleConfiguration。

get-bucket-lifecycle

以下程式碼範例顯示如何使用 get-bucket-lifecycle。

AWS CLI

下列命令會擷取名為 之儲存貯體的生命週期組態amzn-s3-demo-bucket：

```
aws s3api get-bucket-lifecycle --bucket amzn-s3-demo-bucket
```

輸出：

```
{
  "Rules": [
    {
      "ID": "Move to Glacier after sixty days (objects in logs/2015/)",
      "Prefix": "logs/2015/",
      "Status": "Enabled",
      "Transition": {
        "Days": 60,
        "StorageClass": "GLACIER"
      }
    },
    {
      "Expiration": {
        "Date": "2016-01-01T00:00:00.000Z"
      },
      "ID": "Delete 2014 logs in 2016.",
      "Prefix": "logs/2014/",
      "Status": "Enabled"
    }
```

```
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketLifecycle。

get-bucket-location

以下程式碼範例顯示如何使用 get-bucket-location。

AWS CLI

amzn-s3-demo-bucket如果存在限制，以下命令會擷取名為 之儲存貯體的位置限制條件：

```
aws s3api get-bucket-location --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "LocationConstraint": "us-west-2"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketLocation。

get-bucket-logging

以下程式碼範例顯示如何使用 get-bucket-logging。

AWS CLI

擷取儲存貯體的記錄狀態

下列get-bucket-logging範例會擷取指定儲存貯體的記錄狀態。

```
aws s3api get-bucket-logging \
    --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "LoggingEnabled": {
```

```
        "TargetPrefix": "",
        "TargetBucket": "amzn-s3-demo-bucket-logs"
          }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketLogging。

get-bucket-metrics-configuration

以下程式碼範例顯示如何使用 get-bucket-metrics-configuration。

AWS CLI

擷取具有特定 ID 之儲存貯體的指標組態

下列get-bucket-metrics-configuration範例顯示指定儲存貯體和 ID 的指標組態。

```
aws s3api get-bucket-metrics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 123
```

輸出：

```
{
    "MetricsConfiguration": {
        "Filter": {
            "Prefix": "logs"
        },
        "Id": "123"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketMetricsConfiguration。

get-bucket-notification-configuration

以下程式碼範例顯示如何使用 get-bucket-notification-configuration。

AWS CLI

下列命令會擷取名為 之儲存貯體的通知組態amzn-s3-demo-bucket：

```
aws s3api get-bucket-notification-configuration --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "TopicConfigurations": [
        {
            "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWNl",
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
            "Events": [
                "s3:ObjectCreated:*"
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketNotificationConfiguration。

get-bucket-notification

以下程式碼範例顯示如何使用 get-bucket-notification。

AWS CLI

下列命令會擷取名為 之儲存貯體的通知組態amzn-s3-demo-bucket：

```
aws s3api get-bucket-notification --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "TopicConfiguration": {
        "Topic": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
        "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWNl",
        "Event": "s3:ObjectCreated:*",
        "Events": [
            "s3:ObjectCreated:*"
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketNotification。

get-bucket-ownership-controls

以下程式碼範例顯示如何使用 get-bucket-ownership-controls。

AWS CLI

擷取儲存貯體的儲存貯體擁有權設定

下列get-bucket-ownership-controls範例會擷取儲存貯體的儲存貯體擁有權設定。

```
aws s3api get-bucket-ownership-controls \
    --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "OwnershipControls": {
        "Rules": [
            {
                "ObjectOwnership": "BucketOwnerEnforced"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Amazon S3 使用者指南》中的檢視 S3 儲存貯體的物件擁有權設定。Amazon S3

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketOwnershipControls。

get-bucket-policy-status

以下程式碼範例顯示如何使用 get-bucket-policy-status。

AWS CLI

擷取儲存貯體的政策狀態，指出儲存貯體是否為公有

下列get-bucket-policy-status範例會擷取儲存貯體 的政策狀態amzn-s3-demo-bucket。

```
aws s3api get-bucket-policy-status \
    --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "PolicyStatus": {
        "IsPublic": false
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketPolicyStatus。

get-bucket-policy

以下程式碼範例顯示如何使用 get-bucket-policy。

AWS CLI

下列命令會擷取名為 之儲存貯體的儲存貯體政策amzn-s3-demo-bucket：

```
aws s3api get-bucket-policy --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "Policy": "{\"Version\":\"2008-10-17\",\"Statement\":[{\"Sid\":\"\",\"Effect
\":\"Allow\",\"Principal\":\"*\",\"Action\":\"s3:GetObject\",\"Resource\":
\"arn:aws:s3:::amzn-s3-demo-bucket/*\"},{\"Sid\":\"\",\"Effect\":\"Deny\",
\"Principal\":\"*\",\"Action\":\"s3:GetObject\",\"Resource\":\"arn:aws:s3:::amzn-s3-
demo-bucket/secret/*\"}]}"
}
```

取得並放置儲存貯體policyThe範例示範如何下載 Amazon S3 儲存貯體政策、修改檔案，然後使用 put-bucket-policy 套用修改過的儲存貯體政策。若要將儲存貯體政策下載至檔案，您可以執行：

```
aws s3api get-bucket-policy --bucket amzn-s3-demo-bucket --query Policy --output
 text > policy.json
```

然後，您可以視需要修改policy.json檔案。最後，您可以執行下列動作，將此修改後的政策套用回 S3 儲存貯體：

policy.json 檔案。最後，您可以執行下列動作，將此修改後的政策套用回 S3 儲存貯體：

檔案。最後，您可以執行下列動作，將此修改後的政策套用回 S3 儲存貯體：

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy file://policy.json
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketPolicy。

get-bucket-replication

以下程式碼範例顯示如何使用 get-bucket-replication。

AWS CLI

下列命令會擷取名為 之儲存貯體的複寫組態amzn-s3-demo-bucket：

```
aws s3api get-bucket-replication --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "ReplicationConfiguration": {
        "Rules": [
            {
                "Status": "Enabled",
                "Prefix": "",
                "Destination": {
                    "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket-backup",
                    "StorageClass": "STANDARD"
                },
                "ID": "ZmUwNzE4ZmQ4tMjVhOS00MTlkLOGI4NDkzZTIWJjNTUtYTA1"
            }
        ],
        "Role": "arn:aws:iam::123456789012:role/s3-replication-role"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketReplication。

get-bucket-request-payment

以下程式碼範例顯示如何使用 get-bucket-request-payment。

AWS CLI

擷取儲存貯體的請求付款組態

下列get-bucket-request-payment範例會擷取請求者支付指定儲存貯體的組態。

```
aws s3api get-bucket-request-payment \
    --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "Payer": "BucketOwner"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketRequestPayment。

get-bucket-tagging

以下程式碼範例顯示如何使用 get-bucket-tagging。

AWS CLI

下列命令會擷取名為 之儲存貯體的標記組態amzn-s3-demo-bucket：

```
aws s3api get-bucket-tagging --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "TagSet": [
        {
            "Value": "marketing",
            "Key": "organization"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketTagging。

get-bucket-versioning

以下程式碼範例顯示如何使用 get-bucket-versioning。

AWS CLI

下列命令會擷取名為 之儲存貯體的版本控制組態amzn-s3-demo-bucket：

```
aws s3api get-bucket-versioning --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "Status": "Enabled"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketVersioning。

get-bucket-website

以下程式碼範例顯示如何使用 get-bucket-website。

AWS CLI

下列命令會擷取名為 之儲存貯體的靜態網站組態amzn-s3-demo-bucket：

```
aws s3api get-bucket-website --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "IndexDocument": {
        "Suffix": "index.html"
    },
    "ErrorDocument": {
        "Key": "error.html"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetBucketWebsite。

get-object-acl

以下程式碼範例顯示如何使用 get-object-acl。

AWS CLI

下列命令會擷取名為 之儲存貯體中物件的存取控制清單amzn-s3-demo-bucket：

```
aws s3api get-object-acl --bucket amzn-s3-demo-bucket --key index.html
```

輸出：

```
{
    "Owner": {
        "DisplayName": "my-username",
        "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "my-username",
                "ID":
 "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
            },
            "Permission": "FULL_CONTROL"
        },
        {
            "Grantee": {
                "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
            },
            "Permission": "READ"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetObjectAcl。

get-object-attributes

以下程式碼範例顯示如何使用 get-object-attributes。

AWS CLI

從物件擷取中繼資料而不傳回物件本身

下列get-object-attributes範例會從物件 擷取中繼資料doc1.rtf。

```
aws s3api get-object-attributes \
    --bucket amzn-s3-demo-bucket \
    --key doc1.rtf \
    --object-attributes "StorageClass" "ETag" "ObjectSize"
```

輸出：

```
{
    "LastModified": "2022-03-15T19:37:31+00:00",
    "VersionId": "IuCPjXTDzHNfldAuitVBIKJpF2p1fg4P",
    "ETag": "b662d79adeb7c8d787ea7eafb9ef6207",
    "StorageClass": "STANDARD",
    "ObjectSize": 405
}
```

如需詳細資訊，請參閱《Amazon S3 API 參考》中的 GetObjectAttributes。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetObjectAttributes。

get-object-legal-hold

以下程式碼範例顯示如何使用 get-object-legal-hold。

AWS CLI

擷取物件的法務保存狀態

下列get-object-legal-hold範例會擷取指定物件的法務保存狀態。

```
aws s3api get-object-legal-hold \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --key doc1.rtf
```

輸出：

```
{
    "LegalHold": {
        "Status": "ON"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetObjectLegalHold。

get-object-lock-configuration

以下程式碼範例顯示如何使用 get-object-lock-configuration。

AWS CLI

擷取儲存貯體的物件鎖定組態

下列get-object-lock-configuration範例會擷取指定儲存貯體的物件鎖定組態。

```
aws s3api get-object-lock-configuration \
    --bucket amzn-s3-demo-bucket-with-object-lock
```

輸出：

```
{
    "ObjectLockConfiguration": {
        "ObjectLockEnabled": "Enabled",
        "Rule": {
            "DefaultRetention": {
                "Mode": "COMPLIANCE",
                "Days": 50
            }
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetObjectLockConfiguration。

get-object-retention

以下程式碼範例顯示如何使用 get-object-retention。

AWS CLI

擷取物件的物件保留組態

下列get-object-retention範例會擷取指定物件的物件保留組態。

```
aws s3api get-object-retention \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --key doc1.rtf
```

輸出：

```
{
    "Retention": {
        "Mode": "GOVERNANCE",
        "RetainUntilDate": "2025-01-01T00:00:00.000Z"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetObjectRetention。

get-object-tagging

以下程式碼範例顯示如何使用 get-object-tagging。

AWS CLI

擷取連接至物件的標籤

下列get-object-tagging範例會從指定的物件擷取指定金鑰的值。

```
aws s3api get-object-tagging \
    --bucket amzn-s3-demo-bucket \
    --key doc1.rtf
```

輸出：

```
{
    "TagSet": [
        {
            "Value": "confidential",
```

```
            "Key": "designation"
        }
    ]
}
```

下列get-object-tagging範例會嘗試擷取doc2.rtf沒有標籤之物件 的標籤集。

```
aws s3api get-object-tagging \
    --bucket amzn-s3-demo-bucket \
    --key doc2.rtf
```

輸出：

```
{
    "TagSet": []
}
```

下列get-object-tagging範例會擷取doc3.rtf具有多個標籤之物件 的標籤集。

```
aws s3api get-object-tagging \
    --bucket amzn-s3-demo-bucket \
    --key doc3.rtf
```

輸出：

```
{
    "TagSet": [
        {
            "Value": "confidential",
            "Key": "designation"
        },
        {
            "Value": "finance",
            "Key": "department"
        },
        {
            "Value": "payroll",
            "Key": "team"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetObjectTagging。

get-object-torrent

以下程式碼範例顯示如何使用 get-object-torrent。

AWS CLI

下列命令會為名為 的儲存貯體中的物件建立 torrentamzn-s3-demo-bucket：

```
aws s3api get-object-torrent --bucket amzn-s3-demo-bucket --key large-video-
file.mp4 large-video-file.torrent
```

Torrent 檔案會儲存在本機目前的資料夾中。請注意，輸出檔案名稱 (large-video-file.torrent) 是在沒有選項名稱的情況下指定，且必須是 命令中的最後一個引數。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetObjectTorrent。

get-object

以下程式碼範例顯示如何使用 get-object。

AWS CLI

下列範例使用 get-object命令從 Amazon S3 下載物件：

```
aws s3api get-object --bucket text-content --key dir/
my_images.tar.bz2 my_images.tar.bz2
```

請注意，outfile 參數是在沒有選項名稱的情況下指定，例如 "--outfile"。輸出檔案名稱必須是 命令中的最後一個參數。

以下範例示範如何使用 從 物件--range下載特定位元組範圍。請注意，位元組範圍必須以 "bytes="：

```
aws s3api get-object --bucket text-content --key dir/my_data --
range bytes=8888-9999 my_data_range
```

如需擷取物件的詳細資訊，請參閱《Amazon S3 開發人員指南》中的取得物件。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetObject。

get-public-access-block

以下程式碼範例顯示如何使用 get-public-access-block。

AWS CLI

設定或修改儲存貯體的封鎖公開存取組態

下列get-public-access-block範例顯示指定儲存貯體的封鎖公開存取組態。

```
aws s3api get-public-access-block \
    --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "PublicAccessBlockConfiguration": {
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "BlockPublicAcls": true,
        "RestrictPublicBuckets": true
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPublicAccessBlock。

head-bucket

以下程式碼範例顯示如何使用 head-bucket。

AWS CLI

下列命令會驗證對名為 之儲存貯體的存取權amzn-s3-demo-bucket：

```
aws s3api head-bucket --bucket amzn-s3-demo-bucket
```

如果儲存貯體存在且您有權存取，則不會傳回任何輸出。否則會顯示錯誤訊息。例如：

```
A client error (404) occurred when calling the HeadBucket operation: Not Found
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 HeadBucket。

head-object

以下程式碼範例顯示如何使用 head-object。

AWS CLI

下列命令會擷取名為 之儲存貯體中物件的中繼資料amzn-s3-demo-bucket：

```
aws s3api head-object --bucket amzn-s3-demo-bucket --key index.html
```

輸出：

```
{
    "AcceptRanges": "bytes",
    "ContentType": "text/html",
    "LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",
    "ContentLength": 77,
    "VersionId": "null",
    "ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
    "Metadata": {}
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 HeadObject。

list-bucket-analytics-configurations

以下程式碼範例顯示如何使用 list-bucket-analytics-configurations。

AWS CLI

擷取儲存貯體的分析組態清單

以下內容會list-bucket-analytics-configurations擷取指定儲存貯體的分析組態清單。

```
aws s3api list-bucket-analytics-configurations \
    --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "AnalyticsConfigurationList": [
        {
            "StorageClassAnalysis": {},
            "Id": "1"
        }
    ],
    "IsTruncated": false
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBucketAnalyticsConfigurations。

`list-bucket-intelligent-tiering-configurations`

以下程式碼範例顯示如何使用 `list-bucket-intelligent-tiering-configurations`。

AWS CLI

擷取儲存貯體上的所有 S3 Intelligent-Tiering 組態

下列`list-bucket-intelligent-tiering-configurations`範例會擷取儲存貯體上的所有 S3 Intelligent-Tiering 組態。

```
aws s3api list-bucket-intelligent-tiering-configurations \
    --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "IsTruncated": false,
    "IntelligentTieringConfigurationList": [
        {
            "Id": "ExampleConfig",
            "Filter": {
                "Prefix": "images"
            },
            "Status": "Enabled",
            "Tierings": [
                {
                    "Days": 90,
                    "AccessTier": "ARCHIVE_ACCESS"
```

```
                },
                {
                    "Days": 180,
                    "AccessTier": "DEEP_ARCHIVE_ACCESS"
                }
            ]
        },
        {
            "Id": "ExampleConfig2",
            "Status": "Disabled",
            "Tierings": [
                {
                    "Days": 730,
                    "AccessTier": "ARCHIVE_ACCESS"
                }
            ]
        },
        {
            "Id": "ExampleConfig3",
            "Filter": {
                "Tag": {
                    "Key": "documents",
                    "Value": "taxes"
                }
            },
            "Status": "Enabled",
            "Tierings": [
                {
                    "Days": 90,
                    "AccessTier": "ARCHIVE_ACCESS"
                },
                {
                    "Days": 365,
                    "AccessTier": "DEEP_ARCHIVE_ACCESS"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon S3 使用者指南》中的使用 S3 Intelligent-Tiering。 Amazon S3

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBucketIntelligentTieringConfigurations。

list-bucket-inventory-configurations

以下程式碼範例顯示如何使用 list-bucket-inventory-configurations。

AWS CLI

擷取儲存貯體的庫存組態清單

下列list-bucket-inventory-configurations範例列出指定儲存貯體的庫存組態。

```
aws s3api list-bucket-inventory-configurations \
    --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "InventoryConfigurationList": [
        {
            "IsEnabled": true,
            "Destination": {
                "S3BucketDestination": {
                    "Format": "ORC",
                    "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
                    "AccountId": "123456789012"
                }
            },
            "IncludedObjectVersions": "Current",
            "Id": "1",
            "Schedule": {
                "Frequency": "Weekly"
            }
        },
        {
            "IsEnabled": true,
            "Destination": {
                "S3BucketDestination": {
                    "Format": "CSV",
                    "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
                    "AccountId": "123456789012"
                }
            },
            "IncludedObjectVersions": "Current",
            "Id": "2",
```

```
            "Schedule": {
                "Frequency": "Daily"
            }
        }
    ],
    "IsTruncated": false
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBucketInventoryConfigurations。

`list-bucket-metrics-configurations`

以下程式碼範例顯示如何使用 `list-bucket-metrics-configurations`。

AWS CLI

擷取儲存貯體的指標組態清單

下列`list-bucket-metrics-configurations`範例會擷取指定儲存貯體的指標組態清單。

```
aws s3api list-bucket-metrics-configurations \
    --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "IsTruncated": false,
    "MetricsConfigurationList": [
        {
            "Filter": {
                "Prefix": "logs"
            },
            "Id": "123"
        },
        {
            "Filter": {
                "Prefix": "tmp"
            },
            "Id": "234"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBucketMetricsConfigurations。

list-buckets

以下程式碼範例顯示如何使用 list-buckets。

AWS CLI

下列命令使用 list-buckets命令來顯示所有 Amazon S3 儲存貯體的名稱 （跨所有區域）：

```
aws s3api list-buckets --query "Buckets[].Name"
```

查詢選項會將 的輸出list-buckets向下篩選為僅儲存貯體名稱。

如需儲存貯體的詳細資訊，請參閱《Amazon S3 開發人員指南》中的使用 Amazon S3 儲存貯體。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListBuckets。

list-multipart-uploads

以下程式碼範例顯示如何使用 list-multipart-uploads。

AWS CLI

下列命令會列出名為 之儲存貯體的所有作用中分段上傳amzn-s3-demo-bucket：

```
aws s3api list-multipart-uploads --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "Uploads": [
        {
            "Initiator": {
                "DisplayName": "username",
                "ID": "arn:aws:iam::0123456789012:user/username"
            },
            "Initiated": "2015-06-02T18:01:30.000Z",
            "UploadId":
 "dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3URC
            "StorageClass": "STANDARD",
            "Key": "multipart/01",
```

```
            "Owner": {
                "DisplayName": "aws-account-name",
                "ID":
 "100719349fc3b6dcd7c820a124bf7aecd408092c3d7b51b38494939801fc248b"
            }
        }
    ],
    "CommonPrefixes": []
}
```

進行中的分段上傳會在 Amazon S3 中產生儲存成本。完成或中止作用中的分段上傳，以從您的帳戶中移除其組件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMultipartUploads。

list-object-versions

以下程式碼範例顯示如何使用 `list-object-versions`。

AWS CLI

下列命令會擷取名為 之儲存貯體中物件的版本資訊amzn-s3-demo-bucket：

```
aws s3api list-object-versions --bucket amzn-s3-demo-bucket --prefix index.html
```

輸出：

```
{
    "DeleteMarkers": [
        {
            "Owner": {
                "DisplayName": "my-username",
                "ID":
 "7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
            },
            "IsLatest": true,
            "VersionId": "B2VsEK5saUNNHKcOAJj7hIE86RozToyq",
            "Key": "index.html",
            "LastModified": "2015-11-10T00:57:03.000Z"
        },
        {
            "Owner": {
                "DisplayName": "my-username",
```

```
                "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
            },
            "IsLatest": false,
            "VersionId": ".FLQEZscLIcfxSq.jsFJ.szUkmng2Yw6",
            "Key": "index.html",
            "LastModified": "2015-11-09T23:32:20.000Z"
        }
    ],
    "Versions": [
        {
            "LastModified": "2015-11-10T00:20:11.000Z",
            "VersionId": "Rb_l2T8UHDkFEwCgJjhlgPOZC0qJ.vpD",
            "ETag": "\"0622528de826c0df5db1258a23b80be5\"",
            "StorageClass": "STANDARD",
            "Key": "index.html",
            "Owner": {
                "DisplayName": "my-username",
                "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
            },
            "IsLatest": false,
            "Size": 38
        },
        {
            "LastModified": "2015-11-09T23:26:41.000Z",
            "VersionId": "rasWWGpgk9E4s0LyTJgusGeRQKLVIAFf",
            "ETag": "\"06225825b8028de826c0df5db1a23be5\"",
            "StorageClass": "STANDARD",
            "Key": "index.html",
            "Owner": {
                "DisplayName": "my-username",
                "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
            },
            "IsLatest": false,
            "Size": 38
        },
        {
            "LastModified": "2015-11-09T22:50:50.000Z",
            "VersionId": "null",
            "ETag": "\"d1f45267a863c8392e07d24dd592f1b9\"",
            "StorageClass": "STANDARD",
            "Key": "index.html",
```

```
            "Owner": {
                "DisplayName": "my-username",
                "ID":
 "7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
            },
            "IsLatest": false,
            "Size": 533823
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListObjectVersions。

list-objects-v2

以下程式碼範例顯示如何使用 list-objects-v2。

AWS CLI

若要取得儲存貯體中的物件清單

下列list-objects-v2範例列出指定儲存貯體中的物件。

```
aws s3api list-objects-v2 \
    --bucket amzn-s3-demo-bucket
```

輸出：

```
{
    "Contents": [
        {
            "LastModified": "2019-11-05T23:11:50.000Z",
            "ETag": "\"621503c373607d548b37cff8778d992c\"",
            "StorageClass": "STANDARD",
            "Key": "doc1.rtf",
            "Size": 391
        },
        {
            "LastModified": "2019-11-05T23:11:50.000Z",
            "ETag": "\"a2cecc36ab7c7fe3a71a273b9d45b1b5\"",
            "StorageClass": "STANDARD",
            "Key": "doc2.rtf",
```

```
            "Size": 373
        },
        {
            "LastModified": "2019-11-05T23:11:50.000Z",
            "ETag": "\"08210852f65a2e9cb999972539a64d68\"",
            "StorageClass": "STANDARD",
            "Key": "doc3.rtf",
            "Size": 399
        },
        {
            "LastModified": "2019-11-05T23:11:50.000Z",
            "ETag": "\"d1852dd683f404306569471af106988e\"",
            "StorageClass": "STANDARD",
            "Key": "doc4.rtf",
            "Size": 6225
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListObjectsV2。

list-objects

以下程式碼範例顯示如何使用 list-objects。

AWS CLI

下列範例使用 list-objects 命令來顯示指定儲存貯體中所有物件的名稱：

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size:
 Size}'
```

此範例使用 --query 引數，將 list-objects 的輸出篩選為每個物件的金鑰值和大小

如需物件的詳細資訊，請參閱《Amazon S3 開發人員指南》中的使用 Amazon S3 物件。

- 如需 API 詳細資訊，請參閱《AWS CLI API 參考》中的 ListObjects。

list-parts

以下程式碼範例顯示如何使用 list-parts。

AWS CLI

下列命令會列出在儲存貯體 multipart/01中使用金鑰 上傳分段上傳的所有部分amzn-s3-demo-bucket：

```
aws s3api list-parts --bucket amzn-s3-demo-bucket --key 'multipart/01' --upload-
id dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

輸出：

```
{
    "Owner": {
        "DisplayName": "aws-account-name",
        "ID": "100719349fc3b6dcd7c820a124bf7aecd408092c3d7b51b38494939801fc248b"
    },
    "Initiator": {
        "DisplayName": "username",
        "ID": "arn:aws:iam::0123456789012:user/username"
    },
    "Parts": [
        {
            "LastModified": "2015-06-02T18:07:35.000Z",
            "PartNumber": 1,
            "ETag": "\"e868e0f4719e394144ef36531ee6824c\"",
            "Size": 5242880
        },
        {
            "LastModified": "2015-06-02T18:07:42.000Z",
            "PartNumber": 2,
            "ETag": "\"6bb2b12753d66fe86da4998aa33fffb0\"",
            "Size": 5242880
        },
        {
            "LastModified": "2015-06-02T18:07:47.000Z",
            "PartNumber": 3,
            "ETag": "\"d0a0112e841abec9c9ec83406f0159c8\"",
            "Size": 5242880
        }
    ],
    "StorageClass": "STANDARD"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListParts。

ls

以下程式碼範例顯示如何使用 ls。

AWS CLI

範例 1：列出所有使用者擁有的儲存貯體

下列ls命令會列出使用者擁有的所有儲存貯體。在此範例中，使用者擁有儲存貯體 amzn-s3-demo-bucket和 amzn-s3-demo-bucket2。時間戳記是建立儲存貯體的日期，顯示在機器的時區中。此日期可能會在變更儲存貯體時變更，例如編輯其儲存貯體政策。請注意，如果 s3:// 用於路徑引數 <S3Uri>，它也會列出所有儲存貯體。

```
aws s3 ls
```

輸出：

```
2013-07-11 17:08:50 amzn-s3-demo-bucket
2013-07-24 14:55:44 amzn-s3-demo-bucket2
```

範例 2：列出儲存貯體中的所有字首和物件

下列ls命令會列出指定儲存貯體和字首下的物件和常見字首。在此範例中，使用者擁有amzn-s3-demo-bucket具有物件 test.txt和 的儲存貯體somePrefix/test.txt。LastWriteTime 和 Length是任意的。請注意，由於ls命令與本機檔案系統沒有互動，因此不需要 s3:// URI 結構描述來解決模棱兩可的問題，而且可以省略。

```
aws s3 ls s3://amzn-s3-demo-bucket
```

輸出：

```
                           PRE somePrefix/
2013-07-25 17:06:27         88 test.txt
```

範例 3：列出特定儲存貯體和字首中的所有字首和物件

下列ls命令會列出指定儲存貯體和字首下的物件和常見字首。不過，指定的儲存貯體和字首下沒有物件或通用字首。

```
aws s3 ls s3://amzn-s3-demo-bucket/noExistPrefix
```

輸出：

```
None
```

範例 4：遞迴列出儲存貯體中的所有字首和物件

下列ls命令會遞迴列出儲存貯體中的物件。儲存貯體中的所有內容都會依序列出，而不是顯示在輸出PRE dirname/中。

```
aws s3 ls s3://amzn-s3-demo-bucket \
    --recursive
```

輸出：

```
2013-09-02 21:37:53         10 a.txt
2013-09-02 21:37:53    2863288 foo.zip
2013-09-02 21:32:57         23 foo/bar/.baz/a
2013-09-02 21:32:58         41 foo/bar/.baz/b
2013-09-02 21:32:57        281 foo/bar/.baz/c
2013-09-02 21:32:57         73 foo/bar/.baz/d
2013-09-02 21:32:57        452 foo/bar/.baz/e
2013-09-02 21:32:57        896 foo/bar/.baz/hooks/bar
2013-09-02 21:32:57        189 foo/bar/.baz/hooks/foo
2013-09-02 21:32:57        398 z.txt
```

範例 5：摘要儲存貯體中的所有字首和物件

下列ls命令示範使用 --human-readable 和 --summarize 選項的相同命令。--human-readable 會以 Bytes/MiB/KiB/GiB/TiB/PiB/EiB 顯示檔案大小。--summarize 會在結果清單結尾顯示物件總數和大小總計：

```
aws s3 ls s3://amzn-s3-demo-bucket \
    --recursive \
    --human-readable \
    --summarize
```

輸出：

```
2013-09-02 21:37:53   10 Bytes a.txt
2013-09-02 21:37:53  2.9 MiB foo.zip
2013-09-02 21:32:57   23 Bytes foo/bar/.baz/a
```

```
2013-09-02 21:32:58   41 Bytes foo/bar/.baz/b
2013-09-02 21:32:57  281 Bytes foo/bar/.baz/c
2013-09-02 21:32:57   73 Bytes foo/bar/.baz/d
2013-09-02 21:32:57  452 Bytes foo/bar/.baz/e
2013-09-02 21:32:57  896 Bytes foo/bar/.baz/hooks/bar
2013-09-02 21:32:57  189 Bytes foo/bar/.baz/hooks/foo
2013-09-02 21:32:57  398 Bytes z.txt

Total Objects: 10
   Total Size: 2.9 MiB
```

範例 6：從 S3 存取點列出

下列ls命令列出來自存取點 () 的物件myaccesspoint：

```
aws s3 ls s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
```

輸出：

```
                           PRE somePrefix/
2013-07-25 17:06:27         88 test.txt
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 Ls。

mb

以下程式碼範例顯示如何使用 mb。

AWS CLI

範例 1：建立儲存貯體

下列mb命令會建立儲存貯體。在此範例中，使用者將儲存貯體設為 amzn-s3-demo-bucket。儲存貯體是在使用者組態檔案中指定的區域中建立：

```
aws s3 mb s3://amzn-s3-demo-bucket
```

輸出：

```
make_bucket: s3://amzn-s3-demo-bucket
```

範例 2：在指定區域中建立儲存貯體

下列mb命令會在 --region 參數指定的區域中建立儲存貯體。在此範例中，使用者在區域 中製作儲存貯amzn-s3-demo-bucket體us-west-1：

```
aws s3 mb s3://amzn-s3-demo-bucket \
    --region us-west-1
```

輸出：

```
make_bucket: s3://amzn-s3-demo-bucket
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 Mb。

mv

以下程式碼範例顯示如何使用 mv。

AWS CLI

範例 1：將本機檔案移至指定的儲存貯體

下列mv命令會將單一檔案移至指定的儲存貯體和金鑰。

```
aws s3 mv test.txt s3://amzn-s3-demo-bucket/test2.txt
```

輸出：

```
move: test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

範例 2：將物件移至指定的儲存貯體和金鑰

下列mv命令會將單一 s3 物件移至指定的儲存貯體和金鑰。

```
aws s3 mv s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket/test2.txt
```

輸出：

```
move: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

範例 3：將 S3 物件移至本機目錄

下列mv命令會將單一物件移至本機指定的檔案。

```
aws s3 mv s3://amzn-s3-demo-bucket/test.txt test2.txt
```

輸出：

```
move: s3://amzn-s3-demo-bucket/test.txt to test2.txt
```

範例 4：將具有原始名稱的物件移至指定的儲存貯體

下列mv命令會將單一物件移至指定的儲存貯體，同時保留其原始名稱：

```
aws s3 mv s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket2/
```

輸出：

```
move: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket2/test.txt
```

範例 5：將儲存貯體中的所有物件和字首移至本機目錄

與 參數 一起傳遞時--recursive，下列mv命令會遞迴地將所有指定字首和儲存貯體下的物件移至指定的目錄。在此範例中，儲存貯amzn-s3-demo-bucket體具有 物件 test1.txt和 test2.txt。

```
aws s3 mv s3://amzn-s3-demo-bucket . \
    --recursive
```

輸出：

```
move: s3://amzn-s3-demo-bucket/test1.txt to test1.txt
move: s3://amzn-s3-demo-bucket/test2.txt to test2.txt
```

範例 6：將儲存貯體中的所有物件和字首移至本機目錄，但 ``.jpg`` 檔案除外

使用 參數 傳遞時--recursive，下列mv命令會遞迴移動指定目錄下的所有檔案至指定的儲存貯體和字首，同時使用 --exclude 參數排除某些檔案。在此範例中， 目錄myDir具有 檔案test1.txt和 test2.jpg。

```
aws s3 mv myDir s3://amzn-s3-demo-bucket/ \
    --recursive \
    --exclude "*.jpg"
```

輸出：

```
move: myDir/test1.txt to s3://amzn-s3-demo-bucket2/test1.txt
```

範例 7：將儲存貯體中的所有物件和字首移至本機目錄，但指定的字首除外

使用 參數 傳遞時--recursive，下列mv命令會遞迴移動指定儲存貯體下的所有物件至另一個儲存貯體，同時使用 --exclude 參數排除某些物件。在此範例中，儲存貯amzn-s3-demo-bucket體具有 物件 test1.txt和 another/test1.txt。

```
aws s3 mv s3://amzn-s3-demo-bucket/ s3://amzn-s3-demo-bucket2/ \
    --recursive \
    --exclude "amzn-s3-demo-bucket/another/*"
```

輸出：

```
move: s3://amzn-s3-demo-bucket/test1.txt to s3://amzn-s3-demo-bucket2/test1.txt
```

範例 8：將物件移至指定的儲存貯體，並設定 ACL

下列mv命令會將 ACL 設定為 時，將單一物件移至指定的儲存貯體和金鑰public-read-write。

```
aws s3 mv s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket/test2.txt \
    --acl public-read-write
```

輸出：

```
move: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

範例 9：將本機檔案移至指定的儲存貯體並授予許可

下列mv命令說明使用 --grants選項，將讀取存取權授予所有使用者，並完全控制由其電子郵件地址識別的特定使用者。

```
aws s3 mv file.txt s3://amzn-s3-demo-bucket/ \
```

```
    --grants read=uri=http://acs.amazonaws.com/groups/global/
AllUsers full=emailaddress=user@example.com
```

輸出：

```
move: file.txt to s3://amzn-s3-demo-bucket/file.txt
```

範例 10：將檔案移至 S3 存取點

下列mv命令mydoc.txt會將名為 的單一檔案移至名為 之金鑰myaccesspoint的存取點mykey。

```
aws s3 mv mydoc.txt s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/mykey
```

輸出：

```
move: mydoc.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
mykey
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 Mv。

presign

以下程式碼範例顯示如何使用 presign。

AWS CLI

範例 1：使用連結至 S3 儲存貯體中物件的預設一小時生命週期建立預先簽章的 URL

下列presign命令會為指定的儲存貯體和金鑰產生預先簽章的 URL，有效期為一小時。

```
aws s3 presign s3://amzn-s3-demo-bucket/test2.txt
```

輸出：

```
https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/key?X-Amz-Algorithm=AWS4-
HMAC-SHA256&X-Amz-Credential=AKIAEXAMPLE123456789%2F20210621%2Fus-
west-2%2Fs3%2Faws4_request&X-Amz-Date=20210621T041609Z&X-Amz-Expires=3600&X-Amz-
SignedHeaders=host&X-Amz-
Signature=EXAMBLE1234494d5fba3fed607f98018e1dfc62e2529ae96d844123456
```

範例 2：建立具有連結至 S3 儲存貯體中物件之自訂生命週期的預先簽章 URL

下列presign命令會為指定的儲存貯體和金鑰產生預先簽章的 URL，有效期間為一週。

```
aws s3 presign s3://amzn-s3-demo-bucket/test2.txt \
    --expires-in 604800
```

輸出：

```
https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/key?X-Amz-Algorithm=AWS4-
HMAC-SHA256&X-Amz-Credential=AKIAEXAMPLE123456789%2F20210621%2Fus-
west-2%2Fs3%2Faws4_request&X-Amz-Date=20210621T041609Z&X-Amz-Expires=604800&X-Amz-
SignedHeaders=host&X-Amz-
Signature=EXAMBLE1234494d5fba3fed607f98018e1dfc62e2529ae96d844123456
```

如需詳細資訊，請參閱《S3 開發人員指南》中的與其他人共用物件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的預先簽章。

put-bucket-accelerate-configuration

以下程式碼範例顯示如何使用 put-bucket-accelerate-configuration。

AWS CLI

設定儲存貯體的加速組態

下列put-bucket-accelerate-configuration範例會啟用指定儲存貯體的加速組態。

```
aws s3api put-bucket-accelerate-configuration \
    --bucket amzn-s3-demo-bucket \
    --accelerate-configuration Status=Enabled
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketAccelerateConfiguration。

put-bucket-acl

以下程式碼範例顯示如何使用 put-bucket-acl。

AWS CLI

此範例full control授予兩個 AWS 使用者 (user1@example.com 和 user2@example.com) 和read許可給每個人：

```
aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --grant-full-
control emailaddress=user1@example.com,emailaddress=user2@example.com --grant-
read uri=http://acs.amazonaws.com/groups/global/AllUsers
```

如需自訂 ACLs(s3api ACL 命令，例如 put-bucket-acl，請使用相同的速記引數表示法），請參閱 http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html：//。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketAcl。

put-bucket-analytics-configuration

以下程式碼範例顯示如何使用 put-bucket-analytics-configuration。

AWS CLI

設定儲存貯體的分析組態

下列put-bucket-analytics-configuration範例會設定指定儲存貯體的分析。

```
aws s3api put-bucket-analytics-configuration \
    --bucket amzn-s3-demo-bucket --id 1 \
    --analytics-configuration '{"Id": "1","StorageClassAnalysis": {}}'
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketAnalyticsConfiguration。

put-bucket-cors

以下程式碼範例顯示如何使用 put-bucket-cors。

AWS CLI

下列範例會啟用來自 www.example.com PUT的 POST、 和 DELETE請求，並啟用來自任何網域的GET請求：

```
aws s3api put-bucket-cors --bucket amzn-s3-demo-bucket --cors-configuration file://
cors.json

cors.json:
{
  "CORSRules": [
    {
      "AllowedOrigins": ["http://www.example.com"],
      "AllowedHeaders": ["*"],
      "AllowedMethods": ["PUT", "POST", "DELETE"],
      "MaxAgeSeconds": 3000,
      "ExposeHeaders": ["x-amz-server-side-encryption"]
    },
    {
      "AllowedOrigins": ["*"],
      "AllowedHeaders": ["Authorization"],
      "AllowedMethods": ["GET"],
      "MaxAgeSeconds": 3000
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketCors。

put-bucket-encryption

以下程式碼範例顯示如何使用 put-bucket-encryption。

AWS CLI

設定儲存貯體的伺服器端加密

下列put-bucket-encryption範例會將 AES256 加密設定為指定儲存貯體的預設值。

```
aws s3api put-bucket-encryption \
    --bucket amzn-s3-demo-bucket \
    --server-side-encryption-configuration '{"Rules":
 [{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}}]}'
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketEncryption。

put-bucket-intelligent-tiering-configuration

以下程式碼範例顯示如何使用 put-bucket-intelligent-tiering-configuration。

AWS CLI

更新儲存貯體上的 S3 Intelligent-Tiering 組態

下列put-bucket-intelligent-tiering-configuration範例會在儲存貯體上更新名為 ExampleConfig 的 S3 Intelligent-Tiering 組態。組態會在 90 天後將未在字首影像下存取的物件轉換為封存存取，並在 180 天後將 Deep Archive Access 轉換為封存存取。

```
aws s3api put-bucket-intelligent-tiering-configuration \
    --bucket amzn-s3-demo-bucket \
    --id "ExampleConfig" \
    --intelligent-tiering-configuration file://intelligent-tiering-
configuration.json
```

intelligent-tiering-configuration.json 的內容：

```
{
    "Id": "ExampleConfig",
    "Status": "Enabled",
    "Filter": {
        "Prefix": "images"
        },
    "Tierings": [
        {
            "Days": 90,
            "AccessTier": "ARCHIVE_ACCESS"
        },
        {
            "Days": 180,
            "AccessTier": "DEEP_ARCHIVE_ACCESS"
        }
    ]
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon S3 使用者指南》中的在現有儲存貯體上設定物件擁有權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketIntelligentTieringConfiguration。

put-bucket-inventory-configuration

以下程式碼範例顯示如何使用 put-bucket-inventory-configuration。

AWS CLI

範例 1：設定儲存貯體的庫存組態

下列put-bucket-inventory-configuration範例會設定儲存貯體 的每週 ORC 格式庫存報告amzn-s3-demo-bucket。

```
aws s3api put-bucket-inventory-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1 \
    --inventory-configuration '{"Destination": { "S3BucketDestination":
 { "AccountId": "123456789012", "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
 "Format": "ORC" }}, "IsEnabled": true, "Id": "1", "IncludedObjectVersions":
 "Current", "Schedule": { "Frequency": "Weekly" }}'
```

此命令不會產生輸出。

範例 2：設定儲存貯體的庫存組態

下列put-bucket-inventory-configuration範例會設定儲存貯體 的每日 CSV 格式庫存報告amzn-s3-demo-bucket。

```
aws s3api put-bucket-inventory-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 2 \
    --inventory-configuration '{"Destination": { "S3BucketDestination":
 { "AccountId": "123456789012", "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
 "Format": "CSV" }}, "IsEnabled": true, "Id": "2", "IncludedObjectVersions":
 "Current", "Schedule": { "Frequency": "Daily" }}'
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketInventoryConfiguration。

put-bucket-lifecycle-configuration

以下程式碼範例顯示如何使用 put-bucket-lifecycle-configuration。

AWS CLI

下列命令會將生命週期組態套用至名為 的儲存貯體amzn-s3-demo-bucket：

```
aws s3api put-bucket-lifecycle-configuration --bucket amzn-s3-demo-bucket --
lifecycle-configuration  file://lifecycle.json
```

檔案lifecycle.json是目前資料夾中的 JSON 文件，可指定兩個規則：

```
{
    "Rules": [
        {
            "ID": "Move rotated logs to Glacier",
            "Prefix": "rotated/",
            "Status": "Enabled",
            "Transitions": [
                {
                    "Date": "2015-11-10T00:00:00.000Z",
                    "StorageClass": "GLACIER"
                }
            ]
        },
        {
            "Status": "Enabled",
            "Prefix": "",
            "NoncurrentVersionTransitions": [
                {
                    "NoncurrentDays": 2,
                    "StorageClass": "GLACIER"
                }
            ],
            "ID": "Move old versions to Glacier"
        }
    ]
}
```

第一個規則會在指定的日期rotated將字首為 的檔案移至 Glacier。當舊物件版本不再是最新版本時，第二個規則會將舊物件版本移至 Glacier。如需可接受時間戳記格式的資訊，請參閱《AWS CLI 使用者指南》中的指定參數值。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketLifecycleConfiguration。

put-bucket-lifecycle

以下程式碼範例顯示如何使用 put-bucket-lifecycle。

AWS CLI

下列命令會將生命週期組態套用至儲存貯體 amzn-s3-demo-bucket：

```
aws s3api put-bucket-lifecycle --bucket amzn-s3-demo-bucket --lifecycle-
configuration file://lifecycle.json
```

檔案lifecycle.json是目前資料夾中的 JSON 文件，可指定兩個規則：

```
{
  "Rules": [
    {
      "ID": "Move to Glacier after sixty days (objects in logs/2015/)",
      "Prefix": "logs/2015/",
      "Status": "Enabled",
      "Transition": {
        "Days": 60,
        "StorageClass": "GLACIER"
      }
    },
    {
      "Expiration": {
        "Date": "2016-01-01T00:00:00.000Z"
      },
      "ID": "Delete 2014 logs in 2016.",
      "Prefix": "logs/2014/",
      "Status": "Enabled"
    }
  ]
}
```

第一個規則會在 60 天後將檔案移至 Amazon Glacier。第二個規則會在指定的日期從 Amazon S3 刪除檔案。如需可接受時間戳記格式的資訊，請參閱《AWS CLI 使用者指南》中的指定參數值。

上述範例中的每個規則都會指定套用的政策 (Transition 或 Expiration) 和檔案字首 （資料夾名稱）。您也可以透過指定空白字首來建立套用至整個儲存貯體的規則：

```
{
  "Rules": [
    {
      "ID": "Move to Glacier after sixty days (all objects in bucket)",
      "Prefix": "",
      "Status": "Enabled",
      "Transition": {
        "Days": 60,
        "StorageClass": "GLACIER"
      }
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketLifecycle。

put-bucket-logging

以下程式碼範例顯示如何使用 put-bucket-logging。

AWS CLI

範例 1：設定儲存貯體政策記錄

下列put-bucket-logging範例會設定 amzn-s3-demo-bucket 的記錄政策。首先，使用 put-bucket-policy命令在儲存貯體政策中授予記錄服務主體許可。

```
aws s3api put-bucket-policy \
    --bucket amzn-s3-demo-bucket \
    --policy file://policy.json
```

policy.json 的內容：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3ServerAccessLogsPolicy",
            "Effect": "Allow",
            "Principal": {"Service": "logging.s3.amazonaws.com"},
            "Action": "s3:PutObject",
```

```
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/Logs/*",
            "Condition": {
                "ArnLike": {"aws:SourceARN": "arn:aws:s3:::SOURCE-BUCKET-NAME"},
                "StringEquals": {"aws:SourceAccount": "SOURCE-AWS-ACCOUNT-ID"}
            }
        }
    ]
}
```

若要套用記錄政策，請使用 put-bucket-logging。

```
aws s3api put-bucket-logging \
    --bucket amzn-s3-demo-bucket \
    --bucket-logging-status file://logging.json
```

logging.json 的內容：

```
{
     "LoggingEnabled": {
         "TargetBucket": "amzn-s3-demo-bucket",
         "TargetPrefix": "Logs/"
     }
 }
```

需要 put-bucket-policy命令才能將s3:PutObject許可授予記錄服務主體。

如需詳細資訊，請參閱《Amazon S3 使用者指南》中的 Amazon S3 伺服器存取記錄。 Amazon S3

範例 2：設定僅記錄單一使用者的存取的儲存貯體政策

下列put-bucket-logging範例會設定 amzn-s3-demo-bucket 的記錄政策。 AWS 使用者 bob@example.com 將擁有對日誌檔案的完全控制權，且沒有人具有任何存取權。首先，使用 授予 S3 許可put-bucket-acl。

```
aws s3api put-bucket-acl \
    --bucket amzn-s3-demo-bucket \
    --grant-write URI=http://acs.amazonaws.com/groups/s3/LogDelivery \
    --grant-read-acp URI=http://acs.amazonaws.com/groups/s3/LogDelivery
```

然後使用 套用記錄政策put-bucket-logging。

```
aws s3api put-bucket-logging \
    --bucket amzn-s3-demo-bucket \
    --bucket-logging-status file://logging.json
```

logging.json 的內容：

```
{
    "LoggingEnabled": {
        "TargetBucket": "amzn-s3-demo-bucket",
        "TargetPrefix": "amzn-s3-demo-bucket-logs/",
        "TargetGrants": [
            {
                "Grantee": {
                    "Type": "AmazonCustomerByEmail",
                    "EmailAddress": "bob@example.com"
                },
                "Permission": "FULL_CONTROL"
            }
        ]
    }
}
```

需要 put-bucket-acl命令才能授予 S3 的日誌交付系統必要的許可（寫入和讀取許可）。

如需詳細資訊，請參閱《Amazon S3 開發人員指南》中的 Amazon S3 伺服器存取記錄。 Amazon S3

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketLogging。

put-bucket-metrics-configuration

以下程式碼範例顯示如何使用 put-bucket-metrics-configuration。

AWS CLI

設定儲存貯體的指標組態

下列put-bucket-metrics-configuration範例會為指定的儲存貯體設定 ID 為 123 的指標組態。

```
aws s3api put-bucket-metrics-configuration \
```

```
    --bucket amzn-s3-demo-bucket \
    --id 123 \
    --metrics-configuration '{"Id": "123", "Filter": {"Prefix": "logs"}}'
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketMetricsConfiguration。

put-bucket-notification-configuration

以下程式碼範例顯示如何使用 put-bucket-notification-configuration。

AWS CLI

啟用儲存貯體的指定通知

下列put-bucket-notification-configuration範例會將通知組態套用至名為 的儲存貯體amzn-s3-demo-bucket。檔案notification.json是目前資料夾中的 JSON 文件，指定要監控的 SNS 主題和事件類型。

```
aws s3api put-bucket-notification-configuration \
    --bucket amzn-s3-demo-bucket \
    --notification-configuration file://notification.json
```

notification.json 的內容：

```
{
    "TopicConfigurations": [
        {
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic",
            "Events": [
                "s3:ObjectCreated:*"
            ]
        }
    ]
}
```

SNS 主題必須連接 IAM 政策，允許 Amazon S3 發佈至該政策。

```
{
```

```
    "Version": "2008-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "example-statement-ID",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "SNS:Publish"
            ],
            "Resource": "arn:aws:sns:us-west-2:123456789012::s3-notification-topic",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:*:*:amzn-s3-demo-bucket"
                }
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketNotificationConfiguration。

put-bucket-notification

以下程式碼範例顯示如何使用 put-bucket-notification。

AWS CLI

會將通知組態套用至名為 的儲存貯體amzn-s3-demo-bucket：

```
aws s3api put-bucket-notification --bucket amzn-s3-demo-bucket --notification-
configuration file://notification.json
```

檔案notification.json是目前資料夾中的 JSON 文件，指定要監控的 SNS 主題和事件類型：

```
{
  "TopicConfiguration": {
    "Event": "s3:ObjectCreated:*",
    "Topic": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic"
  }
```

```
}
```

SNS 主題必須連接 IAM 政策，允許 Amazon S3 發佈至該政策：

```
{
 "Version": "2008-10-17",
 "Id": "example-ID",
 "Statement": [
  {
   "Sid": "example-statement-ID",
   "Effect": "Allow",
   "Principal": {
     "Service": "s3.amazonaws.com"
   },
   "Action": [
    "SNS:Publish"
   ],
   "Resource": "arn:aws:sns:us-west-2:123456789012:amzn-s3-demo-bucket",
   "Condition": {
      "ArnLike": {
      "aws:SourceArn": "arn:aws:s3:*:*:amzn-s3-demo-bucket"
    }
   }
  }
 ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketNotification。

put-bucket-ownership-controls

以下程式碼範例顯示如何使用 put-bucket-ownership-controls。

AWS CLI

更新儲存貯體的儲存貯體擁有權設定

下列put-bucket-ownership-controls範例會更新儲存貯體的儲存貯體擁有權設定。

```
aws s3api put-bucket-ownership-controls \
    --bucket amzn-s3-demo-bucket \
    --ownership-controls="Rules=[{ObjectOwnership=BucketOwnerEnforced}]"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon S3 使用者指南》中的在現有儲存貯體上設定物件擁有權。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketOwnershipControls。

put-bucket-policy

以下程式碼範例顯示如何使用 put-bucket-policy。

AWS CLI

此範例允許所有使用者擷取 amzn-s3-demo-bucket 中的任何物件，MySecretFolder 中的物件除外。它也會將 put和 delete許可授予 AWS 帳戶 的根使用者1234-5678-9012：

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy file://policy.json

policy.json:
{
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": "*",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
      },
      {
         "Effect": "Deny",
         "Principal": "*",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/MySecretFolder/*"
      },
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::123456789012:root"
         },
         "Action": [
            "s3:DeleteObject",
            "s3:PutObject"
         ],
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
      }
```

```
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketPolicy。

put-bucket-replication

以下程式碼範例顯示如何使用 put-bucket-replication。

AWS CLI

設定 S3 儲存貯體的複寫

下列put-bucket-replication範例會將複寫組態套用至指定的 S3 儲存貯體。

```
aws s3api put-bucket-replication \
    --bucket amzn-s3-demo-bucket1 \
    --replication-configuration file://replication.json
```

replication.json 的內容：

```
{
    "Role": "arn:aws:iam::123456789012:role/s3-replication-role",
    "Rules": [
        {
            "Status": "Enabled",
            "Priority": 1,
            "DeleteMarkerReplication": { "Status": "Disabled" },
            "Filter" : { "Prefix": ""},
            "Destination": {
                "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket2"
            }
        }
    ]
}
```

目的地儲存貯體必須已啟用版本控制。指定的角色必須具有寫入目的地儲存貯體的許可，並具有允許 Amazon S3 擔任該角色的信任關係。

範例角色許可政策：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetReplicationConfiguration",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObjectVersion",
                "s3:GetObjectVersionAcl",
                "s3:GetObjectVersionTagging"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ReplicateObject",
                "s3:ReplicateDelete",
                "s3:ReplicateTags"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket2/*"
        }
    ]
}
```

信任關係政策範例：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Simple Storage Service 主控台使用者指南》中的這是主題標題。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketReplication。

put-bucket-request-payment

以下程式碼範例顯示如何使用 put-bucket-request-payment。

AWS CLI

範例 1：啟用儲存貯體的 ``requester pays`` 組態

下列put-bucket-request-payment範例requester pays會針對指定的儲存貯體啟用 。

```
aws s3api put-bucket-request-payment \
    --bucket amzn-s3-demo-bucket \
    --request-payment-configuration '{"Payer":"Requester"}'
```

此命令不會產生輸出。

範例 2：停用儲存貯體的 ``requester pays`` 組態

下列put-bucket-request-payment範例requester pays會針對指定的儲存貯體停用 。

```
aws s3api put-bucket-request-payment \
    --bucket amzn-s3-demo-bucket \
    --request-payment-configuration '{"Payer":"BucketOwner"}'
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketRequestPayment。

put-bucket-tagging

以下程式碼範例顯示如何使用 put-bucket-tagging。

AWS CLI

下列命令會將標記組態套用至名為 的儲存貯體amzn-s3-demo-bucket：

```
aws s3api put-bucket-tagging --bucket amzn-s3-demo-bucket --tagging file://
tagging.json
```

檔案tagging.json是目前資料夾中指定標籤的 JSON 文件：

```
{
   "TagSet": [
     {
       "Key": "organization",
       "Value": "marketing"
     }
   ]
}
```

或者amzn-s3-demo-bucket，直接從命令列將標記組態套用至 ：

```
aws s3api put-bucket-tagging --bucket amzn-s3-demo-bucket --tagging
 'TagSet=[{Key=organization,Value=marketing}]'
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketTagging。

put-bucket-versioning

以下程式碼範例顯示如何使用 put-bucket-versioning。

AWS CLI

下列命令會在名為 的儲存貯體上啟用版本控制amzn-s3-demo-bucket：

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket --versioning-
configuration Status=Enabled
```

下列命令會啟用版本控制，並使用 mfa 程式碼

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket --versioning-
configuration Status=Enabled --mfa "SERIAL 123456"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketVersioning。

put-bucket-website

以下程式碼範例顯示如何使用 put-bucket-website。

AWS CLI

會將靜態網站組態套用至名為 的儲存貯體amzn-s3-demo-bucket：

```
aws s3api put-bucket-website --bucket amzn-s3-demo-bucket --website-
configuration file://website.json
```

檔案website.json是目前資料夾中的 JSON 文件，可指定網站的索引和錯誤頁面：

```
{
    "IndexDocument": {
        "Suffix": "index.html"
    },
    "ErrorDocument": {
        "Key": "error.html"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutBucketWebsite。

put-object-acl

以下程式碼範例顯示如何使用 put-object-acl。

AWS CLI

下列命令full control會授予兩個 AWS 使用者 (user1@example.com 和 user2@example.com) 和read許可給每個人：

```
aws s3api put-object-acl --bucket amzn-s3-demo-bucket --key file.txt --grant-full-
control emailaddress=user1@example.com,emailaddress=user2@example.com --grant-
read uri=http://acs.amazonaws.com/groups/global/AllUsers
```

如需自訂 ACLs 的詳細資訊，請參閱 http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.htmlput-object-acl：//。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutObjectAcl。

put-object-legal-hold

以下程式碼範例顯示如何使用 put-object-legal-hold。

AWS CLI

將法務保存套用至物件

下列put-object-legal-hold範例會設定物件 的法務保存doc1.rtf。

```
aws s3api put-object-legal-hold \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --key doc1.rtf \
    --legal-hold Status=ON
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutObjectLegalHold。

put-object-lock-configuration

以下程式碼範例顯示如何使用 put-object-lock-configuration。

AWS CLI

在儲存貯體上設定物件鎖定組態

下列put-object-lock-configuration範例會在指定的儲存貯體上設定 50 天的物件鎖定。

```
aws s3api put-object-lock-configuration \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --object-lock-configuration '{ "ObjectLockEnabled": "Enabled", "Rule":
 { "DefaultRetention": { "Mode": "COMPLIANCE", "Days": 50 }}}'
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutObjectLockConfiguration。

put-object-retention

以下程式碼範例顯示如何使用 put-object-retention。

AWS CLI

設定物件的物件保留組態

下列put-object-retention範例會設定指定物件的物件保留組態，直到 2025-01-01。

```
aws s3api put-object-retention \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --key doc1.rtf \
    --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutObjectRetention。

put-object-tagging

以下程式碼範例顯示如何使用 put-object-tagging。

AWS CLI

在物件上設定標籤

下列put-object-tagging範例會在指定的物件confidential上設定具有 金鑰designation和 值的標籤。

```
aws s3api put-object-tagging \
    --bucket amzn-s3-demo-bucket \
    --key doc1.rtf \
    --tagging '{"TagSet": [{ "Key": "designation", "Value": "confidential" }]}'
```

此命令不會產生輸出。

下列put-object-tagging範例會在指定的物件上設定多個標籤集。

```
aws s3api put-object-tagging \
    --bucket amzn-s3-demo-bucket-example \
    --key doc3.rtf \
    --tagging '{"TagSet": [{ "Key": "designation", "Value": "confidential" },
 { "Key": "department", "Value": "finance" }, { "Key": "team", "Value":
 "payroll" } ]}'
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutObjectTagging。

put-object

以下程式碼範例顯示如何使用 put-object。

AWS CLI

範例 1：將物件上傳至 Amazon S3

下列put-object命令範例會將物件上傳至 Amazon S3。

```
aws s3api put-object \
    --bucket amzn-s3-demo-bucket \
    --key my-dir/MySampleImage.png \
    --body MySampleImage.png
```

如需上傳物件的詳細資訊，請參閱《Amazon S3 開發人員指南》中的上傳物件 < http://docs.aws.amazon.com/AmazonS3/latest/dev/UploadingObjects.html>。

範例 2：將影片檔案上傳至 Amazon S3

下列put-object命令範例會上傳影片檔案。

```
aws s3api put-object \
    --bucket amzn-s3-demo-bucket \
    --key my-dir/big-video-file.mp4 \
    --body /media/videos/f-sharp-3-data-services.mp4
```

如需上傳物件的詳細資訊，請參閱《Amazon S3 開發人員指南》中的上傳物件 < http://docs.aws.amazon.com/AmazonS3/latest/dev/UploadingObjects.html>。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutObject。

put-public-access-block

以下程式碼範例顯示如何使用 put-public-access-block。

AWS CLI

設定儲存貯體的封鎖公開存取組態

下列put-public-access-block範例會設定指定儲存貯體的限制性封鎖公開存取組態。

```
aws s3api put-public-access-block \
    --bucket amzn-s3-demo-bucket \
    --public-access-block-
configuration "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPub
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutPublicAccessBlock。

rb

以下程式碼範例顯示如何使用 rb。

AWS CLI

範例 1：刪除儲存貯體

下列rb命令會移除儲存貯體。在此範例中，使用者的儲存貯體為 amzn-s3-demo-bucket。請注意，儲存貯體必須為空，才能移除：

```
aws s3 rb s3://amzn-s3-demo-bucket
```

輸出：

```
remove_bucket: amzn-s3-demo-bucket
```

範例 2：強制刪除儲存貯體

下列rb命令會使用 --force 參數，先移除儲存貯體中的所有物件，然後移除儲存貯體本身。在此範例中，使用者的儲存貯體是 amzn-s3-demo-bucket，而 中的物件amzn-s3-demo-bucket是 test1.txt和 test2.txt：

```
aws s3 rb s3://amzn-s3-demo-bucket \
    --force
```

輸出：

```
delete: s3://amzn-s3-demo-bucket/test1.txt
delete: s3://amzn-s3-demo-bucket/test2.txt
remove_bucket: amzn-s3-demo-bucket
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 Rb。

`restore-object`

以下程式碼範例顯示如何使用 `restore-object`。

AWS CLI

建立物件的還原請求

下列`restore-object`範例會還原儲存貯體的指定 Amazon S3 Glacier 物件 `my-glacier-bucket` 10 天。

```
aws s3api restore-object \
    --bucket my-glacier-bucket \
    --key doc1.rtf \
    --restore-request Days=10
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreObject。

`rm`

以下程式碼範例顯示如何使用 `rm`。

AWS CLI

範例 1：刪除 S3 物件

下列`rm`命令會刪除單一 s3 物件：

```
aws s3 rm s3://amzn-s3-demo-bucket/test2.txt
```

輸出：

```
delete: s3://amzn-s3-demo-bucket/test2.txt
```

範例 2：刪除儲存貯體中的所有內容

使用 參數 傳遞時，下列rm命令會遞迴刪除指定儲存貯體和字首下的所有物件--recursive。在此範例中，儲存貯體amzn-s3-demo-bucket包含物件 test1.txt和 test2.txt：

```
aws s3 rm s3://amzn-s3-demo-bucket \
    --recursive
```

輸出：

```
delete: s3://amzn-s3-demo-bucket/test1.txt
delete: s3://amzn-s3-demo-bucket/test2.txt
```

範例 3：刪除儲存貯體中的所有內容，但 ``.jpg`` 檔案除外

使用 參數排除--recursive某些物件時，下列rm命令會遞迴刪除指定儲存貯體和字首下的所有物件--exclude。在此範例中，儲存貯體amzn-s3-demo-bucket具有 物件 test1.txt和 test2.jpg：

```
aws s3 rm s3://amzn-s3-demo-bucket/ \
    --recursive \
    --exclude "*.jpg"
```

輸出：

```
delete: s3://amzn-s3-demo-bucket/test1.txt
```

範例 4：刪除儲存貯體中的所有內容，指定字首下的物件除外

當使用 參數傳遞時，下列rm命令會遞迴刪除指定儲存貯體和字首下的所有物件，--recursive同時使用 --exclude 參數排除特定字首下的所有物件。在此範例中，儲存貯體amzn-s3-demo-bucket具有 物件 test1.txt和 another/test.txt：

```
aws s3 rm s3://amzn-s3-demo-bucket/ \
    --recursive \
    --exclude "another/*"
```

輸出：

```
delete: s3://amzn-s3-demo-bucket/test1.txt
```

範例 5：從 S3 存取點刪除物件

下列rm命令會從存取點 (mykey) 刪除單一物件 ()myaccesspoint。：：下列rm命令會從存取點 (mykey) 刪除單一物件 ()myaccesspoint。

```
aws s3 rm s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey
```

輸出：

```
delete: s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 Rm。

select-object-content

以下程式碼範例顯示如何使用 select-object-content。

AWS CLI

根據 SQL 陳述式篩選 Amazon S3 物件的內容

下列select-object-content範例會使用指定的 SQL 陳述my-data-file.csv式篩選物件，並將輸出傳送至檔案。

```
aws s3api select-object-content \
    --bucket amzn-s3-demo-bucket \
    --key my-data-file.csv \
    --expression "select * from s3object limit 100" \
    --expression-type 'SQL' \
    --input-serialization '{"CSV": {}, "CompressionType": "NONE"}' \
    --output-serialization '{"CSV": {}}' "output.csv"
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SelectObjectContent。

sync

以下程式碼範例顯示如何使用 sync。

AWS CLI

範例 1：將所有本機物件同步至指定的儲存貯體

下列sync命令會將本機檔案上傳至 S3，以將物件從本機目錄同步至指定的字首和儲存貯體。如果本機檔案的大小與 S3 物件的大小不同、本機檔案的上次修改時間比 S3 物件的上次修改時間更新，或本機檔案不存在於指定的儲存貯體和字首下，則需要上傳本機檔案。在此範例中，使用者會將儲存貯體同步amzn-s3-demo-bucket至本機目前的目錄。本機目前目錄包含 檔案test.txt和test2.txt。儲存貯amzn-s3-demo-bucket體不包含物件。

```
aws s3 sync . s3://amzn-s3-demo-bucket
```

輸出：

```
upload: test.txt to s3://amzn-s3-demo-bucket/test.txt
upload: test2.txt to s3://amzn-s3-demo-bucket/test2.txt
```

範例 2：將所有 S3 物件從指定的 S3 儲存貯體同步到另一個儲存貯體

下列sync命令會透過複製 S3 物件，將指定字首和儲存貯體下的物件同步至另一個指定字首和儲存貯體下的物件。如果兩個 S3S3 物件的大小不同、來源的上次修改時間比目的地的上次修改時間更新，或 S3 物件不存在於指定的儲存貯體和字首目的地下方，則 S3 物件將需要複製。

在此範例中，使用者會將儲存貯體同步amzn-s3-demo-bucket至儲存貯體 amzn-s3-demo-bucket2。儲存貯體amzn-s3-demo-bucket包含物件 test.txt和 test2.txt。儲存貯amzn-s3-demo-bucket2體不包含物件：

```
aws s3 sync s3://amzn-s3-demo-bucket s3://amzn-s3-demo-bucket2
```

輸出：

```
copy: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket2/test.txt
```

```
copy: s3://amzn-s3-demo-bucket/test2.txt to s3://amzn-s3-demo-bucket2/test2.txt
```

範例 3：將所有 S3 物件從指定的 S3 儲存貯體同步至本機目錄

下列sync命令會下載 S3 物件，將檔案從指定的 S3 儲存貯體同步至本機目錄。如果 S3S3 物件的大小與本機檔案大小不同、S3 物件的上次修改時間比本機檔案的上次修改時間更新，或本機目錄中不存在 S3 物件，則 S3 物件將需要下載。請注意，從 S3 下載物件時，本機檔案的上次修改時間會變更為 S3 物件的上次修改時間。在此範例中，使用者會將儲存貯體同步amzn-s3-demo-bucket至目前的本機目錄。儲存貯體amzn-s3-demo-bucket包含物件 test.txt和 test2.txt。目前的本機目錄沒有檔案：

```
aws s3 sync s3://amzn-s3-demo-bucket .
```

輸出：

```
download: s3://amzn-s3-demo-bucket/test.txt to test.txt
download: s3://amzn-s3-demo-bucket/test2.txt to test2.txt
```

範例 4：將所有本機物件同步至指定的儲存貯體，並刪除不相符的所有檔案

下列sync命令會將指定字首和儲存貯體下的物件同步至本機目錄中的檔案，方法是將本機檔案上傳至 S3。由於 --delete 參數，在指定字首和儲存貯體下存在但本機目錄中不存在的任何檔案都會遭到刪除。在此範例中，使用者會將儲存貯體同步amzn-s3-demo-bucket至本機目前的目錄。本機目前目錄包含 檔案test.txt和 test2.txt。儲存貯體amzn-s3-demo-bucket包含物件 test3.txt：

```
aws s3 sync . s3://amzn-s3-demo-bucket \
    --delete
```

輸出：

```
upload: test.txt to s3://amzn-s3-demo-bucket/test.txt
upload: test2.txt to s3://amzn-s3-demo-bucket/test2.txt
delete: s3://amzn-s3-demo-bucket/test3.txt
```

範例 5：將所有本機物件同步至指定的儲存貯體，但 ``.jpg`` 檔案除外

下列sync命令會將指定字首和儲存貯體下的物件同步至本機目錄中的檔案，方法是將本機檔案上傳至 S3。由於 --exclude 參數，符合 S3 和本機中現有模式的所有檔案都會從同步中排除。在此範

例中，使用者會將儲存貯體同步amzn-s3-demo-bucket至本機目前的目錄。本機目前目錄包含檔案test.jpg和 test2.txt。儲存貯體amzn-s3-demo-bucket包含與本機 test.jpg大小不同的物件test.jpg：

```
aws s3 sync . s3://amzn-s3-demo-bucket \
    --exclude "*.jpg"
```

輸出：

```
upload: test2.txt to s3://amzn-s3-demo-bucket/test2.txt
```

範例 6：將所有本機物件同步至指定的儲存貯體，但指定的目錄檔案除外

下列sync命令會透過下載 S3 物件，將本機目錄下的檔案同步至指定字首和儲存貯體下的物件。此範例使用 --exclude 參數旗標，從 sync命令中排除指定的目錄和 S3 字首。在此範例中，使用者會將本機目前目錄同步至儲存貯體 amzn-s3-demo-bucket。本機目前目錄包含 檔案test.txt和 another/test2.txt。儲存貯體amzn-s3-demo-bucket包含物件 another/test5.txt和 test1.txt：

```
aws s3 sync s3://amzn-s3-demo-bucket/ . \
    --exclude "*another/*"
```

輸出：

```
download: s3://amzn-s3-demo-bucket/test1.txt to test1.txt
```

範例 7：同步不同區域中儲存貯體之間的所有物件

下列sync命令會在不同區域的兩個儲存貯體之間同步檔案：

```
aws s3 sync s3://my-us-west-2-bucket s3://my-us-east-1-bucket \
    --source-region us-west-2 \
    --region us-east-1
```

輸出：

```
download: s3://my-us-west-2-bucket/test1.txt to s3://my-us-east-1-bucket/test1.txt
```

範例 8：同步至 S3 存取點

下列sync命令會將目前的目錄同步至存取點 (myaccesspoint)：

```
aws s3 sync . s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
```

輸出：

```
upload: test.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/test.txt
upload: test2.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/test2.txt
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的同步。

upload-part-copy

以下程式碼範例顯示如何使用 upload-part-copy。

AWS CLI

將現有物件中的資料複製為資料來源，以上傳部分物件

下列upload-part-copy範例會將現有物件中的資料複製為資料來源，以上傳組件。

```
aws s3api upload-part-copy \
    --bucket amzn-s3-demo-bucket \
    --key "Map_Data_June.mp4" \
    --copy-source "amzn-s3-demo-bucket/copy_of_Map_Data_June.mp4" \
    --part-number 1 \
    --upload-
id "bq0tdE1CDpWQYRPLHuNG50xAT6pA5D.m_RiBy0ggOH6b13pVRY7QjvLlf75iFdJqp_2wztk5hvpUM2SesXgrzbeh
```

輸出：

```
{
    "CopyPartResult": {
        "LastModified": "2019-12-13T23:16:03.000Z",
        "ETag": "\"711470fc377698c393d94aed6305e245\""
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UploadPartCopy。

upload-part

以下程式碼範例顯示如何使用 upload-part。

AWS CLI

下列命令會上傳使用 create-multipart-upload命令啟動的分段上傳中的第一個部分：

```
aws s3api upload-part --bucket amzn-s3-demo-bucket --key
 'multipart/01' --part-number 1 --body part01 --upload-id
  "dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

body 選項會取得本機檔案的名稱或路徑以進行上傳（請勿使用 檔案：// 字首）。最小部分大小為 5 MB。上傳 ID 由 傳回，create-multipart-upload也可以使用 擷取list-multipart-uploads。當您建立分段上傳時，會指定儲存貯體和金鑰。

輸出：

```
{
    "ETag": "\"e868e0f4719e394144ef36531ee6824c\""
}
```

儲存每個部分的 ETag 值以供稍後使用。它們需要完成分段上傳。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UploadPart。

website

以下程式碼範例顯示如何使用 website。

AWS CLI

將 S3 儲存貯體設定為靜態網站

下列命令會將名為 的儲存貯amzn-s3-demo-bucket體設定為靜態網站。索引文件選項指定amzn-s3-demo-bucket訪客在導覽至網站 URL 時將導向至其中的檔案。在此情況下，儲存貯體位於 us-west-2 區域，因此網站會出現在 http://amzn-s3-demo-bucket.s3-website-us-west-2.amazonaws.com。

出現在靜態網站上之儲存貯體中的所有檔案必須設定為允許訪客開啟。檔案許可會與儲存貯體網站組態分開設定。

```
aws s3 website s3://amzn-s3-demo-bucket/ \
    --index-document index.html \
    --error-document error.html
```

如需在 Amazon S3 中託管靜態網站的資訊，請參閱《Amazon Simple Storage Service 開發人員指南》中的託管靜態網站。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的網站。

使用 的 Amazon S3 控制範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon S3 Control 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-access-point`

以下程式碼範例顯示如何使用 `create-access-point`。

AWS CLI

建立存取點

下列`create-access-point`範例`finance-ap`會為帳戶 123456789012 `business-records`中的儲存貯體建立名為 的存取點。執行此範例之前，請將存取點名稱、儲存貯體名稱和帳戶號碼取代為您的使用案例的適當值。

```
aws s3control create-access-point \
```

```
    --account-id 123456789012 \
    --bucket business-records \
    --name finance-ap
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Simple Storage Service 開發人員指南》中的建立存取點。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAccessPoint。

create-job

以下程式碼範例顯示如何使用 create-job。

AWS CLI

建立 Amazon S3 批次操作任務

下列create-job範例會建立 Amazon S3 批次操作任務，將物件標記為 confidential` in the bucket ``employee-records。

```
aws s3control create-job \
    --account-id 123456789012 \
    --operation '{"S3PutObjectTagging": { "TagSet": [{"Key":"confidential",
 "Value":"true"}] }}' \
    --report '{"Bucket":"arn:aws:s3:::employee-records-logs","Prefix":"batch-op-
create-job",
 "Format":"Report_CSV_20180820","Enabled":true,"ReportScope":"AllTasks"}' \
    --manifest '{"Spec":{"Format":"S3BatchOperations_CSV_20180820","Fields":
["Bucket","Key"]},"Location":{"ObjectArn":"arn:aws:s3:::employee-records-logs/inv-
report/7a6a9be4-072c-407e-85a2-
ec3e982f773e.csv","ETag":"69f52a4e9f797e987155d9c8f5880897"}}' \
    --priority 42 \
    --role-arn arn:aws:iam::123456789012:role/S3BatchJobRole
```

輸出：

```
{
    "JobId": "93735294-df46-44d5-8638-6356f335324e"
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateJob。

delete-access-point-policy

以下程式碼範例顯示如何使用 delete-access-point-policy。

AWS CLI

刪除存取點政策

下列delete-access-point-policy範例會從帳戶 123456789012 finance-ap中名為 的存取點刪除存取點政策。執行此範例之前，請將存取點名稱和帳戶號碼取代為您的使用案例的適當值。

```
aws s3control delete-access-point-policy \
    --account-id 123456789012 \
    --name finance-ap
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Simple Storage Service 開發人員指南》中的使用 Amazon S3 存取點管理資料存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAccessPointPolicy。

delete-access-point

以下程式碼範例顯示如何使用 delete-access-point。

AWS CLI

刪除存取點

下列delete-access-point範例會刪除帳戶 123456789012 finance-ap中名為 的存取點。執行此範例之前，請將存取點名稱和帳戶號碼取代為您的使用案例的適當值。

```
aws s3control delete-access-point \
    --account-id 123456789012 \
    --name finance-ap
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Simple Storage Service 開發人員指南》中的使用 Amazon S3 存取點管理資料存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAccessPoint。

delete-public-access-block

以下程式碼範例顯示如何使用 delete-public-access-block。

AWS CLI

刪除 帳戶的封鎖公開存取設定

下列delete-public-access-block範例會刪除指定帳戶的封鎖公開存取設定。

```
aws s3control delete-public-access-block \
    --account-id 123456789012
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePublicAccessBlock。

describe-job

以下程式碼範例顯示如何使用 describe-job。

AWS CLI

描述 Amazon S3 批次操作任務

以下describe-job提供指定批次操作任務的組態參數和狀態。

```
aws s3control describe-job \
    --account-id 123456789012 \
    --job-id 93735294-df46-44d5-8638-6356f335324e
```

輸出：

```
{
    "Job": {
        "TerminationDate": "2019-10-03T21:49:53.944Z",
        "JobId": "93735294-df46-44d5-8638-6356f335324e",
        "FailureReasons": [],
        "Manifest": {
            "Spec": {
```

```
                "Fields": [
                    "Bucket",
                    "Key"
                ],
                "Format": "S3BatchOperations_CSV_20180820"
            },
            "Location": {
                "ETag": "69f52a4e9f797e987155d9c8f5880897",
                "ObjectArn": "arn:aws:s3:::employee-records-logs/inv-
report/7a6a9be4-072c-407e-85a2-ec3e982f773e.csv"
            }
        },
        "Operation": {
            "S3PutObjectTagging": {
                "TagSet": [
                    {
                        "Value": "true",
                        "Key": "confidential"
                    }
                ]
            }
        },
        "RoleArn": "arn:aws:iam::123456789012:role/S3BatchJobRole",
        "ProgressSummary": {
            "TotalNumberOfTasks": 8,
            "NumberOfTasksFailed": 0,
            "NumberOfTasksSucceeded": 8
        },
        "Priority": 42,
        "Report": {
            "ReportScope": "AllTasks",
            "Format": "Report_CSV_20180820",
            "Enabled": true,
            "Prefix": "batch-op-create-job",
            "Bucket": "arn:aws:s3:::employee-records-logs"
        },
        "JobArn": "arn:aws:s3:us-west-2:123456789012:job/93735294-
df46-44d5-8638-6356f335324e",
        "CreationTime": "2019-10-03T21:48:48.048Z",
        "Status": "Complete"
    }
}
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeVpcs。

get-access-point-policy-status

以下程式碼範例顯示如何使用 get-access-point-policy-status。

AWS CLI

擷取存取點政策狀態

下列get-access-point-policy-status範例會擷取帳戶 123456789012 finance-ap中名為之存取點的存取點政策狀態。存取點政策狀態指出存取點的政策是否允許公開存取。執行此範例之前，請將存取點名稱和帳戶號碼取代為您的使用案例的適當值。

```
aws s3control get-access-point-policy-status \
    --account-id 123456789012 \
    --name finance-ap
```

輸出：

```
{
    "PolicyStatus": {
        "IsPublic": false
    }
}
```

如需存取點政策何時視為公有的詳細資訊，請參閱《Amazon Simple Storage Service 開發人員指南》中的「公有」意義。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAccessPointPolicyStatus。

get-access-point-policy

以下程式碼範例顯示如何使用 get-access-point-policy。

AWS CLI

擷取存取點政策

下列get-access-point-policy範例會從帳戶 123456789012 finance-ap中名為 的存取點擷取存取點政策。執行此範例之前，請將存取點名稱和帳戶號碼取代為您的使用案例的適當值。

```
aws s3control get-access-point-policy \
    --account-id 123456789012 \
```

```
    --name finance-ap
```

輸出：

```
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:role/Admin\"},\"Action\":
\"s3:GetObject\",\"Resource\":\"arn:aws:s3:us-west-2:123456789012:accesspoint/
finance-ap/object/records/*\"}]}"
}
```

如需詳細資訊，請參閱《Amazon Simple Storage Service 開發人員指南》中的使用 Amazon S3 存取點管理資料存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAccessPointPolicy。

get-access-point

以下程式碼範例顯示如何使用 get-access-point。

AWS CLI

擷取存取點組態詳細資訊

下列get-access-point範例會擷取帳戶 123456789012 finance-ap中名為 之存取點的組態詳細資訊。執行此範例之前，請將存取點名稱和帳戶號碼取代為您的使用案例的適當值。

```
aws s3control get-access-point \
    --account-id 123456789012 \
    --name finance-ap
```

輸出：

```
{
    "Name": "finance-ap",
    "Bucket": "business-records",
    "NetworkOrigin": "Internet",
    "PublicAccessBlockConfiguration": {
        "BlockPublicAcls": false,
        "IgnorePublicAcls": false,
        "BlockPublicPolicy": false,
        "RestrictPublicBuckets": false
```

```
    },
    "CreationDate": "2020-01-01T00:00:00Z"
}
```

如需詳細資訊，請參閱《Amazon Simple Storage Service 開發人員指南》中的使用 Amazon S3 存取點管理資料存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAccessPoint。

get-multi-region-access-point-routes

以下程式碼範例顯示如何使用 get-multi-region-access-point-routes。

AWS CLI

查詢目前的多區域存取點路由組態

下列get-multi-region-access-point-routes範例會傳回指定多區域存取點的目前路由組態。

```
aws s3control get-multi-region-access-point-routes \
    --region Region \
    --account-id 111122223333 \
    --mrap MultiRegionAccessPoint_ARN
```

輸出：

```
{
    "Mrap": "arn:aws:s3::111122223333:accesspoint/0000000000000.mrap",
    "Routes": [
        {
            "Bucket": "amzn-s3-demo-bucket1",
            "Region": "ap-southeast-2",
            "TrafficDialPercentage": 100
        },
        {
            "Bucket": "amzn-s3-demo-bucket2",
            "Region": "us-west-1",
            "TrafficDialPercentage": 0
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMultiRegionAccessPointRoutes。

get-public-access-block

以下程式碼範例顯示如何使用 get-public-access-block。

AWS CLI

列出帳戶的公有封鎖存取設定

下列get-public-access-block範例顯示指定帳戶的封鎖公開存取設定。

```
aws s3control get-public-access-block \
    --account-id 123456789012
```

輸出：

```
{
   "PublicAccessBlockConfiguration": {
      "BlockPublicPolicy": true,
      "RestrictPublicBuckets": true,
      "IgnorePublicAcls": true,
      "BlockPublicAcls": true
   }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPublicAccessBlock。

list-access-points

以下程式碼範例顯示如何使用 list-access-points。

AWS CLI

範例 1：擷取帳戶的所有存取點清單

下列list-access-points範例顯示連接至帳戶 123456789012 擁有之儲存貯體的所有存取點清單。

```
aws s3control list-access-points \
```

```
    --account-id 123456789012
```

輸出：

```
{
    "AccessPointList": [
        {
            "Name": "finance-ap",
            "NetworkOrigin": "Internet",
            "Bucket": "business-records"
        },
        {
            "Name": "managers-ap",
            "NetworkOrigin": "Internet",
            "Bucket": "business-records"
        },
        {
            "Name": "private-network-ap",
            "NetworkOrigin": "VPC",
            "VpcConfiguration": {
                "VpcId": "1a2b3c"
            },
            "Bucket": "business-records"
        },
        {
            "Name": "customer-ap",
            "NetworkOrigin": "Internet",
            "Bucket": "external-docs"
        },
        {
            "Name": "public-ap",
            "NetworkOrigin": "Internet",
            "Bucket": "external-docs"
        }
    ]
}
```

範例 2：擷取儲存貯體的所有存取點清單

下列`list-access-points`範例會擷取連接至帳戶 123456789012 所`external-docs`擁有儲存貯體的所有存取點清單。

```
aws s3control list-access-points \
```

```
    --account-id 123456789012 \
    --bucket external-docs
```

輸出：

```
{
    "AccessPointList": [
        {
            "Name": "customer-ap",
            "NetworkOrigin": "Internet",
            "Bucket": "external-docs"
        },
        {
            "Name": "public-ap",
            "NetworkOrigin": "Internet",
            "Bucket": "external-docs"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Simple Storage Service 開發人員指南》中的使用 Amazon S3 存取點管理資料存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAccessPoints。

list-jobs

以下程式碼範例顯示如何使用 `list-jobs`。

AWS CLI

列出帳戶 Amazon S3 批次操作任務

下列`list-jobs`範例列出指定帳戶的所有最近批次操作任務。

```
aws s3control list-jobs \
    --account-id 123456789012
```

輸出：

```
{
```

```
    "Jobs": [
        {
            "Operation": "S3PutObjectTagging",
            "ProgressSummary": {
                "NumberOfTasksFailed": 0,
                "NumberOfTasksSucceeded": 8,
                "TotalNumberOfTasks": 8
            },
            "CreationTime": "2019-10-03T21:48:48.048Z",
            "Status": "Complete",
            "JobId": "93735294-df46-44d5-8638-6356f335324e",
            "Priority": 42
        },
        {
            "Operation": "S3PutObjectTagging",
            "ProgressSummary": {
                "NumberOfTasksFailed": 0,
                "NumberOfTasksSucceeded": 0,
                "TotalNumberOfTasks": 0
            },
            "CreationTime": "2019-10-03T21:46:07.084Z",
            "Status": "Failed",
            "JobId": "3f3c7619-02d3-4779-97f6-1d98dd313108",
            "Priority": 42
        },
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListJobs。

put-access-point-policy

以下程式碼範例顯示如何使用 put-access-point-policy。

AWS CLI

設定存取點政策

下列put-access-point-policy範例會將存取點的指定存取點政策放在帳戶 123456789012 finance-ap中。如果存取點finance-ap已有政策，此命令會將現有的政策取代為此命令中指定的政策。執行此範例之前，請將帳戶號碼、存取點名稱和政策陳述式取代為您的使用案例的適當值。

```
aws s3control put-access-point-policy \
    --account-id 123456789012 \
    --name finance-ap \
    --policy file://ap-policy.json
```

ap-policy.json 的內容：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Alice"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/finance-ap/
object/Alice/*"
        }
    ]
}
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Simple Storage Service 開發人員指南》中的使用 Amazon S3 存取點管理資料存取。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutAccessPointPolicy。

put-public-access-block

以下程式碼範例顯示如何使用 put-public-access-block。

AWS CLI

編輯 帳戶的封鎖公開存取設定

下列put-public-access-block範例會將指定帳戶的所有封鎖公開存取設定切換true為 。

```
aws s3control put-public-access-block \
    --account-id 123456789012 \
```

```
    --public-access-block-configuration '{"BlockPublicAcls": true,
"IgnorePublicAcls": true, "BlockPublicPolicy": true, "RestrictPublicBuckets":
true}'
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutPublicAccessBlock。

submit-multi-region-access-point-routes

以下程式碼範例顯示如何使用 submit-multi-region-access-point-routes。

AWS CLI

更新您的多區域存取點路由組態

下列submit-multi-region-access-point-routes範例會更新多區域存取點 amzn-s3-demo-bucket2 ap-southeast-2 區域中 amzn-s3-demo-bucket1和 的路由狀態。

```
aws s3control submit-multi-region-access-point-routes \
    --region ap-southeast-2 \
    --account-id 111122223333 \
    --mrap MultiRegionAccessPoint_ARN \
    --route-updates Bucket=amzn-s3-demo-
bucket1,TrafficDialPercentage=100 Bucket=amzn-s3-demo-
bucket2,TrafficDialPercentage=0
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SubmitMultiRegionAccessPointRoutes。

update-job-priority

以下程式碼範例顯示如何使用 update-job-priority。

AWS CLI

更新 Amazon S3 批次操作任務的任務優先順序

下列update-job-priority範例會將指定的任務更新為新的優先順序。

```
aws s3control update-job-priority \
```

```
    --account-id 123456789012 \
    --job-id 8d9a18fe-c303-4d39-8ccc-860d372da386 \
    --priority 52
```

輸出：

```
{
    "JobId": "8d9a18fe-c303-4d39-8ccc-860d372da386",
    "Priority": 52
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateJobPriority。

update-job-status

以下程式碼範例顯示如何使用 update-job-status。

AWS CLI

更新 Amazon S3 批次操作任務的狀態

下列update-job-status範例會取消正在等待核准的指定任務。

```
aws s3control update-job-status \
    --account-id 123456789012 \
    --job-id 8d9a18fe-c303-4d39-8ccc-860d372da386 \
    --requested-job-status Cancelled
```

輸出：

```
{
    "Status": "Cancelled",
    "JobId": "8d9a18fe-c303-4d39-8ccc-860d372da386"
}
```

下列update-job-status範例會確認並執行正在等待核准的指定 。

```
aws s3control update-job-status \
    --account-id 123456789012 \
    --job-id 5782949f-3301-4fb3-be34-8d5bab54dbca \
    --requested-job-status Ready
```

```
Output::

{
    "Status": "Ready",
    "JobId": "5782949f-3301-4fb3-
be34-8d5bab54dbca"
}
```

下列update-job-status範例會取消正在執行的指定任務。

```
 aws s3control update-job-status \
    --account-id 123456789012 \
    --job-id 5782949f-3301-4fb3-be34-8d5bab54dbca \
    --requested-job-status Cancelled

Output::
{
         "Status": "Cancelling",
         "JobId": "5782949f-3301-4fb3-be34-8d5bab54dbca"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateJobStatus。

使用 的 S3 Glacier 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 S3 Glacier 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`abort-multipart-upload`

以下程式碼範例顯示如何使用 abort-multipart-upload。

AWS CLI

下列命令會刪除對名為 的保存庫進行中的分段上傳my-vault：

```
aws glacier abort-multipart-upload --account-id - --vault-name my-vault
 --upload-id 19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ
```

此命令不會產生任何輸出。Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。上傳 ID 由 `aws glacier initiate-multipart-upload` 命令傳回，也可以透過 `aws glacier list-multipart-uploads` 獲取。

如需使用 CLI 將分段上傳至 Amazon Glacier AWS 的詳細資訊，請參閱《AWS CLI 使用者指南》中的使用 Amazon Glacier。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AbortMultipartUpload。

abort-vault-lock

以下程式碼範例顯示如何使用 `abort-vault-lock`。

AWS CLI

中止進行中保存庫鎖定程序

下列`abort-vault-lock`範例會從指定的保存庫中刪除保存庫鎖定政策，並將保存庫鎖定的鎖定狀態重設為解除鎖定。

```
aws glacier abort-vault-lock \
    --account-id - \
    --vault-name MyVaultName
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Glacier API 開發人員指南》中的中止保存庫鎖定 (DELETE lock-policy)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AbortVaultLock。

add-tags-to-vault

以下程式碼範例顯示如何使用 `add-tags-to-vault`。

AWS CLI

以下命令會將兩個標籤加入名為 my-vault 的文件庫：

```
aws glacier add-tags-to-vault --account-id - --vault-name my-vault --
tags id=1234,date=july2015
```

Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AddTagsToVault。

complete-multipart-upload

以下程式碼範例顯示如何使用 complete-multipart-upload。

AWS CLI

下列命令會完成 3 MiB 封存的分段上傳：

```
aws glacier complete-multipart-upload --archive-size 3145728 --
checksum 9628195fcdbcbbe76cdde456d4646fa7de5f219fb39823836d81f0cc0e18aa67
 --upload-id 19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ --account-id - --vault-name my-vault
```

Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

上傳 ID 由 aws glacier initiate-multipart-upload 命令傳回，也可以透過 aws glacier list-multipart-uploads 獲取。檢查總和參數採用十六進位封存的 SHA-256 樹雜湊。

如需使用 CLI 分段上傳至 Amazon Glacier AWS 的詳細資訊，包括計算樹雜湊的指示，請參閱《AWS CLI 使用者指南》中的使用 Amazon Glacier。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CompleteMultipartUpload。

complete-vault-lock

以下程式碼範例顯示如何使用 complete-vault-lock。

AWS CLI

完成進行中保存庫鎖定程序

下列complete-vault-lock範例會完成指定保存庫的進行中鎖定進度，並將保存庫鎖定的鎖定狀態設定為 Locked。執行 時，您會取得 lock-id 參數的值initiate-lock-process。

```
aws glacier complete-vault-lock \
    --account-id - \
    --vault-name MyVaultName \
    --lock-id 9QZgEXAMPLEPhvL6xEXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Glacier API 開發人員指南》中的完成保存庫鎖定 (POST lockId)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CompleteVaultLock。

create-vault

以下程式碼範例顯示如何使用 create-vault。

AWS CLI

以下命令建立一個名為 my-vault 的文件庫：

```
aws glacier create-vault --vault-name my-vault --account-id -
```

Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateVault。

delete-archive

以下程式碼範例顯示如何使用 delete-archive。

AWS CLI

從文件庫刪除封存

下列 delete-archive 範例會從 example_vault 中移除指定的封存。

```
aws glacier delete-archive \
    --account-id 111122223333  \
```

```
    --vault-name example_vault \
    --archive-id Sc0u9ZP8yaWkmh-XGlIvAVprtLhaLCGnNwNl5I5x9HqPIkX5mjc0DrId3Ln-
Gi_k2HzmlIDZUz117KSdVMdMXLuFWi9PJUitxWO73edQ43eTlMWkH0pd9zVSAuV_XXZBVhKhyGhJ7w
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteArchive。

delete-vault-access-policy

以下程式碼範例顯示如何使用 delete-vault-access-policy。

AWS CLI

移除保存庫的存取政策

下列delete-vault-access-policy範例會移除指定保存庫的存取政策。

```
aws glacier delete-vault-access-policy \
    --account-id 111122223333 \
    --vault-name example_vault
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVaultAccessPolicy。

delete-vault-notifications

以下程式碼範例顯示如何使用 delete-vault-notifications。

AWS CLI

若要移除文件庫的 SNS 通知

下列 delete-vault-notifications 範例會在指定的文件庫移除 Amazon Simple Notification Service (Amazon SNS) 傳送之通知。

```
aws glacier delete-vault-notifications \
    --account-id 111122223333 \
    --vault-name example_vault
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteVaultNotifications。

delete-vault

以下程式碼範例顯示如何使用 delete-vault。

AWS CLI

以下命令刪除一個名為 my-vault 的文件庫：

```
aws glacier delete-vault --vault-name my-vault --account-id -
```

此命令不會產生任何輸出。Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteVault。

describe-job

以下程式碼範例顯示如何使用 describe-job。

AWS CLI

以下命令會擷取文件庫上名為 my-vault 的庫存擷取工作相關資訊：

```
aws glacier describe-job --account-id - --vault-name my-
vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW
```

輸出：

```
{
    "InventoryRetrievalParameters": {
        "Format": "JSON"
    },
    "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
    "Completed": false,
    "JobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW",
    "Action": "InventoryRetrieval",
    "CreationDate": "2015-07-17T20:23:41.616Z",
```

```
    "StatusCode": "InProgress"
}
```

工作 ID 可以在 `aws glacier initiate-job` 和 `aws glacier list-jobs` 的輸出中找到。Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeVpcs。

`describe-vault`

以下程式碼範例顯示如何使用 `describe-vault`。

AWS CLI

以下命令會擷取名為 `my-vault` 的文件庫資料：

```
aws glacier describe-vault --vault-name my-vault --account-id -
```

Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeVault。

`get-data-retrieval-policy`

以下程式碼範例顯示如何使用 `get-data-retrieval-policy`。

AWS CLI

下列命令會取得使用中帳戶的資料擷取政策：

```
aws glacier get-data-retrieval-policy --account-id -
```

輸出：

```
{
    "Policy": {
        "Rules": [
            {
                "BytesPerHour": 10737418240,
                "Strategy": "BytesPerHour"
            }
```

```
        ]
    }
}
```

Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDataRetrievalPolicy。

get-job-output

以下程式碼範例顯示如何使用 get-job-output。

AWS CLI

以下命令會將文件庫庫存任務的輸出儲存到目前目錄名為 output.json 的檔案中：

```
aws glacier get-job-output --account-id - --vault-name my-
vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61ds04CnMW output.json
```

job-id 可在 aws glacier list-jobs 的輸出中使用。請注意，輸出檔案名稱是位置引數，不以選項名稱作為前綴。Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

輸出：

```
{
    "status": 200,
    "acceptRanges": "bytes",
    "contentType": "application/json"
}
```

output.json:

```
{"VaultARN":"arn:aws:glacier:us-west-2:0123456789012:vaults/
my-vault","InventoryDate":"2015-04-07T00:26:18Z","ArchiveList":
[{"ArchiveId":"kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw","ArchiveDescription":"multipart
 upload
 test","CreationDate":"2015-04-06T22:24:34Z","Size":3145728,"SHA256TreeHash":"9628195fcdbcbb
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetJobOutput。

get-vault-access-policy

以下程式碼範例顯示如何使用 get-vault-access-policy。

AWS CLI

擷取保存庫的存取政策

下列get-vault-access-policy範例會擷取指定保存庫的存取政策。

```
aws glacier get-vault-access-policy \
    --account-id 111122223333 \
    --vault-name example_vault
```

輸出：

```
{
    "policy": {
        "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::444455556666:root
\"},\"Action\":\"glacier:ListJobs\",\"Resource\":\"arn:aws:glacier:us-
east-1:111122223333:vaults/example_vault\"},{\"Effect\":\"Allow\",\"Principal\":
{\"AWS\":\"arn:aws:iam::444455556666:root\"},\"Action\":\"glacier:UploadArchive\",
\"Resource\":\"arn:aws:glacier:us-east-1:111122223333:vaults/example_vault\"}]}"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetVaultAccessPolicy。

get-vault-lock

以下程式碼範例顯示如何使用 get-vault-lock。

AWS CLI

取得保存庫鎖定的詳細資訊

下列get-vault-lock範例擷取指定保存庫鎖定的詳細資訊。

```
aws glacier get-vault-lock \
    --account-id - \
    --vault-name MyVaultName
```

輸出：

```
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"Define-vault-
lock\",\"Effect\":\"Deny\",\"Principal\":{\"AWS\":\"arn:aws:iam::999999999999:root
\"},\"Action\":\"glacier:DeleteArchive\",\"Resource\":\"arn:aws:glacier:us-
west-2:99999999999:vaults/MyVaultName\",\"Condition\":{\"NumericLessThanEquals\":
{\"glacier:ArchiveAgeinDays\":\"365\"}}}]}",
    "State": "Locked",
    "CreationDate": "2019-07-29T22:25:28.640Z"
}
```

如需詳細資訊，請參閱《Amazon Glacier API 開發人員指南》中的取得保存庫鎖定 (GET lock-policy)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetVaultLock。

get-vault-notifications

以下程式碼範例顯示如何使用 get-vault-notifications。

AWS CLI

以下命令會取得名為 my-vault 的文件庫通知組態描述：

```
aws glacier get-vault-notifications --account-id - --vault-name my-vault
```

輸出：

```
{
    "vaultNotificationConfig": {
        "Events": [
            "InventoryRetrievalCompleted",
            "ArchiveRetrievalCompleted"
        ],
        "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault"
    }
```

```
}
```

如果尚未為文件庫設定任何通知，則會傳回錯誤。Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetVaultNotifications。

initiate-job

以下程式碼範例顯示如何使用 initiate-job。

AWS CLI

下列命令會啟動任務，以取得保存庫 的庫存my-vault：

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-parameters
 '{"Type": "inventory-retrieval"}'
```

輸出：

```
{
    "location": "/0123456789012/vaults/my-vault/jobs/
zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW",
    "jobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW"
}
```

Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

下列命令會啟動任務，以從保存庫 擷取封存my-vault：

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-
parameters file://job-archive-retrieval.json
```

job-archive-retrieval.json 是本機資料夾中的 JSON 檔案，指定任務類型、封存 ID 和一些選用參數：

```
{
  "Type": "archive-retrieval",
```

```
  "ArchiveId": "kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw",
  "Description": "Retrieve archive on 2015-07-17",
  "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-topic"
}
```

封存 IDs可在 aws glacier upload-archive和 的輸出中使用aws glacier get-job-output。

輸出：

```
{
    "location": "/011685312445/vaults/mwunderl/jobs/l7IL5-
EkXyEY9Ws95fClzIbk2O5uLYaFdAYOi-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJO59-8N9aHWav",
    "jobId": "l7IL5-EkXy2O5uLYaFdAYOiEY9Ws95fClzIbk-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJO59-8N9aHWav"
}
```

如需任務參數格式的詳細資訊，請參閱《Amazon Glacier API 參考》中的啟動任務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 InitiateJob。

initiate-multipart-upload

以下程式碼範例顯示如何使用 initiate-multipart-upload。

AWS CLI

下列命令會啟動分段上傳至名為 my-vault的保存庫，每個檔案的部分大小為 1 MiB (1024 x 1024 位元組）：

```
aws glacier initiate-multipart-upload --account-id - --part-size 1048576 --vault-
name my-vault --archive-description "multipart upload test"
```

封存描述參數是選用的。Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

此命令會在成功時輸出上傳 ID。使用 上傳封存的每個部分時，請使用上傳 IDaws glacier upload-multipart-part。如需使用 CLI 將分段上傳至 Amazon Glacier AWS 的詳細資訊，請參閱《AWS CLI 使用者指南》中的使用 Amazon Glacier。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 InitiateMultipartUpload。

initiate-vault-lock

以下程式碼範例顯示如何使用 initiate-vault-lock。

AWS CLI

啟動保存庫鎖定程序

下列initiate-vault-lock範例會在指定的保存庫上安裝保存庫鎖定政策，並將保存庫鎖定的鎖定狀態設定為 InProgress。您必須在 complete-vault-lock 24 小時內呼叫 以完成程序，將保存庫鎖定的狀態設定為 Locked。

```
aws glacier initiate-vault-lock \
    --account-id - \
    --vault-name MyVaultName \
    --policy file://vault_lock_policy.json
```

vault_lock_policy.json 的內容：

```
{"Policy":"{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"Define-vault-
lock\",\"Effect\":\"Deny\",\"Principal\":{\"AWS\":\"arn:aws:iam::999999999999:root
\"},\"Action\":\"glacier:DeleteArchive\",\"Resource\":\"arn:aws:glacier:us-
west-2:999999999999:vaults/examplevault\",\"Condition\":{\"NumericLessThanEquals\":
{\"glacier:ArchiveAgeinDays\":\"365\"}}}]}"}
```

輸出是保存庫鎖定 ID，您可以用來完成保存庫鎖定程序。

```
{
    "lockId": "9QZgEXAMPLEPhvL6xEXAMPLE"
}
```

如需詳細資訊，請參閱《Amazon Glacier API 開發人員指南》中的啟動保存庫鎖定 (POST lock-policy)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 InitiateVaultLock。

list-jobs

以下程式碼範例顯示如何使用 list-jobs。

AWS CLI

以下命令會列出名為 my-vault 的文件庫正在進行和最近完成之工作：

```
aws glacier list-jobs --account-id - --vault-name my-vault
```

輸出：

```
{
    "JobList": [
        {
            "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
            "RetrievalByteRange": "0-3145727",
            "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault",
            "Completed": false,
            "SHA256TreeHash":
 "9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
            "JobId": "l7IL5-EkXyEY9Ws95fClzIbk2O5uLYaFdAYOi-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJO59-8N9aHWav",
            "ArchiveId": "kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw",
            "JobDescription": "Retrieve archive on 2015-07-17",
            "ArchiveSizeInBytes": 3145728,
            "Action": "ArchiveRetrieval",
            "ArchiveSHA256TreeHash":
 "9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
            "CreationDate": "2015-07-17T21:16:13.840Z",
            "StatusCode": "InProgress"
        },
        {
            "InventoryRetrievalParameters": {
                "Format": "JSON"
            },
            "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
            "Completed": false,
            "JobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW",
            "Action": "InventoryRetrieval",
            "CreationDate": "2015-07-17T20:23:41.616Z",
            "StatusCode": ""InProgress""
        }
    ]
```

```
}
```

Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListJobs。

list-multipart-uploads

以下程式碼範例顯示如何使用 list-multipart-uploads。

AWS CLI

下列命令顯示名為 的保存庫的所有進行中分段上傳my-vault：

```
aws glacier list-multipart-uploads --account-id - --vault-name my-vault
```

Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

如需使用 CLI 將分段上傳至 Amazon Glacier AWS 的詳細資訊，請參閱《AWS CLI 使用者指南》中的使用 Amazon Glacier。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMultipartUploads。

list-parts

以下程式碼範例顯示如何使用 list-parts。

AWS CLI

下列命令列出分段上傳至名為 的保存庫的上傳組件my-vault：

```
aws glacier list-parts --account-id - --vault-name my-vault --upload-id "SYZi7qnL-
YGqGwAm8Kn3BLP2ElNCvnB-5961R09CSaPmPwkYGHOqeN_nX3-Vhnd2yF0KfB5FkmbnBU9GubbdrCs8ut-D"
```

輸出：

```
{
    "MultipartUploadId": "SYZi7qnL-
YGqGwAm8Kn3BLP2ElNCvnB-5961R09CSaPmPwkYGHOqeN_nX3-Vhnd2yF0KfB5FkmbnBU9GubbdrCs8ut-
D",
    "Parts": [
        {
```

```
            "RangeInBytes": "0-1048575",
            "SHA256TreeHash":
 "e1f2a7cd6e047350f69b9f8cfa60fa606fe2f02802097a9a026360a7edc1f553"
        },
        {
            "RangeInBytes": "1048576-2097151",
            "SHA256TreeHash":
 "43cf3061fb95796aed99a11a6aa3cd8f839eed15e655ab0a597126210636aee6"
        }
    ],
    "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
    "CreationDate": "2015-07-18T00:05:23.830Z",
    "PartSizeInBytes": 1048576
}
```

Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

如需使用 CLI 將分段上傳至 Amazon Glacier AWS 的詳細資訊，請參閱《AWS CLI 使用者指南》中的使用 Amazon Glacier。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListParts。

`list-provisioned-capacity`

以下程式碼範例顯示如何使用 `list-provisioned-capacity`。

AWS CLI

擷取佈建的容量單位

下列`list-provisioned-capacity`範例會擷取指定帳戶之任何佈建容量單位的詳細資訊。

```
aws glacier list-provisioned-capacity \
    --account-id 111122223333
```

輸出：

```
{
    "ProvisionedCapacityList": [
        {
            "CapacityId": "HpASAuvfRFiVDbOjMfEIcr8K",
            "ExpirationDate": "2020-03-18T19:59:24.000Z",
```

```
            "StartDate": "2020-02-18T19:59:24.912Z"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListProvisionedCapacity。

`list-tags-for-vault`

以下程式碼範例顯示如何使用 `list-tags-for-vault`。

AWS CLI

下列命令會列出套用到 `my-vault` 文件庫的標籤：

```
aws glacier list-tags-for-vault --account-id - --vault-name my-vault
```

輸出：

```
{
    "Tags": {
        "date": "july2015",
        "id": "1234"
    }
}
```

Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForVault。

`list-vaults`

以下程式碼範例顯示如何使用 `list-vaults`。

AWS CLI

下列命令列出預設帳戶與區域的文件庫。

```
aws glacier list-vaults --account-id -
```

輸出：

```
{
    "VaultList": [
        {
            "SizeInBytes": 3178496,
            "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
            "LastInventoryDate": "2015-04-07T00:26:19.028Z",
            "VaultName": "my-vault",
            "NumberOfArchives": 1,
            "CreationDate": "2015-04-06T21:23:45.708Z"
        }
    ]
}
```

Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListVaults。

purchase-provisioned-capacity

以下程式碼範例顯示如何使用 purchase-provisioned-capacity。

AWS CLI

購買佈建容量單位

下列purchase-provisioned-capacity範例會購買佈建容量單位。

```
aws glacier purchase-provisioned-capacity \
    --account-id 111122223333
```

輸出：

```
{
    "capacityId": "HpASAuvfRFiVDbOjMfEIcr8K"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PurchaseProvisionedCapacity。

remove-tags-from-vault

以下程式碼範例顯示如何使用 remove-tags-from-vault。

AWS CLI

下列命令date會從名為 的保存庫中移除具有 金鑰的標籤my-vault：

```
aws glacier remove-tags-from-vault --account-id - --vault-name my-vault --tag-keys date
```

Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveTagsFromVault。

set-data-retrieval-policy

以下程式碼範例顯示如何使用 set-data-retrieval-policy。

AWS CLI

下列命令會設定使用中帳戶的資料擷取政策：

```
aws glacier set-data-retrieval-policy --account-id - --policy file://data-retrieval-policy.json
```

data-retrieval-policy.json 是目前資料夾中的 JSON 檔案，可指定資料擷取政策：

```
{
  "Rules":[
     {
         "Strategy":"BytesPerHour",
         "BytesPerHour":10737418240
      }
   ]
}
```

Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

下列命令會將資料擷取政策設定為FreeTier使用內嵌 JSON：

```
aws glacier set-data-retrieval-policy --account-id - --policy '{"Rules": [{"Strategy":"FreeTier"}]}'
```

如需政策格式的詳細資訊，請參閱《Amazon Glacier API 參考》中的設定資料擷取政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetDataRetrievalPolicy。

set-vault-access-policy

以下程式碼範例顯示如何使用 set-vault-access-policy。

AWS CLI

設定保存庫的存取政策

下列set-vault-access-policy範例會將許可政策連接至指定的保存庫。

```
aws glacier set-vault-access-policy \
    --account-id 111122223333 \
    --vault-name example_vault
    --policy '{"Policy": "{\"Version\":\"2012-10-17\",\"Statement\":
[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::444455556666:root
\"},\"Action\":\"glacier:ListJobs\",\"Resource\":\"arn:aws:glacier:us-
east-1:111122223333:vaults/example_vault\"},{\"Effect\":\"Allow\",\"Principal\":
{\"AWS\":\"arn:aws:iam::444455556666:root\"},\"Action\":\"glacier:UploadArchive\",
\"Resource\":\"arn:aws:glacier:us-east-1:111122223333:vaults/example_vault\"}]}"}'
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetVaultAccessPolicy。

set-vault-notifications

以下程式碼範例顯示如何使用 set-vault-notifications。

AWS CLI

下列命令可為名為 my-vault 的文件庫設定 SNS 通知：

```
aws glacier set-vault-notifications --account-id - --vault-name my-vault --vault-
notification-config file://notificationconfig.json
```

notificationconfig.json 是目前資料夾中的 JSON 檔案，用來指定 SNS 主題和要發佈的事件：

```
{
  "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault",
  "Events": ["ArchiveRetrievalCompleted", "InventoryRetrievalCompleted"]
}
```

Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 SetVaultNotifications。

upload-archive

以下程式碼範例顯示如何使用 upload-archive。

AWS CLI

以下命令會將目前 archive.zip 資料夾中的封存上傳至名為 my-vault 的文件庫：

```
aws glacier upload-archive --account-id - --vault-name my-vault --body archive.zip
```

輸出：

```
{
    "archiveId": "kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw",
    "checksum": "969fb39823836d81f0cc028195fcdbcbbe76cdde932d4646fa7de5f21e18aa67",
    "location": "/0123456789012/vaults/my-vault/archives/
kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw"
}
```

Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

若要擷取上傳的封存，請使用 Amazon Glacier 啟動任務命令來啟動擷取任務。

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》的 UploadArchive。

upload-multipart-part

以下程式碼範例顯示如何使用 upload-multipart-part。

AWS CLI

下列命令會上傳封存的前 1 MiB (1024 x 1024 位元組) 部分：

```
aws glacier upload-multipart-part --body part1 --range 'bytes
 0-1048575/*' --account-id - --vault-name my-vault --upload-
```

```
id 19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ
```

Amazon Glacier 在執行操作時需要帳戶 ID 引數，但您可以使用連字號來指定使用中的帳戶。

內文參數會取得本機檔案系統上分段檔案的路徑。範圍參數採用 HTTP 內容範圍，指示分段在完成的封存中佔用的位元組。上傳 ID 由 `aws glacier initiate-multipart-upload` 命令傳回，也可以透過 `aws glacier list-multipart-uploads` 獲取。

如需使用 CLI 將分段上傳至 Amazon Glacier AWS 的詳細資訊，請參閱《AWS CLI 使用者指南》中的使用 Amazon Glacier。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UploadMultipartPart。

使用 的 Secrets Manager 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Secrets Manager 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`batch-get-secret-value`

以下程式碼範例顯示如何使用 `batch-get-secret-value`。

AWS CLI

範例 1：擷取依名稱列出的一組秘密的秘密值

下列`batch-get-secret-value`範例會取得三個秘密的秘密值秘密。

```
aws secretsmanager batch-get-secret-value \
    --secret-id-list MySecret1 MySecret2 MySecret3
```

輸出：

```
{
    "SecretValues": [
        {
            "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret1-
a1b2c3",
            "Name": "MySecret1",
            "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa",
            "SecretString": "{\"username\":\"diego_ramirez\",\"password\":\"EXAMPLE-
PASSWORD\",\"engine\":\"mysql\",\"host\":\"secretsmanagertutorial.cluster.us-
west-2.rds.amazonaws.com\",\"port\":3306,\"dbClusterIdentifier\":
\"secretsmanagertutorial\"}",
            "VersionStages": [
                "AWSCURRENT"
            ],
            "CreatedDate": "1523477145.729"
        },
        {
            "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret2-
a1b2c3",
            "Name": "MySecret2",
            "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
            "SecretString": "{\"username\":\"akua_mansa\",\"password\":\"EXAMPLE-
PASSWORD\"",
            "VersionStages": [
                "AWSCURRENT"
            ],
            "CreatedDate": "1673477781.275"
        },
        {
            "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret3-
a1b2c3",
            "Name": "MySecret3",
            "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEccccc",
            "SecretString": "{\"username\":\"jie_liu\",\"password\":\"EXAMPLE-
PASSWORD\"",
            "VersionStages": [
                "AWSCURRENT"
            ],
            "CreatedDate": "1373477721.124"
        }
    ],
    "Errors": []
```

```
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的擷取批次中的AWS 一組秘密。

範例 2：擷取篩選條件所選取之一組秘密的秘密值

下列batch-get-secret-value範例會取得您帳戶中名稱中具有 的秘密值秘密MySecret。依名稱篩選區分大小寫。

```
aws secretsmanager batch-get-secret-value \
    --filters Key="name",Values="MySecret"
```

輸出：

```
{
    "SecretValues": [
        {
            "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret1-
a1b2c3",
            "Name": "MySecret1",
            "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa",
            "SecretString": "{\"username\":\"diego_ramirez\",\"password\":\"EXAMPLE-
PASSWORD\",\"engine\":\"mysql\",\"host\":\"secretsmanagertutorial.cluster.us-
west-2.rds.amazonaws.com\",\"port\":3306,\"dbClusterIdentifier\":
\"secretsmanagertutorial\"}",
            "VersionStages": [
                "AWSCURRENT"
            ],
            "CreatedDate": "1523477145.729"
        },
        {
            "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret2-
a1b2c3",
            "Name": "MySecret2",
            "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
            "SecretString": "{\"username\":\"akua_mansa\",\"password\":\"EXAMPLE-
PASSWORD\"",
            "VersionStages": [
                "AWSCURRENT"
            ],
            "CreatedDate": "1673477781.275"
        },
        {
```

```
            "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret3-
a1b2c3",
            "Name": "MySecret3",
            "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEccccc",
            "SecretString": "{\"username\":\"jie_liu\",\"password\":\"EXAMPLE-
PASSWORD\"",
            "VersionStages": [
                "AWSCURRENT"
            ],
            "CreatedDate": "1373477721.124"
        }
    ],
    "Errors": []
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的擷取批次中的一組AWS 秘密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetSecretValue。

cancel-rotate-secret

以下程式碼範例顯示如何使用 cancel-rotate-secret。

AWS CLI

關閉秘密的自動輪換

下列cancel-rotate-secret範例會關閉秘密的自動輪換。若要繼續輪換，請呼叫 rotate-secret。

```
aws secretsmanager cancel-rotate-secret \
    --secret-id MyTestSecret
```

輸出：

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
  "Name": "MyTestSecret"
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的輪換秘密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelRotateSecret。

create-secret

以下程式碼範例顯示如何使用 create-secret。

AWS CLI

範例 1：從 JSON 檔案中的登入資料建立秘密

下列 create-secret 範例會透過檔案中的憑證建立機密。如需詳細資訊，請參閱《CLI AWS 使用者指南》中的從檔案載入 CLI 參數。 AWS

```
aws secretsmanager create-secret \
    --name MyTestSecret \
    --secret-string file://mycreds.json
```

mycreds.json 的內容：

```
{
  "engine": "mysql",
  "username": "saanvis",
  "password": "EXAMPLE-PASSWORD",
  "host": "my-database-endpoint.us-west-2.rds.amazonaws.com",
  "dbname": "myDatabase",
  "port": "3306"
}
```

輸出：

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
  "Name": "MyTestSecret",
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的建立秘密。

範例 2：建立秘密

下列 create-secret 範例會建立具有兩個金鑰值對的機密。在命令 shell 中輸入命令時，存在命令歷史記錄被存取或公用程式存取命令參數的風險。如果命令包含秘密的值，這是個問題。如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的減輕使用命令列工具存放秘密的風險。

```
aws secretsmanager create-secret \
    --name MyTestSecret \
    --description "My test secret created with the CLI." \
    --secret-string "{\"user\":\"diegor\",\"password\":\"EXAMPLE-PASSWORD\"}"
```

輸出：

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
  "Name": "MyTestSecret",
  "VersionId": "EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE"
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的建立秘密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSecret。

delete-resource-policy

以下程式碼範例顯示如何使用 delete-resource-policy。

AWS CLI

刪除連接至秘密的資源型政策

下列 delete-resource-policy 範例會刪除連接至機密的以資源為基礎的政策。

```
aws secretsmanager delete-resource-policy \
    --secret-id MyTestSecret
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的身分驗證和存取控制。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteResourcePolicy。

delete-secret

以下程式碼範例顯示如何使用 delete-secret。

AWS CLI

範例 1：刪除秘密

下列 delete-secret 範例會刪除機密。您可以使用 復原秘密，restore-secret直到DeletionDate回應欄位中的日期和時間為止。若要刪除複寫至其他區域的機密，請先使用 remove-regions-from-replication 移除複本，然後呼叫 delete-secret。

```
aws secretsmanager delete-secret \
    --secret-id MyTestSecret \
    --recovery-window-in-days 7
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret",
    "DeletionDate": 1524085349.095
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的刪除秘密。

範例 2：立即刪除秘密

下列 delete-secret 範例會在沒有復原時段的情況下立即刪除機密。您無法復原此機密。

```
aws secretsmanager delete-secret \
    --secret-id MyTestSecret \
    --force-delete-without-recovery
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret",
    "DeletionDate": 1508750180.309
```

```
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的刪除秘密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSecret。

describe-secret

以下程式碼範例顯示如何使用 describe-secret。

AWS CLI

擷取秘密的詳細資訊

下列describe-secret範例顯示秘密的詳細資訊。

```
aws secretsmanager describe-secret \
    --secret-id MyTestSecret
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
Ca8JGt",
    "Name": "MyTestSecret",
    "Description": "My test secret",
    "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/EXAMPLE1-90ab-cdef-fedc-
ba987EXAMPLE",
    "RotationEnabled": true,
    "RotationLambdaARN": "arn:aws:lambda:us-
west-2:123456789012:function:MyTestRotationLambda",
    "RotationRules": {
        "AutomaticallyAfterDays": 2,
        "Duration": "2h",
        "ScheduleExpression": "cron(0 16 1,15 * ? *)"
    },
    "LastRotatedDate": 1525747253.72,
    "LastChangedDate": 1523477145.729,
    "LastAccessedDate": 1524572133.25,
    "Tags": [
        {
            "Key": "SecondTag",
            "Value": "AnotherValue"
```

```
        },
        {
            "Key": "FirstTag",
            "Value": "SomeValue"
        }
    ],
    "VersionIdsToStages": {
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111": [
            "AWSPREVIOUS"
        ],
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222": [
            "AWSCURRENT"
        ],
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333": [
            "AWSPENDING"
        ]
    },
    "CreatedDate": 1521534252.66,
    "PrimaryRegion": "us-west-2",
    "ReplicationStatus": [
        {
            "Region": "eu-west-3",
            "KmsKeyId": "alias/aws/secretsmanager",
            "Status": "InSync",
            "StatusMessage": "Replication succeeded"
        }
    ]
}
```

如需詳細資訊，請參閱《Secret sManager 使用者指南》中的機密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSecret。

get-random-password

以下程式碼範例顯示如何使用 get-random-password。

AWS CLI

產生隨機密碼

下列get-random-password範例會產生長度為 20 個字元的隨機密碼，其中包含至少一個大寫字母、小寫字母、數字和標點符號。

```
aws secretsmanager get-random-password \
    --require-each-included-type \
    --password-length 20
```

輸出：

```
{
    "RandomPassword": "EXAMPLE-PASSWORD"
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的建立和管理秘密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRandomPassword。

get-resource-policy

以下程式碼範例顯示如何使用 get-resource-policy。

AWS CLI

擷取連接至秘密的資源型政策

下列 get-resource-policy 範例會擷取連接至機密的以資源為基礎的政策。

```
aws secretsmanager get-resource-policy \
    --secret-id MyTestSecret
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret",
    "ResourcePolicy": "{\n\"Version\":\"2012-10-17\",\n\"Statement\":[{\n\"Effect\":
\"Allow\",\n
    \"Principal\":{\n\"AWS\":\"arn:aws:iam::123456789012:root\"\n},\n\"Action\":
    \"secretsmanager:GetSecretValue\",\n\"Resource\":\"*\"\n}]\n}"
}
```

如需詳細資訊，請參閱 Secrets Manager 使用者指南中的身分驗證和存取控制。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResourcePolicy。

get-secret-value

以下程式碼範例顯示如何使用 get-secret-value。

AWS CLI

範例 1：擷取秘密的加密秘密值

下列 get-secret-value 範例會取得目前機密值。

```
aws secretsmanager get-secret-value \
    --secret-id MyTestSecret
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "SecretString": "{\"user\":\"diegor\",\"password\":\"EXAMPLE-PASSWORD\"}",
    "VersionStages": [
        "AWSCURRENT"
    ],
    "CreatedDate": 1523477145.713
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的擷取秘密。

範例 2：擷取先前的秘密值

下列get-secret-value範例會取得先前的秘密值：

```
aws secretsmanager get-secret-value \
    --secret-id MyTestSecret
    --version-stage AWSPREVIOUS
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "SecretString": "{\"user\":\"diegor\",\"password\":\"PREVIOUS-EXAMPLE-PASSWORD
\"}",
    "VersionStages": [
        "AWSPREVIOUS"
    ],
    "CreatedDate": 1523477145.713
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的擷取秘密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSecretValue。

list-secret-version-ids

以下程式碼範例顯示如何使用 list-secret-version-ids。

AWS CLI

列出與秘密相關聯的所有秘密版本

下列list-secret-version-ids範例會取得秘密所有版本的清單。

```
aws secretsmanager list-secret-version-ids \
    --secret-id MyTestSecret
```

輸出：

```
{
  "Versions": [
    {
        "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "VersionStages": [
            "AWSPREVIOUS"
        ],
        "LastAccessedDate": 1523477145.713,
        "CreatedDate": 1523477145.713
    },
```

```
    {
        "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "VersionStages": [
            "AWSCURRENT"
        ],
        "LastAccessedDate": 1523477145.713,
        "CreatedDate": 1523486221.391
    },
    {
        "CreatedDate": 1.51197446236E9,
        "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333;"
    }
    ],
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

如需詳細資訊，請參閱 Secrets Manager 使用者指南中的版本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSecretVersionIds。

`list-secrets`

以下程式碼範例顯示如何使用 `list-secrets`。

AWS CLI

範例 1：列出您帳戶中的秘密

下列 `list-secrets` 範例會取得您帳戶中的機密清單。

```
aws secretsmanager list-secrets
```

輸出：

```
{
    "SecretList": [
        {
            "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-a1b2c3",
            "Name": "MyTestSecret",
```

```
            "LastChangedDate": 1523477145.729,
            "SecretVersionsToStages": {
                "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111": [
                    "AWSCURRENT"
                ]
            }
        },
        {
            "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:AnotherSecret-d4e5f6",
            "Name": "AnotherSecret",
            "LastChangedDate": 1523482025.685,
            "SecretVersionsToStages": {
                "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222": [
                    "AWSCURRENT"
                ]
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的尋找秘密。

範例 2：篩選您帳戶中的秘密清單

下列list-secrets範例會取得您帳戶中名稱Test中具有 的秘密清單。依名稱篩選區分大小寫。

```
aws secretsmanager list-secrets \
    --filter Key="name",Values="Test"
```

輸出：

```
{
    "SecretList": [
        {
            "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-a1b2c3",
            "Name": "MyTestSecret",
            "LastChangedDate": 1523477145.729,
            "SecretVersionsToStages": {
                "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111": [
                    "AWSCURRENT"
                ]
```

```
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的尋找秘密。

範例 3：列出您帳戶中由其他服務管理的秘密

下列list-secrets範例會傳回您帳戶中由 Amazon RDS 管理的秘密。

```
aws secretsmanager list-secrets \
    --filter Key="owning-service",Values="rds"
```

輸出：

```
{
    "SecretList": [
        {
            "Name": "rds!cluster-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Tags": [
                {
                    "Value": "arn:aws:rds:us-
west-2:123456789012:cluster:database-1",
                    "Key": "aws:rds:primaryDBClusterArn"
                },
                {
                    "Value": "rds",
                    "Key": "aws:secretsmanager:owningService"
                }
            ],
            "RotationRules": {
                "AutomaticallyAfterDays": 1
            },
            "LastChangedDate": 1673477781.275,
            "LastRotatedDate": 1673477781.26,
            "SecretVersionsToStages": {
                "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa": [
                    "AWSPREVIOUS"
                ],
                "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb": [
                    "AWSCURRENT",
                    "AWSPENDING"
```

```
                ]
            },
            "OwningService": "rds",
            "RotationEnabled": true,
            "CreatedDate": 1673467300.7,
            "LastAccessedDate": 1673395200.0,
            "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:rds!
cluster-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111-a1b2c3",
            "Description": "Secret associated with primary RDS DB cluster:
 arn:aws:rds:us-west-2:123456789012:cluster:database-1"
        }
    ]
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的其他 服務管理的機密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSecrets。

put-resource-policy

以下程式碼範例顯示如何使用 put-resource-policy。

AWS CLI

將資源型政策新增至秘密

下列 put-resource-policy 範例會將許可政策新增至機密，首先檢查政策是否不提供機密的廣泛存取權限。系統會從檔案讀取政策。如需詳細資訊，請參閱《CLI AWS 使用者指南》中的從檔案載入 CLI 參數。 AWS

```
aws secretsmanager put-resource-policy \
    --secret-id MyTestSecret \
    --resource-policy file://mypolicy.json \
    --block-public-policy
```

mypolicy.json 的內容：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/MyRole"
            },
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "*"
        }
    ]
}
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的將許可政策連接至秘密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutResourcePolicy。

put-secret-value

以下程式碼範例顯示如何使用 put-secret-value。

AWS CLI

範例 1：在秘密中存放新的秘密值

下列put-secret-value範例會使用兩個鍵值對建立新的秘密版本。

```
aws secretsmanager put-secret-value \
    --secret-id MyTestSecret \
    --secret-string "{\"user\":\"diegor\",\"password\":\"EXAMPLE-PASSWORD\"}"
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-1a2b3c",
    "Name": "MyTestSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```
    "VersionStages": [
        "AWSCURRENT"
    ]
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的修改秘密。

範例 2：從 JSON 檔案中的登入資料存放新的秘密值

下列 put-secret-value 範例會透過檔案中的憑證建立新版本的機密。如需詳細資訊，請參閱《CLI AWS 使用者指南》中的從檔案載入 AWS CLI 參數。

```
aws secretsmanager put-secret-value \
    --secret-id MyTestSecret \
    --secret-string file://mycreds.json
```

mycreds.json 的內容：

```
{
  "engine": "mysql",
  "username": "saanvis",
  "password": "EXAMPLE-PASSWORD",
  "host": "my-database-endpoint.us-west-2.rds.amazonaws.com",
  "dbname": "myDatabase",
  "port": "3306"
}
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "VersionStages": [
        "AWSCURRENT"
    ]
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的修改秘密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutSecretValue。

remove-regions-from-replication

以下程式碼範例顯示如何使用 remove-regions-from-replication。

AWS CLI

刪除複本秘密

下列 remove-regions-from-replication 範例會刪除 eu-west-3 中的複本機密。若要刪除複寫至其他區域的主要機密，請先刪除複本，然後呼叫 delete-secret。

```
aws secretsmanager remove-regions-from-replication \
    --secret-id MyTestSecret \
    --remove-replica-regions eu-west-3
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-1a2b3c",
    "ReplicationStatus": []
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的刪除複本秘密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveRegionsFromReplication。

replicate-secret-to-regions

以下程式碼範例顯示如何使用 replicate-secret-to-regions。

AWS CLI

將秘密複寫到另一個區域

下列 replicate-secret-to-regions 範例會將機密複寫至 eu-west-3。複本會使用 AWS 受管金鑰 加密aws/secretsmanager。

```
aws secretsmanager replicate-secret-to-regions \
    --secret-id MyTestSecret \
    --add-replica-regions Region=eu-west-3
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-1a2b3c",
    "ReplicationStatus": [
        {
            "Region": "eu-west-3",
            "KmsKeyId": "alias/aws/secretsmanager",
            "Status": "InProgress"
        }
    ]
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的將秘密複寫到另一個區域。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReplicateSecretToRegions。

`restore-secret`

以下程式碼範例顯示如何使用 `restore-secret`。

AWS CLI

還原先前刪除的秘密

下列 `restore-secret` 範例會還原先前排程刪除的機密。

```
aws secretsmanager restore-secret \
    --secret-id MyTestSecret
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的刪除秘密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreSecret。

rotate-secret

以下程式碼範例顯示如何使用 rotate-secret。

AWS CLI

範例 1：設定和啟動秘密的自動輪換

下列rotate-secret範例會設定並啟動秘密的自動輪換。Secrets Manager 會立即輪換秘密一次，然後在兩個小時時段內每八小時輪換一次。輸出會顯示由輪換所建立VersionId之新秘密版本的 。

```
aws secretsmanager rotate-secret \
    --secret-id MyTestDatabaseSecret \
    --rotation-lambda-arn arn:aws:lambda:us-
west-2:1234566789012:function:SecretsManagerTestRotationLambda \
    --rotation-rules "{\"ScheduleExpression\": \"cron(0 8/8 * * ? *)\", \"Duration
\": \"2h\"}"
```

輸出：

```
{
    "ARN": "aws:arn:secretsmanager:us-
west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
    "Name": "MyTestDatabaseSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的輪換秘密。

範例 2：設定和啟動輪換間隔的自動輪換

下列rotate-secret範例會設定並啟動秘密的自動輪換。Secrets Manager 會立即輪換秘密一次，然後每 10 天輪換一次。輸出會顯示由輪換所建立VersionId之新秘密版本的 。

```
aws secretsmanager rotate-secret \
    --secret-id MyTestDatabaseSecret \
    --rotation-lambda-arn arn:aws:lambda:us-
west-2:1234566789012:function:SecretsManagerTestRotationLambda \
    --rotation-rules "{\"ScheduleExpression\": \"rate(10 days)\"}"
```

輸出：

```
{
    "ARN": "aws:arn:secretsmanager:us-
west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
    "Name": "MyTestDatabaseSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的輪換秘密。

範例 3：立即輪換秘密

下列 rotate-secret 範例會立即開始輪換。輸出會顯示由輪換所建立VersionId之新秘密版本的 。機密必須已設定輪換。

```
aws secretsmanager rotate-secret \
    --secret-id MyTestDatabaseSecret
```

輸出：

```
{
    "ARN": "aws:arn:secretsmanager:us-
west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
    "Name": "MyTestDatabaseSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的輪換秘密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RotateSecret。

stop-replication-to-replica

以下程式碼範例顯示如何使用 stop-replication-to-replica。

AWS CLI

將複本秘密提升為主要

下列 stop-replication-to-replica 範例會移除複本機密至主要機密之間的連結。複本機密會提升為複本區域中的主要機密。您必須從複本區域內呼叫 stop-replication-to-replica。

```
aws secretsmanager stop-replication-to-replica \
    --secret-id MyTestSecret
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3"
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的提升複本秘密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopReplicationToReplica。

`tag-resource`

以下程式碼範例顯示如何使用 `tag-resource`。

AWS CLI

範例 1：將標籤新增至秘密

下列 範例顯示如何使用速記語法連接標籤。

```
aws secretsmanager tag-resource \
    --secret-id MyTestSecret \
    --tags Key=FirstTag,Value=FirstValue
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的標記您的秘密。

範例 2：將多個標籤新增至秘密

下列 `tag-resource` 範例會將兩個金鑰值標籤連接至機密。

```
aws secretsmanager tag-resource \
    --secret-id MyTestSecret \
    --tags '[{"Key": "FirstTag", "Value": "FirstValue"}, {"Key": "SecondTag",
 "Value": "SecondValue"}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的標記秘密。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從秘密中移除標籤

下列 untag-resource 範例會從機密中移除兩個標籤。對於每個標籤，金鑰和值都會移除。

```
aws secretsmanager untag-resource \
    --secret-id MyTestSecret \
    --tag-keys '[ "FirstTag", "SecondTag"]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的標記秘密。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-secret-version-stage

以下程式碼範例顯示如何使用 update-secret-version-stage。

AWS CLI

範例 1：將秘密還原至先前的版本

下列update-secret-version-stage範例會將 AWS CURRENT 預備標籤移至舊版的秘密，將秘密還原至舊版。若要尋找先前版本的 ID，請使用 list-secret-version-ids。在此範例中，CURRENT AWS標籤的版本為 a1b2c3d4-5678-90ab-cdef-EXAMPLE11111，而 AWS PREVIOUS 標籤的版本為 a1b2c3d4-5678-90ab-cdef-EXAMPLE22222。在此範例中，您將 AWS CURRENT 標籤從版本 11111 移至 22222。由於 CURRENT AWS標籤已從版本中移除，update-secret-version-stage會自動將 AWS PREVIOUS 標籤移至該版本 (11111)。效果是交換 CURRENT AWS和 AWS PREVIOUS 版本。

```
aws secretsmanager update-secret-version-stage \
    --secret-id MyTestSecret \
    --version-stage AWSCURRENT \
    --move-to-version-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 \
    --remove-from-version-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

如需詳細資訊，請參閱 Secrets Manager 使用者指南中的版本。

範例 2：新增附加至秘密版本的預備標籤

下列update-secret-version-stage範例會將預備標籤新增至秘密的版本。您可以透過執行list-secret-version-ids和檢視受影響版本的VersionStages回應欄位來檢閱結果。

```
aws secretsmanager update-secret-version-stage \
    --secret-id MyTestSecret \
    --version-stage STAGINGLABEL1 \
    --move-to-version-id EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

如需詳細資訊，請參閱 Secrets Manager 使用者指南中的版本。

範例 3：刪除連接至秘密版本的預備標籤

下列update-secret-version-stage範例會刪除連接至秘密版本的預備標籤。您可以透過執行list-secret-version-ids和檢視受影響版本的VersionStages回應欄位來檢閱結果。

```
aws secretsmanager update-secret-version-stage \
    --secret-id MyTestSecret \
    --version-stage STAGINGLABEL1 \
    --remove-from-version-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

如需詳細資訊，請參閱 Secrets Manager 使用者指南中的版本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSecretVersionStage。

update-secret

以下程式碼範例顯示如何使用 update-secret。

AWS CLI

範例 1：更新秘密的描述

下列 update-secret 範例會更新機密的描述。

```
aws secretsmanager update-secret \
    --secret-id MyTestSecret \
    --description "This is a new description for the secret."
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的修改秘密。

範例 2：更新與秘密相關聯的加密金鑰

下列 update-secret 範例會更新用於加密機密值的 KMS 金鑰。KMS 金鑰必須位於與機密相同的區域。

```
aws secretsmanager update-secret \
    --secret-id MyTestSecret \
    --kms-key-id arn:aws:kms:us-west-2:123456789012:key/EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE
```

輸出：

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
    "Name": "MyTestSecret"
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的修改秘密。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSecret。

validate-resource-policy

以下程式碼範例顯示如何使用 validate-resource-policy。

AWS CLI

驗證資源政策

下列validate-resource-policy範例會檢查資源政策是否未授予對秘密的廣泛存取權。政策會從磁碟上的檔案讀取。如需詳細資訊，請參閱《CLI AWS 使用者指南》中的從檔案載入 CLI 參數。AWS

```
aws secretsmanager validate-resource-policy \
    --resource-policy file://mypolicy.json
```

mypolicy.json 的內容：

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/MyRole"
            },
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "*"
        }
    ]
}
```

輸出：

```
{
    "PolicyValidationPassed": true,
    "ValidationErrors": []
}
```

如需詳細資訊，請參閱《Secrets Manager 使用者指南》中的 Secrets Manager 的許可參考。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ValidateResourcePolicy。

使用 的 Security Hub 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Security Hub 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`accept-administrator-invitation`

以下程式碼範例顯示如何使用 `accept-administrator-invitation`。

AWS CLI

接受來自管理員帳戶的邀請

下列accept-administrator-invitation範例接受來自指定管理員帳戶的指定邀請。

```
aws securityhub accept-invitation \
    --administrator-id 123456789012 \
    --invitation-id 7ab938c5d52d7904ad09f9e7c20cc4eb
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理管理員和成員帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptAdministratorInvitation。

accept-invitation

以下程式碼範例顯示如何使用 accept-invitation。

AWS CLI

接受來自管理員帳戶的邀請

下列accept-invitation範例接受來自指定管理員帳戶的指定邀請。

```
aws securityhub accept-invitation \
    --master-id 123456789012 \
    --invitation-id 7ab938c5d52d7904ad09f9e7c20cc4eb
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理管理員和成員帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptInvitation。

batch-delete-automation-rules

以下程式碼範例顯示如何使用 batch-delete-automation-rules。

AWS CLI

刪除自動化規則

下列batch-delete-automation-rules範例會刪除指定的自動化規則。您可以使用單一命令刪除一或多個規則。只有 Security Hub 管理員帳戶可以執行此命令。

```
aws securityhub batch-delete-automation-rules \
    --automation-rules-arns '["arn:aws:securityhub:us-
east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"]'
```

輸出：

```
{
    "ProcessedAutomationRules": [
        "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    ],
    "UnprocessedAutomationRules": []
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的刪除自動化規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDeleteAutomationRules。

batch-disable-standards

以下程式碼範例顯示如何使用 batch-disable-standards。

AWS CLI

停用標準

下列batch-disable-standards範例會停用與指定訂閱 ARN 相關聯的標準。

```
aws securityhub batch-disable-standards \
    --standards-subscription-arns "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"
```

輸出：

```
{
    "StandardsSubscriptions": [
        {
            "StandardsArn": "arn:aws:securityhub:eu-central-1::standards/pci-dss/
v/3.2.1",
```

```
            "StandardsInput": { },
            "StandardsStatus": "DELETING",
            "StandardsSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的停用或啟用安全標準。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchDisableStandards。

batch-enable-standards

以下程式碼範例顯示如何使用 batch-enable-standards。

AWS CLI

啟用標準

下列batch-enable-standards範例會啟用請求帳戶的 PCI DSS 標準。

```
aws securityhub batch-enable-standards \
    --standards-subscription-requests '{"StandardsArn":"arn:aws:securityhub:us-
west-1::standards/pci-dss/v/3.2.1"}'
```

輸出：

```
{
    "StandardsSubscriptions": [
        {
            "StandardsArn": "arn:aws:securityhub:us-west-1::standards/pci-dss/
v/3.2.1",
            "StandardsInput": { },
            "StandardsStatus": "PENDING",
            "StandardsSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的停用或啟用安全標準。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchEnableStandards。

batch-get-automation-rules

以下程式碼範例顯示如何使用 batch-get-automation-rules。

AWS CLI

取得自動化規則的詳細資訊

下列batch-get-automation-rules範例會取得指定自動化規則的詳細資訊。您可以使用單一命令取得一或多個自動化規則的詳細資訊。

```
aws securityhub batch-get-automation-rules \
    --automation-rules-arns '["arn:aws:securityhub:us-
east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"]'
```

輸出：

```
{
    "Rules": [
        {
            "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "RuleStatus": "ENABLED",
            "RuleOrder": 1,
            "RuleName": "Suppress informational findings",
            "Description": "Suppress GuardDuty findings with Informational
 severity",
            "IsTerminal": false,
            "Criteria": {
                "ProductName": [
                    {
                        "Value": "GuardDuty",
                        "Comparison": "EQUALS"
                    }
                ],
                "SeverityLabel": [
                    {
                        "Value": "INFORMATIONAL",
                        "Comparison": "EQUALS"
                    }
                ],
```

```
                "WorkflowStatus": [
                    {
                        "Value": "NEW",
                        "Comparison": "EQUALS"
                    }
                ],
                "RecordState": [
                    {
                        "Value": "ACTIVE",
                        "Comparison": "EQUALS"
                    }
                ]
            },
            "Actions": [
                {
                    "Type": "FINDING_FIELDS_UPDATE",
                    "FindingFieldsUpdate": {
                        "Note": {
                            "Text": "Automatically suppress GuardDuty findings with
 Informational severity",
                            "UpdatedBy": "sechub-automation"
                        },
                        "Workflow": {
                            "Status": "SUPPRESSED"
                        }
                    }
                }
            ],
            "CreatedAt": "2023-05-31T17:56:14.837000+00:00",
            "UpdatedAt": "2023-05-31T17:59:38.466000+00:00",
            "CreatedBy": "arn:aws:iam::123456789012:role/Admin"
        }
    ],
    "UnprocessedAutomationRules": []
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的檢視自動化規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetAutomationRules。

batch-get-configuration-policy-associations

以下程式碼範例顯示如何使用 batch-get-configuration-policy-associations。

AWS CLI

取得批次目標的組態關聯詳細資訊

下列batch-get-configuration-policy-associations範例會擷取指定目標的關聯詳細資訊。您可以提供帳戶 IDs、組織單位 IDs 或目標的根 ID。

```
aws securityhub batch-get-configuration-policy-associations \
    --target '{"OrganizationalUnitId": "ou-6hi7-8j91kl2m"}'
```

輸出：

```
{
    "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "TargetId": "ou-6hi7-8j91kl2m",
    "TargetType": "ORGANIZATIONAL_UNIT",
    "AssociationType": "APPLIED",
    "UpdatedAt": "2023-09-26T21:13:01.816000+00:00",
    "AssociationStatus": "SUCCESS",
    "AssociationStatusMessage": "Association applied successfully on this target."
}
```

如需詳細資訊，請參閱 Security Hub 使用者指南中的檢視 Security Hub 組態政策。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetConfigurationPolicyAssociations。

batch-get-security-controls

以下程式碼範例顯示如何使用 batch-get-security-controls。

AWS CLI

若要取得安全控制詳細資訊

下列batch-get-security-controls範例取得目前 AWS 帳戶和 AWS 區域中安全控制 ACM.1 和 IAM.1 的詳細資訊。

```
aws securityhub batch-get-security-controls \
    --security-control-ids '["ACM.1", "IAM.1"]'
```

輸出：

```
{
    "SecurityControls": [
        {
            "SecurityControlId": "ACM.1",
            "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/ACM.1",
            "Title": "Imported and ACM-issued certificates should be renewed after a
 specified time period",
            "Description": "This control checks whether an AWS Certificate Manager
 (ACM) certificate is renewed within the specified time period. It checks both
 imported certificates and certificates provided by ACM. The control fails if the
 certificate isn't renewed within the specified time period. Unless you provide a
 custom parameter value for the renewal period, Security Hub uses a default value of
 30 days.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
ACM.1/remediation",
            "SeverityRating": "MEDIUM",
            "SecurityControlStatus": "ENABLED"
            "UpdateStatus": "READY",
            "Parameters": {
                "daysToExpiration": {
                    "ValueType": CUSTOM,
                    "Value": {
                        "Integer": 15
                    }
                }
            },
            "LastUpdateReason": "Updated control parameter"
        },
        {
            "SecurityControlId": "IAM.1",
            "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/IAM.1",
            "Title": "IAM policies should not allow full \"*\" administrative
 privileges",
            "Description": "This AWS control checks whether the default version of
 AWS Identity and Access Management (IAM) policies (also known as customer managed
 policies) do not have administrator access with a statement that has \"Effect\":
 \"Allow\" with \"Action\": \"*\" over \"Resource\": \"*\". It only checks for
 the Customer Managed Policies that you created, but not inline and AWS Managed
 Policies.",
```

```
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
IAM.1/remediation",
            "SeverityRating": "HIGH",
            "SecurityControlStatus": "ENABLED"
            "UpdateStatus": "READY",
            "Parameters": {}
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的檢視控制項的詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetSecurityControls。

batch-get-standards-control-associations

以下程式碼範例顯示如何使用 batch-get-standards-control-associations。

AWS CLI

取得控制項的啟用狀態

下列batch-get-standards-control-associations範例會識別指定的控制項是否在指定的標準中啟用。

```
aws securityhub batch-get-standards-control-associations \
    --standards-control-association-ids '[{"SecurityControlId":
 "Config.1","StandardsArn": "arn:aws:securityhub:us-east-1:123456789012:ruleset/cis-
aws-foundations-benchmark/v/1.2.0"}, {"SecurityControlId": "IAM.6","StandardsArn":
 "arn:aws:securityhub:us-east-1:123456789012:standards/aws-foundational-security-
best-practices/v/1.0.0"}]'
```

輸出：

```
{
    "StandardsControlAssociationDetails": [
        {
            "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-
benchmark/v/1.2.0",
            "SecurityControlId": "Config.1",
            "SecurityControlArn": "arn:aws:securityhub:us-
east-1:068873283051:security-control/Config.1",
```

```
            "AssociationStatus": "ENABLED",
            "RelatedRequirements": [
                "CIS AWS Foundations 2.5"
            ],
            "UpdatedAt": "2022-10-27T16:07:12.960000+00:00",
            "StandardsControlTitle": "Ensure AWS Config is enabled",
            "StandardsControlDescription": "AWS Config is a web service that
 performs configuration management of supported AWS resources within your account
 and delivers log files to you. The recorded information includes the configuration
 item (AWS resource), relationships between configuration items (AWS resources), and
 any configuration changes between resources. It is recommended to enable AWS Config
 in all regions.",
            "StandardsControlArns": [
                "arn:aws:securityhub:us-east-1:068873283051:control/cis-aws-
foundations-benchmark/v/1.2.0/2.5"
            ]
        },
        {
            "StandardsArn": "arn:aws:securityhub:us-east-1::standards/aws-
foundational-security-best-practices/v/1.0.0",
            "SecurityControlId": "IAM.6",
            "SecurityControlArn": "arn:aws:securityhub:us-
east-1:068873283051:security-control/IAM.6",
            "AssociationStatus": "DISABLED",
            "RelatedRequirements": [],
            "UpdatedAt": "2022-11-22T21:30:35.080000+00:00",
            "UpdatedReason": "test",
            "StandardsControlTitle": "Hardware MFA should be enabled for the root
 user",
            "StandardsControlDescription": "This AWS control checks whether your AWS
 account is enabled to use a hardware multi-factor authentication (MFA) device to
 sign in with root user credentials.",
            "StandardsControlArns": [
                "arn:aws:securityhub:us-east-1:068873283051:control/aws-
foundational-security-best-practices/v/1.0.0/IAM.6"
            ]
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的在特定標準中啟用和停用控制項。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchGetStandardsControlAssociations。

batch-import-findings

以下程式碼範例顯示如何使用 batch-import-findings。

AWS CLI

更新問題清單

下列batch-import-findings範例會更新問題清單。

```
aws securityhub batch-import-findings \
     --findings '
        [{
            "AwsAccountId": "123456789012",
            "CreatedAt": "2020-05-27T17:05:54.832Z",
            "Description": "Vulnerability in a CloudTrail trail",
            "FindingProviderFields": {
                "Severity": {
                    "Label": "LOW",
                    "Original": "10"
                },
                "Types": [
                    "Software and Configuration Checks/Vulnerabilities/CVE"
                ]
            },
            "GeneratorId": "TestGeneratorId",
            "Id": "Id1",
            "ProductArn": "arn:aws:securityhub:us-
west-1:123456789012:product/123456789012/default",
            "Resources": [
                {
                    "Id": "arn:aws:cloudtrail:us-west-1:123456789012:trail/
TrailName",
                    "Partition": "aws",
                    "Region": "us-west-1",
                    "Type": "AwsCloudTrailTrail"
                }
            ],
            "SchemaVersion": "2018-10-08",
            "Title": "CloudTrail trail vulnerability",
            "UpdatedAt": "2020-06-02T16:05:54.832Z"
        }]'
```

輸出：

```
{
    "FailedCount": 0,
    "SuccessCount": 1,
    "FailedFindings": []
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的使用 BatchImportFindings 建立和更新問題清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchImportFindings。

batch-update-automation-rules

以下程式碼範例顯示如何使用 batch-update-automation-rules。

AWS CLI

更新自動化規則

下列batch-update-automation-rules範例會更新指定的自動化規則。您可以使用單一命令更新一或多個規則。只有 Security Hub 管理員帳戶可以執行此命令。

```
aws securityhub batch-update-automation-rules \
    --update-automation-rules-request-items '[ \
        { \
            "Actions": [{ \
                "Type": "FINDING_FIELDS_UPDATE", \
                "FindingFieldsUpdate": { \
                    "Note": { \
                        "Text": "Known issue that is a risk", \
                        "UpdatedBy": "sechub-automation" \
                    }, \
                    "Workflow": { \
                        "Status": "NEW" \
                    } \
                } \
            }], \
            "Criteria": { \
                "SeverityLabel": [{ \
                    "Value": "LOW", \
                    "Comparison": "EQUALS" \
                }] \
```

```
            }, \
            "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111", \
            "RuleOrder": 1, \
            "RuleStatus": "DISABLED" \
        } \
    ]'
```

輸出：

```
{
    "ProcessedAutomationRules": [
        "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    ],
    "UnprocessedAutomationRules": []
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的編輯自動化規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchUpdateAutomationRules。

batch-update-findings

以下程式碼範例顯示如何使用 batch-update-findings。

AWS CLI

範例 1：更新問題清單

下列batch-update-findings範例會更新兩個問題清單，以新增備註、變更嚴重性標籤，並加以解決。

```
aws securityhub batch-update-findings \
    --finding-identifiers '[{"Id": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111", "ProductArn": "arn:aws:securityhub:us-
west-1::product/aws/securityhub"}, {"Id": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222", "ProductArn": "arn:aws:securityhub:us-
west-1::product/aws/securityhub"}]' \
    --note '{"Text": "Known issue that is not a risk.", "UpdatedBy": "user1"}' \
```

```
    --severity '{"Label": "LOW"}' \
    --workflow '{"Status": "RESOLVED"}'
```

輸出：

```
{
    "ProcessedFindings": [
        {
            "Id": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/
v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub"
        },
        {
            "Id": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/
v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub"
        }
    ],
    "UnprocessedFindings": []
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的使用 BatchUpdateFindings 更新問題清單。

範例 2：使用速記語法更新問題清單

下列batch-update-findings範例會更新兩個調查結果，以新增備註、變更嚴重性標籤，並使用速記語法加以解決。

```
aws securityhub batch-update-findings \
    --finding-identifiers Id="arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",ProductArn="arn:aws:securityhub:us-
west-1::product/aws/securityhub" Id="arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",ProductArn="arn:aws:securityhub:us-
west-1::product/aws/securityhub" \
    --note Text="Known issue that is not a risk.",UpdatedBy="user1" \
    --severity Label="LOW" \
    --workflow Status="RESOLVED"
```

輸出：

```
{
    "ProcessedFindings": [
        {
            "Id": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/
v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub"
        },
        {
            "Id": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/
v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub"
        }
    ],
    "UnprocessedFindings": []
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的使用 BatchUpdateFindings 更新問題清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchUpdateFindings。

batch-update-standards-control-associations

以下程式碼範例顯示如何使用 batch-update-standards-control-associations。

AWS CLI

更新已啟用標準中控制項的啟用狀態

下列batch-update-standards-control-associations範例會在指定的標準中停用 CloudTrail.1。

```
aws securityhub batch-update-standards-control-associations \
    --standards-control-association-updates '[{"SecurityControlId": "CloudTrail.1",
 "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0", "AssociationStatus": "DISABLED", "UpdatedReason": "Not applicable
 to environment"}, {"SecurityControlId": "CloudTrail.1", "StandardsArn":
 "arn:aws:securityhub:::standards/cis-aws-foundations-benchmark/v/1.4.0",
 "AssociationStatus": "DISABLED", "UpdatedReason": "Not applicable to
 environment"}]'
```

此命令成功後就不會產生輸出。

如需詳細資訊，請參閱《AWS Security Hub 使用者指南》中的在特定標準中啟用和停用控制項，以及在所有標準中啟用和停用控制項。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchUpdateStandardsControlAssociations。

create-action-target

以下程式碼範例顯示如何使用 create-action-target。

AWS CLI

建立自訂動作

下列create-action-target範例會建立自訂動作。它提供動作的名稱、描述和識別符。

```
aws securityhub create-action-target \
    --name "Send to remediation" \
    --description "Action to send the finding for remediation tracking" \
    --id "Remediation"
```

輸出：

```
{
    "ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/custom/
Remediation"
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的建立自訂動作並將其與 CloudWatch Events 規則建立關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateActionTarget。

create-automation-rule

以下程式碼範例顯示如何使用 create-automation-rule。

AWS CLI

建立自動化規則

下列create-automation-rule範例會在目前 AWS 帳戶和 AWS 區域中建立自動化規則。Security Hub 會根據指定的條件篩選問題清單，並將動作套用至相符的問題清單。只有 Security Hub 管理員帳戶可以執行此命令。

```
aws securityhub create-automation-rule \
    --actions '[{ \
        "Type": "FINDING_FIELDS_UPDATE", \
        "FindingFieldsUpdate": { \
            "Severity": { \
                "Label": "HIGH" \
            }, \
            "Note": { \
                "Text": "Known issue that is a risk. Updated by automation rules", \
                "UpdatedBy": "sechub-automation" \
            } \
        } \
    }]' \
    --criteria '{ \
        "SeverityLabel": [{ \
            "Value": "INFORMATIONAL", \
            "Comparison": "EQUALS" \
        }] \
    }' \
    --description "A sample rule" \
    --no-is-terminal \
    --rule-name "sample rule" \
    --rule-order 1 \
    --rule-status "ENABLED"
```

輸出：

```
{
    "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的建立自動化規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAutomationRule。

create-configuration-policy

以下程式碼範例顯示如何使用 create-configuration-policy。

AWS CLI

建立組態政策

下列create-configuration-policy範例會建立具有指定設定的組態政策。

```
aws securityhub create-configuration-policy \
    --name "SampleConfigurationPolicy" \
    --description "SampleDescription" \
    --configuration-policy '{"SecurityHub": {"ServiceEnabled":
 true, "EnabledStandardIdentifiers": ["arn:aws:securityhub:eu-
central-1::standards/aws-foundational-security-best-practices/
v/1.0.0","arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"],"SecurityControlsConfiguration":{"DisabledSecurityControlIdentifiers":
 ["CloudTrail.2"], "SecurityControlCustomParameters": [{"SecurityControlId":
 "ACM.1", "Parameters": {"daysToExpiration": {"ValueType": "CUSTOM", "Value":
 {"Integer": 15}}}}]}}}' \
    --tags '{"Environment": "Prod"}'
```

輸出：

```
{
    "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Name": "SampleConfigurationPolicy",
    "Description": "SampleDescription",
    "UpdatedAt": "2023-11-28T20:28:04.494000+00:00",
    "CreatedAt": "2023-11-28T20:28:04.494000+00:00",
    "ConfigurationPolicy": {
        "SecurityHub": {
            "ServiceEnabled": true,
            "EnabledStandardIdentifiers": [
                "arn:aws:securityhub:eu-central-1::standards/aws-foundational-
security-best-practices/v/1.0.0",
                "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"
            ],
            "SecurityControlsConfiguration": {
```

```
                "DisabledSecurityControlIdentifiers": [
                    "CloudTrail.2"
                ],
                "SecurityControlCustomParameters": [
                    {
                        "SecurityControlId": "ACM.1",
                        "Parameters": {
                            "daysToExpiration": {
                                "ValueType": "CUSTOM",
                                "Value": {
                                    "Integer": 15
                                }
                            }
                        }
                    }
                ]
            }
        }
    }
}
```

如需詳細資訊，請參閱 Security Hub 使用者指南中的建立和關聯 Security Hub 組態政策。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateConfigurationPolicy。

create-finding-aggregator

以下程式碼範例顯示如何使用 create-finding-aggregator。

AWS CLI

啟用問題清單彙總

下列create-finding-aggregator範例會設定問題清單彙總。它從美國東部（維吉尼亞）執行，指定美國東部（維吉尼亞）做為彙總區域。它表示 僅連結指定的區域，且 不會自動連結新的區域。它會選取美國西部（加利佛尼亞北部）和美國西部（奧勒岡）做為連結的區域。

```
aws securityhub create-finding-aggregator \
    --region us-east-1 \
    --region-linking-mode SPECIFIED_REGIONS \
    --regions us-west-1,us-west-2
```

輸出：

```
{
    "FindingAggregatorArn": "arn:aws:securityhub:us-east-1:222222222222:finding-
aggregator/123e4567-e89b-12d3-a456-426652340000",
    "FindingAggregationRegion": "us-east-1",
    "RegionLinkingMode": "SPECIFIED_REGIONS",
    "Regions": "us-west-1,us-west-2"
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的啟用問題清單彙總。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFindingAggregator。

`create-insight`

以下程式碼範例顯示如何使用 `create-insight`。

AWS CLI

建立自訂洞見

下列`create-insight`範例會建立名為關鍵角色調查結果的自訂洞見，以傳回與 AWS 角色相關的關鍵調查結果。

```
aws securityhub create-insight \
    --filters '{"ResourceType": [{ "Comparison": "EQUALS", "Value": "AwsIamRole"}],
 "SeverityLabel": [{"Comparison": "EQUALS", "Value": "CRITICAL"}]}' \
    --group-by-attribute "ResourceId" \
    --name "Critical role findings"
```

輸出：

```
{
    "InsightArn": "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理自訂洞見。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateInsight。

create-members

以下程式碼範例顯示如何使用 create-members。

AWS CLI

將帳戶新增為成員帳戶

下列create-members範例會將兩個帳戶新增為請求管理員帳戶的成員帳戶。

```
aws securityhub create-members \
    --account-details '[{"AccountId": "123456789111"}, {"AccountId":
 "123456789222"}]'
```

輸出：

```
{
    "UnprocessedAccounts": []
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理管理員和成員帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateMembers。

decline-invitations

以下程式碼範例顯示如何使用 decline-invitations。

AWS CLI

拒絕成為成員帳戶的邀請

下列decline-invitations範例會拒絕邀請成為指定管理員帳戶的成員帳戶。成員帳戶是請求帳戶。

```
aws securityhub decline-invitations \
    --account-ids "123456789012"
```

輸出：

```
{
    "UnprocessedAccounts": []
```

```
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理管理員和成員帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeclineInvitations。

delete-action-target

以下程式碼範例顯示如何使用 delete-action-target。

AWS CLI

刪除自訂動作

下列delete-action-target範例會刪除指定 ARN 識別的自訂動作。

```
aws securityhub delete-action-target \
    --action-target-arn "arn:aws:securityhub:us-west-1:123456789012:action/custom/
Remediation"
```

輸出：

```
{
    "ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/custom/
Remediation"
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的建立自訂動作並將其與 CloudWatch Events 規則建立關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteActionTarget。

delete-configuration-policy

以下程式碼範例顯示如何使用 delete-configuration-policy。

AWS CLI

刪除組態政策

下列delete-configuration-policy範例會刪除指定的組態政策。

```
aws securityhub delete-configuration-policy \
```

```
    --identifier "arn:aws:securityhub:eu-central-1:123456789012:configuration-
policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Security Hub 使用者指南》中的刪除和取消 Security Hub 組態政策的關聯。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteConfigurationPolicy。

delete-finding-aggregator

以下程式碼範例顯示如何使用 delete-finding-aggregator。

AWS CLI

停止問題清單彙總

下列delete-finding-aggregator範例會停止尋找彙總。它從美國東部（維吉尼亞）執行，這是彙總區域。

```
aws securityhub delete-finding-aggregator \
    --region us-east-1 \
    --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-
aggregator/123e4567-e89b-12d3-a456-426652340000
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的停止問題清單彙總。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFindingAggregator。

delete-insight

以下程式碼範例顯示如何使用 delete-insight。

AWS CLI

刪除自訂洞見

下列delete-insight範例會刪除具有指定 ARN 的自訂洞見。

```
aws securityhub delete-insight \
```

```
    --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

輸出：

```
{
   "InsightArn": "arn:aws:securityhub:eu-
central-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理自訂洞見。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteInsight。

delete-invitations

以下程式碼範例顯示如何使用 delete-invitations。

AWS CLI

刪除成為成員帳戶的邀請

下列delete-invitations範例會刪除邀請，以成為指定管理員帳戶的成員帳戶。成員帳戶是請求帳戶。

```
aws securityhub delete-invitations \
    --account-ids "123456789012"
```

輸出：

```
{
    "UnprocessedAccounts": []
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理管理員和成員帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteInvitations。

delete-members

以下程式碼範例顯示如何使用 delete-members。

AWS CLI

刪除成員帳戶

下列delete-members範例會從請求管理員帳戶刪除指定的成員帳戶。

```
aws securityhub delete-members \
    --account-ids "123456789111" "123456789222"
```

輸出：

```
{
    "UnprocessedAccounts": []
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理管理員和成員帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteMembers。

describe-action-targets

以下程式碼範例顯示如何使用 describe-action-targets。

AWS CLI

擷取自訂動作的詳細資訊

下列describe-action-targets範例會擷取指定 ARN 所識別之自訂動作的相關資訊。

```
aws securityhub describe-action-targets \
    --action-target-arns "arn:aws:securityhub:us-west-1:123456789012:action/custom/
Remediation"
```

輸出：

```
{
    "ActionTargets": [
        {
            "ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/
custom/Remediation",
            "Description": "Action to send the finding for remediation tracking",
            "Name": "Send to remediation"
        }
```

```
    ]
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的建立自訂動作並將其與 CloudWatch Events 規則建立關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeActionTargets。

describe-hub

以下程式碼範例顯示如何使用 describe-hub。

AWS CLI

取得中樞資源的相關資訊

下列describe-hub範例會傳回指定中樞資源的訂閱日期。中樞資源由其 ARN 識別。

```
aws securityhub describe-hub \
    --hub-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default"
```

輸出：

```
{
    "HubArn": "arn:aws:securityhub:us-west-1:123456789012:hub/default",
    "SubscribedAt": "2019-11-19T23:15:10.046Z"
}
```

如需詳細資訊，請參閱 AWS CloudFormation 使用者指南中的 AWS：：SecurityHub：：Hub。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeHub。

describe-organization-configuration

以下程式碼範例顯示如何使用 describe-organization-configuration。

AWS CLI

檢視如何為組織設定 Security Hub

下列describe-organization-configuration範例會傳回有關組織在 Security Hub 中設定方式的資訊。在此範例中，組織使用中央組態。只有 Security Hub 管理員帳戶可以執行此命令。

```
aws securityhub describe-organization-configuration
```

輸出：

```
{
    "AutoEnable": false,
    "MemberAccountLimitReached": false,
    "AutoEnableStandards": "NONE",
    "OrganizationConfiguration": {
        "ConfigurationType": "LOCAL",
        "Status": "ENABLED",
        "StatusMessage": "Central configuration has been enabled successfully"
    }
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的使用 AWS Organizations 管理帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeOrganizationConfiguration。

describe-products

以下程式碼範例顯示如何使用 describe-products。

AWS CLI

傳回可用產品整合的相關資訊

下列describe-products範例一次傳回一個可用的產品整合。

```
aws securityhub describe-products \
    --max-results 1
```

輸出：

```
{
    "NextToken": "U2FsdGVkX18vvPlOqb7RDrWRWVFBJI46MOIAb+nZmRJmR15NoRi2gm13sdQEn3O/
pq/78dGs+bKpgA+7HMPHO0qX33/zoRI+uIG/F9yLNhcOrOWzFUdy36JcXLQji3Rpnn/
cD1SVkGA98qI3zPOSDg==",
    "Products": [
        {
            "ProductArn": "arn:aws:securityhub:us-west-1:123456789333:product/
crowdstrike/crowdstrike-falcon",
```

```
            "ProductName": "CrowdStrike Falcon",
            "CompanyName": "CrowdStrike",
            "Description": "CrowdStrike Falcon's single lightweight sensor unifies
 next-gen antivirus, endpoint detection and response, and 24/7 managed hunting, via
 the cloud.",
            "Categories": [
                "Endpoint Detection and Response (EDR)",
                "AV Scanning and Sandboxing",
                "Threat Intelligence Feeds and Reports",
                "Endpoint Forensics",
                "Network Forensics"
            ],
            "IntegrationTypes": [
                "SEND_FINDINGS_TO_SECURITY_HUB"
            ],
            "MarketplaceUrl": "https://aws.amazon.com/marketplace/seller-profile?
id=a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "ActivationUrl": "https://falcon.crowdstrike.com/support/documentation",
            "ProductSubscriptionResourcePolicy": "{\"Version\":
\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":
\"123456789333\"},\"Action\":[\"securityhub:BatchImportFindings\"],\"Resource\":
\"arn:aws:securityhub:us-west-1:123456789012:product-subscription/crowdstrike/
crowdstrike-falcon\",\"Condition\":{\"StringEquals\":{\"securityhub:TargetAccount
\":\"123456789012\"}}},{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":
\"123456789012\"},\"Action\":[\"securityhub:BatchImportFindings\"],\"Resource
\":\"arn:aws:securityhub:us-west-1:123456789333:product/crowdstrike/crowdstrike-
falcon\",\"Condition\":{\"StringEquals\":{\"securityhub:TargetAccount\":
\"123456789012\"}}}]}"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理產品整合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeProducts。

describe-standards-controls

以下程式碼範例顯示如何使用 describe-standards-controls。

AWS CLI

請求已啟用標準中的控制項清單

下列describe-standards-controls範例會請求請求者帳戶訂閱 PCI DSS 標準的控制項清單。請求一次會傳回兩個控制項。

```
aws securityhub describe-standards-controls \
    --standards-subscription-arn "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1" \
    --max-results 2
```

輸出：

```
{
    "Controls": [
        {
            "StandardsControlArn": "arn:aws:securityhub:us-
west-1:123456789012:control/pci-dss/v/3.2.1/PCI.AutoScaling.1",
            "ControlStatus": "ENABLED",
            "ControlStatusUpdatedAt": "2020-05-15T18:49:04.473000+00:00",
            "ControlId": "PCI.AutoScaling.1",
            "Title": "Auto scaling groups associated with a load balancer should use
 health checks",
            "Description": "This AWS control checks whether your Auto Scaling groups
 that are associated with a load balancer are using Elastic Load Balancing health
 checks.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
PCI.AutoScaling.1/remediation",
            "SeverityRating": "LOW",
            "RelatedRequirements": [
                "PCI DSS 2.2"
            ]
        },
        {
            "StandardsControlArn": "arn:aws:securityhub:us-
west-1:123456789012:control/pci-dss/v/3.2.1/PCI.CW.1",
            "ControlStatus": "ENABLED",
            "ControlStatusUpdatedAt": "2020-05-15T18:49:04.498000+00:00",
            "ControlId": "PCI.CW.1",
            "Title": "A log metric filter and alarm should exist for usage of the
 \"root\" user",
            "Description": "This control checks for the CloudWatch metric
 filters using the following pattern { $.userIdentity.type = \"Root\" &&
 $.userIdentity.invokedBy NOT EXISTS && $.eventType != \"AwsServiceEvent\" }
 It checks that the log group name is configured for use with active multi-
region CloudTrail, that there is at least one Event Selector for a Trail with
```

```
 IncludeManagementEvents set to true and ReadWriteType set to All, and that there is
 at least one active subscriber to an SNS topic associated with the alarm.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
PCI.CW.1/remediation",
            "SeverityRating": "MEDIUM",
            "RelatedRequirements": [
                "PCI DSS 7.2.1"
            ]
        }
    ],
    "NextToken": "U2FsdGVkX1+eNkPoZHVl11ip5HUYQPWSWZGmftcmJiHL8JoKEsCDuaKayiPDyLK
+LiTkShveoOdvfxXCkOBaGhohIXhsIedN+LSjQV/
l7kfCfJcq4PziNC1N9xe9aq2pjlLVZnznTfSImrodT5bRNHe4fELCQq/z+5ka
+5Lzmc11axcwTd5lKgQyQqmUVoeriHZhyIiBgWKf7oNYdBVG8OEortVWvSkoUTt
+B2ThcnC7l43kI0UNxlkZ6sc64AsW"
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的檢視控制項的詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStandardsControls。

describe-standards

以下程式碼範例顯示如何使用 describe-standards。

AWS CLI

傳回可用標準的清單

下列describe-standards範例會傳回可用標準的清單。

```
aws securityhub describe-standards
```

輸出：

```
{
    "Standards": [
        {
            "StandardsArn": "arn:aws:securityhub:us-west-1::standards/aws-
foundational-security-best-practices/v/1.0.0",
            "Name": "AWS Foundational Security Best Practices v1.0.0",
            "Description": "The AWS Foundational Security Best Practices standard
 is a set of automated security checks that detect when AWS accounts and deployed
```

```
 resources do not align to security best practices. The standard is defined by AWS
 security experts. This curated set of controls helps improve your security posture
 in AWS, and cover AWS's most popular and foundational services.",
            "EnabledByDefault": true
        },
        {
            "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-
benchmark/v/1.2.0",
            "Name": "CIS AWS Foundations Benchmark v1.2.0",
            "Description": "The Center for Internet Security (CIS) AWS Foundations
 Benchmark v1.2.0 is a set of security configuration best practices for AWS. This
 Security Hub standard automatically checks for your compliance readiness against a
 subset of CIS requirements.",
            "EnabledByDefault": true
        },
        {
            "StandardsArn": "arn:aws:securityhub:us-west-1::standards/pci-dss/
v/3.2.1",
            "Name": "PCI DSS v3.2.1",
            "Description": "The Payment Card Industry Data Security Standard (PCI
 DSS) v3.2.1 is an information security standard for entities that store, process,
 and/or transmit cardholder data. This Security Hub standard automatically checks
 for your compliance readiness against a subset of PCI DSS requirements.",
            "EnabledByDefault": false
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Security Hub AWS 使用者指南中的 Security Hub 中的安全標準。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeStandards。

`disable-import-findings-for-product`

以下程式碼範例顯示如何使用 `disable-import-findings-for-product`。

AWS CLI

從產品整合停止接收問題清單

下列`disable-import-findings-for-product`範例會停用產品整合之指定訂閱的調查結果流程。

```
aws securityhub disable-import-findings-for-product \
    --product-subscription-arn "arn:aws:securityhub:us-west-1:123456789012:product-subscription/crowdstrike/crowdstrike-falcon"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理產品整合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableImportFindingsForProduct。

disable-organization-admin-account

以下程式碼範例顯示如何使用 disable-organization-admin-account。

AWS CLI

移除 Security Hub 管理員帳戶

下列disable-organization-admin-account範例會撤銷指定帳戶的指派，做為 AWS Organizations 的 Security Hub 管理員帳戶。

```
aws securityhub disable-organization-admin-account \
    --admin-account-id 777788889999
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Security Hub 使用者指南中的指定 Security Hub 管理員帳戶。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableOrganizationAdminAccount。

disable-security-hub

以下程式碼範例顯示如何使用 disable-security-hub。

AWS CLI

停用 AWS Security Hub

下列disable-security-hub範例會停用請求帳戶的 AWS Security Hub。

```
aws securityhub disable-security-hub
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Security Hub 使用者指南》中的停用 AWS Security Hub。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisableSecurityHub。

disassociate-from-administrator-account

以下程式碼範例顯示如何使用 disassociate-from-administrator-account。

AWS CLI

取消與管理員帳戶的關聯

下列disassociate-from-administrator-account範例會取消請求帳戶與其目前管理員帳戶的關聯。

```
aws securityhub disassociate-from-administrator-account
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理管理員和成員帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateFromAdministratorAccount。

disassociate-from-master-account

以下程式碼範例顯示如何使用 disassociate-from-master-account。

AWS CLI

取消與管理員帳戶的關聯

下列disassociate-from-master-account範例會取消請求帳戶與其目前管理員帳戶的關聯。

```
aws securityhub disassociate-from-master-account
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理管理員和成員帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateFromMasterAccount。

disassociate-members

以下程式碼範例顯示如何使用 disassociate-members。

AWS CLI

取消成員帳戶的關聯

下列disassociate-members範例會取消指定成員帳戶與請求管理員帳戶的關聯。

```
aws securityhub disassociate-members  \
    --account-ids "123456789111" "123456789222"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理管理員和成員帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateMembers。

enable-import-findings-for-product

以下程式碼範例顯示如何使用 enable-import-findings-for-product。

AWS CLI

從產品整合開始接收問題清單

下列enable-import-findings-for-product範例會啟用來自指定產品整合的問題清單流程。

```
aws securityhub enable-import-findings-for-product \
    --product-arn "arn:aws:securityhub:us-east-1:123456789333:product/crowdstrike/
crowdstrike-falcon"
```

輸出：

```
{
    "ProductSubscriptionArn": "arn:aws:securityhub:us-east-1:123456789012:product-
subscription/crowdstrike/crowdstrike-falcon"
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理產品整合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableImportFindingsForProduct。

enable-organization-admin-account

以下程式碼範例顯示如何使用 enable-organization-admin-account。

AWS CLI

將組織帳戶指定為 Security Hub 管理員帳戶

下列enable-organization-admin-account範例會將指定的帳戶指定為 Security Hub 管理員帳戶。

```
aws securityhub enable-organization-admin-account \
    --admin-account-id 777788889999
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Security Hub 使用者指南中的指定 Security Hub 管理員帳戶。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableOrganizationAdminAccount。

enable-security-hub

以下程式碼範例顯示如何使用 enable-security-hub。

AWS CLI

啟用 AWS Security Hub

下列enable-security-hub範例會為請求帳戶啟用 AWS Security Hub。它會設定 Security Hub 以啟用預設標準。對於中樞資源，它會將值指派給Security標籤 Department。

```
aws securityhub enable-security-hub \
    --enable-default-standards \
    --tags '{"Department": "Security"}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Security Hub 使用者指南中的啟用 AWS Security Hub。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 EnableSecurityHub。

get-administrator-account

以下程式碼範例顯示如何使用 get-administrator-account。

AWS CLI

擷取管理員帳戶的相關資訊

下列get-administrator-account範例會擷取請求帳戶管理員帳戶的相關資訊。

```
aws securityhub get-administrator-account
```

輸出：

```
{
   "Master": {
      "AccountId": "123456789012",
      "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
      "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
      "MemberStatus": "ASSOCIATED"
   }
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理管理員和成員帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAdministratorAccount。

get-configuration-policy-association

以下程式碼範例顯示如何使用 get-configuration-policy-association。

AWS CLI

取得目標的組態關聯詳細資訊

下列get-configuration-policy-association範例會擷取指定目標的關聯詳細資訊。您可以提供帳戶 ID、組織單位 ID 或目標的根 ID。

```
aws securityhub get-configuration-policy-association \
    --target '{"OrganizationalUnitId": "ou-6hi7-8j91kl2m"}'
```

輸出：

```
{
    "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "TargetId": "ou-6hi7-8j91kl2m",
    "TargetType": "ORGANIZATIONAL_UNIT",
    "AssociationType": "APPLIED",
    "UpdatedAt": "2023-09-26T21:13:01.816000+00:00",
    "AssociationStatus": "SUCCESS",
    "AssociationStatusMessage": "Association applied successfully on this target."
}
```

如需詳細資訊，請參閱 Security Hub 使用者指南中的檢視 Security Hub 組態政策。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetConfigurationPolicyAssociation。

get-configuration-policy

以下程式碼範例顯示如何使用 get-configuration-policy。

AWS CLI

檢視組態政策詳細資訊

下列get-configuration-policy範例會擷取指定組態政策的詳細資訊。

```
aws securityhub get-configuration-policy \
   --identifier "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

輸出：

```
{
    "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Id": "ce5ed1e7-9639-4e2f-9313-fa87fcef944b",
    "Name": "SampleConfigurationPolicy",
    "Description": "SampleDescription",
    "UpdatedAt": "2023-11-28T20:28:04.494000+00:00",
    "CreatedAt": "2023-11-28T20:28:04.494000+00:00",
    "ConfigurationPolicy": {
        "SecurityHub": {
            "ServiceEnabled": true,
```

```
            "EnabledStandardIdentifiers": [
                "arn:aws:securityhub:eu-central-1::standards/aws-foundational-
security-best-practices/v/1.0.0",
                "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"
            ],
            "SecurityControlsConfiguration": {
                "DisabledSecurityControlIdentifiers": [
                    "CloudTrail.2"
                ],
                "SecurityControlCustomParameters": [
                    {
                        "SecurityControlId": "ACM.1",
                        "Parameters": {
                            "daysToExpiration": {
                                "ValueType": "CUSTOM",
                                "Value": {
                                    "Integer": 15
                                }
                            }
                        }
                    }
                ]
            }
        }
    }
}
```

如需詳細資訊，請參閱 Security Hub 使用者指南中的檢視 Security Hub 組態政策。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetConfigurationPolicy。

get-enabled-standards

以下程式碼範例顯示如何使用 get-enabled-standards。

AWS CLI

擷取已啟用標準的相關資訊

下列get-enabled-standards範例會擷取 PCI DSS 標準的相關資訊。

```
aws securityhub get-enabled-standards \
```

```
    --standards-subscription-arn "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"
```

輸出：

```
{
    "StandardsSubscriptions": [
        {
            "StandardsArn": "arn:aws:securityhub:us-west-1::standards/pci-dss/
v/3.2.1",
            "StandardsInput": { },
            "StandardsStatus": "READY",
            "StandardsSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Security Hub AWS 使用者指南中的 Security Hub 中的安全標準。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetEnabledStandards。

get-finding-aggregator

以下程式碼範例顯示如何使用 get-finding-aggregator。

AWS CLI

擷取目前的調查結果彙總組態

下列get-finding-aggregator範例會擷取目前的調查結果彙總組態。

```
aws securityhub get-finding-aggregator \
    --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-
aggregator/123e4567-e89b-12d3-a456-426652340000
```

輸出：

```
{
    "FindingAggregatorArn": "arn:aws:securityhub:us-east-1:222222222222:finding-
aggregator/123e4567-e89b-12d3-a456-426652340000",
    "FindingAggregationRegion": "us-east-1",
    "RegionLinkingMode": "SPECIFIED_REGIONS",
```

```
    "Regions": "us-west-1,us-west-2"
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的檢視目前的調查結果彙總組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFindingAggregator。

get-finding-history

以下程式碼範例顯示如何使用 get-finding-history。

AWS CLI

若要取得問題清單歷史記錄

下列get-finding-history範例會取得指定問題清單過去 90 天的歷史記錄。在此範例中，結果僅限於兩個調查結果歷史記錄的記錄。

```
aws securityhub get-finding-history \
    --finding-identifier Id="arn:aws:securityhub:us-
east-1:123456789012:security-control/S3.17/finding/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",ProductArn="arn:aws:securityhub:us-east-1::product/aws/securityhub"
```

輸出：

```
{
    "Records": [
        {
            "FindingIdentifier": {
                "Id": "arn:aws:securityhub:us-east-1:123456789012:security-control/
S3.17/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
                "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/
securityhub"
            },
            "UpdateTime": "2023-06-02T03:15:25.685000+00:00",
            "FindingCreated": false,
            "UpdateSource": {
                "Type": "BATCH_IMPORT_FINDINGS",
                "Identity": "arn:aws:securityhub:us-east-1::product/aws/securityhub"
            },
            "Updates": [
                {
                    "UpdatedField": "Compliance.RelatedRequirements",
```

```
                    "OldValue": "[\"NIST.800-53.r5 SC-12(2)\",\"NIST.800-53.r5
 SC-12(3)\",\"NIST.800-53.r5 SC-12(6)\",\"NIST.800-53.r5 CM-3(6)\",\"NIST.800-53.r5
 SC-13\",\"NIST.800-53.r5 SC-28\",\"NIST.800-53.r5 SC-28(1)\",\"NIST.800-53.r5
 SC-7(10)\"]",
                    "NewValue": "[\"NIST.800-53.r5 SC-12(2)\",\"NIST.800-53.r5
 CM-3(6)\",\"NIST.800-53.r5 SC-13\",\"NIST.800-53.r5 SC-28\",\"NIST.800-53.r5
 SC-28(1)\",\"NIST.800-53.r5 SC-7(10)\",\"NIST.800-53.r5 CA-9(1)\",\"NIST.800-53.r5
 SI-7(6)\",\"NIST.800-53.r5 AU-9\"]"
                },
                {
                    "UpdatedField": "LastObservedAt",
                    "OldValue": "2023-06-01T09:15:38.587Z",
                    "NewValue": "2023-06-02T03:15:22.946Z"
                },
                {
                    "UpdatedField": "UpdatedAt",
                    "OldValue": "2023-06-01T09:15:31.049Z",
                    "NewValue": "2023-06-02T03:15:14.861Z"
                },
                {
                    "UpdatedField": "ProcessedAt",
                    "OldValue": "2023-06-01T09:15:41.058Z",
                    "NewValue": "2023-06-02T03:15:25.685Z"
                }
            ]
        },
        {
            "FindingIdentifier": {
                "Id": "arn:aws:securityhub:us-east-1:123456789012:security-control/
S3.17/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
                "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/
securityhub"
            },
            "UpdateTime": "2023-05-23T02:06:51.518000+00:00",
            "FindingCreated": "true",
            "UpdateSource": {
                "Type": "BATCH_IMPORT_FINDINGS",
                "Identity": "arn:aws:securityhub:us-east-1::product/aws/securityhub"
            },
            "Updates": []
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的尋找歷史記錄。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFindingHistory。

get-findings

以下程式碼範例顯示如何使用 get-findings。

AWS CLI

範例 1：傳回針對特定標準產生的調查結果

下列get-findings範例會傳回 PCI DSS 標準的調查結果。

```
aws securityhub get-findings \
    --filters '{"GeneratorId":[{"Value": "pci-dss","Comparison":"PREFIX"}]}' \
    --max-items 1
```

輸出：

```
{
    "Findings": [
        {
            "SchemaVersion": "2018-10-08",
            "Id": "arn:aws:securityhub:eu-central-1:123456789012:subscription/pci-
dss/v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub",
            "GeneratorId": "pci-dss/v/3.2.1/PCI.Lambda.2",
            "AwsAccountId": "123456789012",
            "Types": [
                "Software and Configuration Checks/Industry and Regulatory
 Standards/PCI-DSS"
            ],
            "FindingProviderFields": {
                "Severity": {
                    "Original": 0,
                    "Label": "INFORMATIONAL"
                },
                "Types": [
                    "Software and Configuration Checks/Industry and Regulatory
 Standards/PCI-DSS"
                ]
            },
```

```
            "FirstObservedAt": "2020-06-02T14:02:49.159Z",
            "LastObservedAt": "2020-06-02T14:02:52.397Z",
            "CreatedAt": "2020-06-02T14:02:49.159Z",
            "UpdatedAt": "2020-06-02T14:02:52.397Z",
            "Severity": {
                "Original": 0,
                "Label": "INFORMATIONAL",
                "Normalized": 0
            },
            "Title": "PCI.Lambda.2 Lambda functions should be in a VPC",
            "Description": "This AWS control checks whether a Lambda function is in
 a VPC.",
            "Remediation": {
                "Recommendation": {
                    "Text": "For directions on how to fix this issue, please consult
 the AWS Security Hub PCI DSS documentation.",
                    "Url": "https://docs.aws.amazon.com/console/securityhub/
PCI.Lambda.2/remediation"
                }
            },
            "ProductFields": {
                "StandardsArn": "arn:aws:securityhub:::standards/pci-dss/v/3.2.1",
                "StandardsSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1",
                "ControlId": "PCI.Lambda.2",
                "RecommendationUrl": "https://docs.aws.amazon.com/console/
securityhub/PCI.Lambda.2/remediation",
                "RelatedAWSResources:0/name": "securityhub-lambda-inside-
vpc-0e904a3b",
                "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
                "StandardsControlArn": "arn:aws:securityhub:us-
west-1:123456789012:control/pci-dss/v/3.2.1/PCI.Lambda.2",
                "aws/securityhub/SeverityLabel": "INFORMATIONAL",
                "aws/securityhub/ProductName": "Security Hub",
                "aws/securityhub/CompanyName": "AWS",
                "aws/securityhub/FindingId": "arn:aws:securityhub:eu-
central-1::product/aws/securityhub/arn:aws:securityhub:eu-
central-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        },
            "Resources": [
                {
                    "Type": "AwsAccount",
                    "Id": "AWS::::Account:123456789012",
```

```
                    "Partition": "aws",
                    "Region": "us-west-1"
                }
            ],
            "Compliance": {
                "Status": "PASSED",
                "RelatedRequirements": [
                    "PCI DSS 1.2.1",
                    "PCI DSS 1.3.1",
                    "PCI DSS 1.3.2",
                    "PCI DSS 1.3.4"
                ]
            },
            "WorkflowState": "NEW",
            "Workflow": {
                "Status": "NEW"
            },
            "RecordState": "ARCHIVED"
        }
    ],
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ=="
}
```

範例 2：傳回工作流程狀態為 NOTIFIED 的關鍵嚴重性調查結果

下列get-findings範例會傳回嚴重性標籤值為 CRITICAL 且工作流程狀態為 NOTIFIED 的問題清單。結果會依可信度值以遞減順序排序。

```
aws securityhub get-findings \
    --filters '{"SeverityLabel":[{"Value":
 "CRITICAL","Comparison":"EQUALS"}],"WorkflowStatus":
 [{"Value":"NOTIFIED","Comparison":"EQUALS"}]}' \
    --sort-criteria '{ "Field": "Confidence", "SortOrder": "desc"}' \
    --max-items 1
```

輸出：

```
{
    "Findings": [
        {
            "SchemaVersion": "2018-10-08",
            "Id": "arn:aws:securityhub:us-west-1: 123456789012:subscription/cis-aws-
foundations-benchmark/v/1.2.0/1.13/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```
            "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/securityhub",
            "GeneratorId": "arn:aws:securityhub:::ruleset/cis-aws-foundations-
benchmark/v/1.2.0/rule/1.13",
            "AwsAccountId": "123456789012",
            "Types": [
                "Software and Configuration Checks/Industry and Regulatory
 Standards/CIS AWS Foundations Benchmark"
            ],
            "FindingProviderFields" {
                "Severity": {
                    "Original": 90,
                    "Label": "CRITICAL"
                },
                "Types": [
                    "Software and Configuration Checks/Industry and Regulatory
 Standards/CIS AWS Foundations Benchmark"
                ]
            },
            "FirstObservedAt": "2020-05-21T20:16:34.752Z",
            "LastObservedAt": "2020-06-09T08:16:37.171Z",
            "CreatedAt": "2020-05-21T20:16:34.752Z",
            "UpdatedAt": "2020-06-09T08:16:36.430Z",
            "Severity": {
                "Original": 90,
                "Label": "CRITICAL",
                "Normalized": 90
            },
            "Title": "1.13 Ensure MFA is enabled for the \"root\" account",
            "Description": "The root account is the most privileged user in an AWS
 account. MFA adds an extra layer of protection on top of a user name and password.
 With MFA enabled, when a user signs in to an AWS website, they will be prompted for
 their user name and password as well as for an authentication code from their AWS
 MFA device.",
            "Remediation": {
                "Recommendation": {
                    "Text": "For directions on how to fix this issue, please consult
 the AWS Security Hub CIS documentation.",
                    "Url": "https://docs.aws.amazon.com/console/securityhub/
standards-cis-1.13/remediation"
                }
            },
            "ProductFields": {
                "StandardsGuideArn": "arn:aws:securityhub:::ruleset/cis-aws-
foundations-benchmark/v/1.2.0",
```

```
                "StandardsGuideSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/cis-aws-foundations-benchmark/v/1.2.0",
                "RuleId": "1.13",
                "RecommendationUrl": "https://docs.aws.amazon.com/console/
securityhub/standards-cis-1.13/remediation",
                "RelatedAWSResources:0/name": "securityhub-root-account-mfa-
enabled-5pftha",
                "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
                "StandardsControlArn": "arn:aws:securityhub:us-
west-1:123456789012:control/cis-aws-foundations-benchmark/v/1.2.0/1.13",
                "aws/securityhub/SeverityLabel": "CRITICAL",
                "aws/securityhub/ProductName": "Security Hub",
                "aws/securityhub/CompanyName": "AWS",
                "aws/securityhub/FindingId": "arn:aws:securityhub:us-
west-1::product/aws/securityhub/arn:aws:securityhub:us-
west-1:123456789012:subscription/cis-aws-foundations-benchmark/v/1.2.0/1.13/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
            },
            "Resources": [
                {
                    "Type": "AwsAccount",
                    "Id": "AWS::::Account:123456789012",
                    "Partition": "aws",
                    "Region": "us-west-1"
                }
            ],
            "Compliance": {
                "Status": "FAILED"
            },
            "WorkflowState": "NEW",
            "Workflow": {
                "Status": "NOTIFIED"
            },
            "RecordState": "ACTIVE"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的篩選和分組問題清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFindings。

get-insight-results

以下程式碼範例顯示如何使用 get-insight-results。

AWS CLI

擷取洞見的結果

下列get-insight-results範例會傳回具有指定 ARN 之洞見的洞見結果清單。

```
aws securityhub get-insight-results \
    --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

輸出：

```
{
    "InsightResults": {
        "GroupByAttribute": "ResourceId",
        "InsightArn": "arn:aws:securityhub:us-
west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",
        "ResultValues": [
            {
                "Count": 10,
                "GroupByAttributeValue": "AWS::::Account:123456789111"
            },
            {
                "Count": 3,
                "GroupByAttributeValue": "AWS::::Account:123456789222"
            }
        ]
    }
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的檢視洞察結果和調查結果並對其採取行動。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInsightResults。

get-insights

以下程式碼範例顯示如何使用 get-insights。

AWS CLI

擷取洞見的詳細資訊

下列get-insights範例會使用指定的 ARN 擷取洞見的組態詳細資訊。

```
aws securityhub get-insights \
    --insight-arns "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

輸出：

```
{
    "Insights": [
        {
            "Filters": {
               "ResourceType": [
                    {
                        "Comparison": "EQUALS",
                        "Value": "AwsIamRole"
                    }
                ],
                "SeverityLabel": [
                    {
                        "Comparison": "EQUALS",
                        "Value": "CRITICAL"
                    }
                ],
            },
            "GroupByAttribute": "ResourceId",
            "InsightArn": "arn:aws:securityhub:us-
west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",
            "Name": "Critical role findings"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的 Security Hub 中的 Insights。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInsights。

get-invitations-count

以下程式碼範例顯示如何使用 get-invitations-count。

AWS CLI

擷取未接受的邀請數量

下列get-invitations-count範例會擷取請求帳戶拒絕或未回應的邀請數量。

```
aws securityhub get-invitations-count
```

輸出：

```
{
  "InvitationsCount": 3
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理管理員和成員帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetInvitationsCount。

get-master-account

以下程式碼範例顯示如何使用 get-master-account。

AWS CLI

擷取管理員帳戶的相關資訊

下列get-master-account範例會擷取請求帳戶管理員帳戶的相關資訊。

```
aws securityhub get-master-account
```

輸出：

```
{
   "Master": {
      "AccountId": "123456789012",
      "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
      "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
```

```
        "MemberStatus": "ASSOCIATED"
    }
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理管理員和成員帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMasterAccount。

get-members

以下程式碼範例顯示如何使用 get-members。

AWS CLI

擷取所選成員帳戶的相關資訊

下列get-members範例會擷取指定成員帳戶的相關資訊。

```
aws securityhub get-members \
    --account-ids "444455556666" "777788889999"
```

輸出：

```
{
    "Members": [
        {
            "AccountId": "123456789111",
            "AdministratorId": "123456789012",
            "InvitedAt": 2020-06-01T20:15:15.289000+00:00,
            "MasterId": "123456789012",
            "MemberStatus": "ASSOCIATED",
            "UpdatedAt": 2020-06-01T20:15:15.289000+00:00
        },
        {
            "AccountId": "123456789222",
            "AdministratorId": "123456789012",
            "InvitedAt": 2020-06-01T20:15:15.289000+00:00,
            "MasterId": "123456789012",
            "MemberStatus": "ASSOCIATED",
            "UpdatedAt": 2020-06-01T20:15:15.289000+00:00
        }
    ],
```

```
    "UnprocessedAccounts": [ ]
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理管理員和成員帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMembers。

get-security-control-definition

以下程式碼範例顯示如何使用 get-security-control-definition。

AWS CLI

取得安全控制定義詳細資訊

下列get-security-control-definition範例會擷取 Security Hub 安全性控制項的定義詳細資訊。詳細資訊包括控制項標題、描述、區域可用性、參數和其他資訊。

```
aws securityhub get-security-control-definition \
    --security-control-id ACM.1
```

輸出：

```
{
    "SecurityControlDefinition": {
        "SecurityControlId": "ACM.1",
        "Title": "Imported and ACM-issued certificates should be renewed after a
 specified time period",
        "Description": "This control checks whether an AWS Certificate Manager
 (ACM) certificate is renewed within the specified time period. It checks both
 imported certificates and certificates provided by ACM. The control fails if the
 certificate isn't renewed within the specified time period. Unless you provide a
 custom parameter value for the renewal period, Security Hub uses a default value of
 30 days.",
        "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/ACM.1/
remediation",
        "SeverityRating": "MEDIUM",
        "CurrentRegionAvailability": "AVAILABLE",
        "ParameterDefinitions": {
            "daysToExpiration": {
                "Description": "Number of days within which the ACM certificate must
 be renewed",
```

```
                "ConfigurationOptions": {
                    "Integer": {
                        "DefaultValue": 30,
                        "Min": 14,
                        "Max": 365
                    }
                }
            }
        }
    }
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的自訂控制參數。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSecurityControlDefinition。

`invite-members`

以下程式碼範例顯示如何使用 `invite-members`。

AWS CLI

傳送邀請給成員帳戶

下列`invite-members`範例會將邀請傳送至指定的成員帳戶。

```
aws securityhub invite-members \
    --account-ids "123456789111" "123456789222"
```

輸出：

```
{
    "UnprocessedAccounts": []
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理管理員和成員帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 InviteMembers。

`list-automation-rules`

以下程式碼範例顯示如何使用 `list-automation-rules`。

AWS CLI

檢視自動化規則清單

下列list-automation-rules範例列出 AWS 帳戶的自動化規則。只有 Security Hub 管理員帳戶可以執行此命令。

```
aws securityhub list-automation-rules \
    --max-results 3 \
    --next-token NULL
```

輸出：

```
{
    "AutomationRulesMetadata": [
        {
            "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "RuleStatus": "ENABLED",
            "RuleOrder": 1,
            "RuleName": "Suppress informational findings",
            "Description": "Suppress GuardDuty findings with Informational
 severity",
            "IsTerminal": false,
            "CreatedAt": "2023-05-31T17:56:14.837000+00:00",
            "UpdatedAt": "2023-05-31T17:59:38.466000+00:00",
            "CreatedBy": "arn:aws:iam::123456789012:role/Admin"
        },
        {
            "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "RuleStatus": "ENABLED",
            "RuleOrder": 1,
            "RuleName": "sample rule",
            "Description": "A sample rule",
            "IsTerminal": false,
            "CreatedAt": "2023-07-15T23:37:20.223000+00:00",
            "UpdatedAt": "2023-07-15T23:37:20.223000+00:00",
            "CreatedBy": "arn:aws:iam::123456789012:role/Admin"
        },
        {
            "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
```

```
            "RuleStatus": "ENABLED",
            "RuleOrder": 1,
            "RuleName": "sample rule",
            "Description": "A sample rule",
            "IsTerminal": false,
            "CreatedAt": "2023-07-15T23:45:25.126000+00:00",
            "UpdatedAt": "2023-07-15T23:45:25.126000+00:00",
            "CreatedBy": "arn:aws:iam::123456789012:role/Admin"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的檢視自動化規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAutomationRules。

`list-configuration-policies`

以下程式碼範例顯示如何使用 `list-configuration-policies`。

AWS CLI

列出組態政策摘要

下列`list-configuration-policies`範例列出組織的組態政策摘要。

```
aws securityhub list-configuration-policies \
    --max-items 3
```

輸出：

```
{
    "ConfigurationPolicySummaries": [
        {
            "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-
policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Name": "SampleConfigurationPolicy1",
            "Description": "SampleDescription1",
            "UpdatedAt": "2023-09-26T21:08:36.214000+00:00",
            "ServiceEnabled": true
        },
```

```
        {
            "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-
policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "Name": "SampleConfigurationPolicy2",
            "Description": "SampleDescription2"
            "UpdatedAt": "2023-11-28T19:26:25.207000+00:00",
            "ServiceEnabled": true
        },
        {
            "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-
policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "Name": "SampleConfigurationPolicy3",
            "Description": "SampleDescription3",
            "UpdatedAt": "2023-11-28T20:28:04.494000+00:00",
            "ServiceEnabled": true
        }
}
```

如需詳細資訊，請參閱 Security Hub 使用者指南中的檢視 Security Hub 組態政策。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListConfigurationPolicies。

`list-configuration-policy-associations`

以下程式碼範例顯示如何使用 `list-configuration-policy-associations`。

AWS CLI

列出組態關聯

下列`list-configuration-policy-associations`範例列出組織的組態關聯摘要。回應包括與組態政策和自我管理行為的關聯。

```
aws securityhub list-configuration-policy-associations \
    --filters '{"AssociationType": "APPLIED"}' \
    --max-items 4
```

輸出：

```
{
```

```
    "ConfigurationPolicyAssociationSummaries": [
        {
            "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "TargetId": "r-1ab2",
            "TargetType": "ROOT",
            "AssociationType": "APPLIED",
            "UpdatedAt": "2023-11-28T19:26:49.417000+00:00",
            "AssociationStatus": "FAILED",
            "AssociationStatusMessage": "Policy association failed because 2
 organizational units or accounts under this root failed."
        },
        {
            "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "TargetId": "ou-1ab2-c3de4f5g",
            "TargetType": "ORGANIZATIONAL_UNIT",
            "AssociationType": "APPLIED",
            "UpdatedAt": "2023-09-26T21:14:05.283000+00:00",
            "AssociationStatus": "FAILED",
            "AssociationStatusMessage": "One or more children under this target
 failed association."
        },
        {
            "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
            "TargetId": "ou-6hi7-8j91kl2m",
            "TargetType": "ORGANIZATIONAL_UNIT",
            "AssociationType": "APPLIED",
            "UpdatedAt": "2023-09-26T21:13:01.816000+00:00",
            "AssociationStatus": "SUCCESS",
            "AssociationStatusMessage": "Association applied successfully on this
 target."
        },
        {
            "ConfigurationPolicyId": "SELF_MANAGED_SECURITY_HUB",
            "TargetId": "111122223333",
            "TargetType": "ACCOUNT",
            "AssociationType": "APPLIED",
            "UpdatedAt": "2023-11-28T22:01:26.409000+00:00",
            "AssociationStatus": "SUCCESS"
    }
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的檢視組態政策狀態和詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListConfigurationPolicyAssociations。

list-enabled-products-for-import

以下程式碼範例顯示如何使用 list-enabled-products-for-import。

AWS CLI

傳回已啟用產品整合的清單

下列list-enabled-products-for-import範例會傳回目前啟用產品整合的訂閱 ARNS 清單。

```
aws securityhub list-enabled-products-for-import
```

輸出：

```
{
    "ProductSubscriptions": [ "arn:aws:securityhub:us-west-1:123456789012:product-subscription/crowdstrike/crowdstrike-falcon", "arn:aws:securityhub:us-west-1:123456789012:product-subscription/aws/securityhub" ]
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理產品整合。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEnabledProductsForImport。

list-finding-aggregators

以下程式碼範例顯示如何使用 list-finding-aggregators。

AWS CLI

列出可用的小工具

下列list-finding-aggregators範例會傳回調查結果彙總組態的 ARN。

```
aws securityhub list-finding-aggregators
```

輸出：

```
{
    "FindingAggregatorArn": "arn:aws:securityhub:us-east-1:222222222222:finding-aggregator/123e4567-e89b-12d3-a456-426652340000"
```

```
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的檢視目前的調查結果彙總組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFindingAggregators。

list-invitations

以下程式碼範例顯示如何使用 list-invitations。

AWS CLI

顯示邀請清單

下列list-invitations範例會擷取傳送至請求帳戶的邀請清單。

```
aws securityhub list-invitations
```

輸出：

```
{
    "Invitations": [
        {
            "AccountId": "123456789012",
            "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
            "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
            "MemberStatus": "ASSOCIATED"
        }
    ],
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理管理員和成員帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListInvitations。

list-members

以下程式碼範例顯示如何使用 list-members。

AWS CLI

擷取成員帳戶清單

下列list-members範例會傳回請求管理員帳戶的成員帳戶清單。

```
aws securityhub list-members
```

輸出：

```
{
    "Members": [
        {
            "AccountId": "123456789111",
            "AdministratorId": "123456789012",
            "InvitedAt": 2020-06-01T20:15:15.289000+00:00,
            "MasterId": "123456789012",
            "MemberStatus": "ASSOCIATED",
            "UpdatedAt": 2020-06-01T20:15:15.289000+00:00
        },
        {
            "AccountId": "123456789222",
            "AdministratorId": "123456789012",
            "InvitedAt": 2020-06-01T20:15:15.289000+00:00,
            "MasterId": "123456789012",
            "MemberStatus": "ASSOCIATED",
            "UpdatedAt": 2020-06-01T20:15:15.289000+00:00
        }
    ],
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理管理員和成員帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMembers。

list-organization-admin-accounts

以下程式碼範例顯示如何使用 list-organization-admin-accounts。

AWS CLI

列出指定的 Security Hub 管理員帳戶

下列list-organization-admin-accounts範例列出組織的 Security Hub 管理員帳戶。

```
aws securityhub list-organization-admin-accounts
```

輸出：

```
{
    AdminAccounts": [
        { "AccountId": "777788889999" },
        { "Status": "ENABLED" }
    ]
}
```

如需詳細資訊，請參閱 Security Hub 使用者指南中的指定 Security Hub 管理員帳戶。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListOrganizationAdminAccounts。

list-security-control-definitions

以下程式碼範例顯示如何使用 list-security-control-definitions。

AWS CLI

範例 1：列出所有可用的安全控制

下列list-security-control-definitions範例列出所有 Security Hub 標準的可用安全控制。此範例會將結果限制為三個控制項。

```
aws securityhub list-security-control-definitions \
    --max-items 3
```

輸出：

```
{
    "SecurityControlDefinitions": [
        {
            "SecurityControlId": "ACM.1",
            "Title": "Imported and ACM-issued certificates should be renewed after a
 specified time period",
            "Description": "This control checks whether an AWS Certificate Manager
 (ACM) certificate is renewed within the specified time period. It checks both
 imported certificates and certificates provided by ACM. The control fails if the
 certificate isn't renewed within the specified time period. Unless you provide a
 custom parameter value for the renewal period, Security Hub uses a default value of
 30 days.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
ACM.1/remediation",
```

```
            "SeverityRating": "MEDIUM",
            "CurrentRegionAvailability": "AVAILABLE",
            "CustomizableProperties": [
                "Parameters"
            ]
        },
        {
            "SecurityControlId": "ACM.2",
            "Title": "RSA certificates managed by ACM should use a key length of at
 least 2,048 bits",
            "Description": "This control checks whether RSA certificates managed by
 AWS Certificate Manager use a key length of at least 2,048 bits. The control fails
 if the key length is smaller than 2,048 bits.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
ACM.2/remediation",
            "SeverityRating": "HIGH",
            "CurrentRegionAvailability": "AVAILABLE",
            "CustomizableProperties": []
        },
        {
            "SecurityControlId": "APIGateway.1",
            "Title": "API Gateway REST and WebSocket API execution logging should be
 enabled",
            "Description": "This control checks whether all stages of an Amazon
 API Gateway REST or WebSocket API have logging enabled. The control fails if
 the 'loggingLevel' isn't 'ERROR' or 'INFO' for all stages of the API. Unless you
 provide custom parameter values to indicate that a specific log type should be
 enabled, Security Hub produces a passed finding if the logging level is either
 'ERROR' or 'INFO'.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
APIGateway.1/remediation",
            "SeverityRating": "MEDIUM",
            "CurrentRegionAvailability": "AVAILABLE",
            "CustomizableProperties": [
                "Parameters"
            ]
        }
    ],
    "NextToken": "U2FsdGVkX1/UprCPzxVbkDeHikDXbDxfgJZ1w2RG1XWsFPTMTIQPVE0m/
FduIGxS7ObRtAbaUt/8/RCQcg2PU0YXI20hH/GrhoOTgv+TSm0qvQVFhkJepWmqh
+NYawjocVBeos6xzn/8qnbF9IuwGg=="
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的檢視標準的詳細資訊。

範例 2：列出特定標準的可用安全控制

下列list-security-control-definitions範例列出 CIS AWS Foundations Benchmark 1.4.0 版的可用安全控制。此範例會將結果限制為三個控制項。

```
aws securityhub list-security-control-definitions \
    --standards-arn "arn:aws:securityhub:us-east-1::standards/cis-aws-foundations-benchmark/v/1.4.0" \
    --max-items 3
```

輸出：

```
{
    "SecurityControlDefinitions": [
        {
            "SecurityControlId": "CloudTrail.1",
            "Title": "CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events",
            "Description": "This AWS control checks that there is at least one multi-region AWS CloudTrail trail includes read and write management events.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.1/remediation",
            "SeverityRating": "HIGH",
            "CurrentRegionAvailability": "AVAILABLE",
            "CustomizableProperties": []
        },
        {
            "SecurityControlId": "CloudTrail.2",
            "Title": "CloudTrail should have encryption at-rest enabled",
            "Description": "This AWS control checks whether AWS CloudTrail is configured to use the server side encryption (SSE) AWS Key Management Service (AWS KMS) customer master key (CMK) encryption. The check will pass if the KmsKeyId is defined.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.2/remediation",
            "SeverityRating": "MEDIUM",
            "CurrentRegionAvailability": "AVAILABLE",
            "CustomizableProperties": []
        },
        {
            "SecurityControlId": "CloudTrail.4",
```

```
            "Title": "CloudTrail log file validation should be enabled",
            "Description": "This AWS control checks whether CloudTrail log file
 validation is enabled.",
            "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
CloudTrail.4/remediation",
            "SeverityRating": "MEDIUM",
            "CurrentRegionAvailability": "AVAILABLE",
            "CustomizableProperties": []
        }
    ],
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAzfQ=="
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的檢視標準的詳細資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSecurityControlDefinitions。

list-standards-control-associations

以下程式碼範例顯示如何使用 list-standards-control-associations。

AWS CLI

取得每個啟用標準中控制項的啟用狀態

下列list-standards-control-associations範例列出每個啟用標準中 CloudTrail.1 的啟用狀態。

```
aws securityhub list-standards-control-associations \
    --security-control-id CloudTrail.1
```

輸出：

```
{
    "StandardsControlAssociationSummaries": [
        {
            "StandardsArn": "arn:aws:securityhub:us-east-2::standards/nist-800-53/
v/5.0.0",
            "SecurityControlId": "CloudTrail.1",
            "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/CloudTrail.1",
            "AssociationStatus": "ENABLED",
            "RelatedRequirements": [
```

```
                "NIST.800-53.r5 AC-2(4)",
                "NIST.800-53.r5 AC-4(26)",
                "NIST.800-53.r5 AC-6(9)",
                "NIST.800-53.r5 AU-10",
                "NIST.800-53.r5 AU-12",
                "NIST.800-53.r5 AU-2",
                "NIST.800-53.r5 AU-3",
                "NIST.800-53.r5 AU-6(3)",
                "NIST.800-53.r5 AU-6(4)",
                "NIST.800-53.r5 AU-14(1)",
                "NIST.800-53.r5 CA-7",
                "NIST.800-53.r5 SC-7(9)",
                "NIST.800-53.r5 SI-3(8)",
                "NIST.800-53.r5 SI-4(20)",
                "NIST.800-53.r5 SI-7(8)",
                "NIST.800-53.r5 SA-8(22)"
            ],
            "UpdatedAt": "2023-05-15T17:52:21.304000+00:00",
            "StandardsControlTitle": "CloudTrail should be enabled and configured
 with at least one multi-Region trail that includes read and write management
 events",
            "StandardsControlDescription": "This AWS control checks that there is
 at least one multi-region AWS CloudTrail trail includes read and write management
 events."
        },
        {
            "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-
benchmark/v/1.2.0",
            "SecurityControlId": "CloudTrail.1",
            "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/CloudTrail.1",
            "AssociationStatus": "ENABLED",
            "RelatedRequirements": [
                "CIS AWS Foundations 2.1"
            ],
            "UpdatedAt": "2020-02-10T21:22:53.998000+00:00",
            "StandardsControlTitle": "Ensure CloudTrail is enabled in all regions",
            "StandardsControlDescription": "AWS CloudTrail is a web service that
 records AWS API calls for your account and delivers log files to you. The recorded
 information includes the identity of the API caller, the time of the API call,
 the source IP address of the API caller, the request parameters, and the response
 elements returned by the AWS service."
        },
        {
```

```
            "StandardsArn": "arn:aws:securityhub:us-east-2::standards/aws-
foundational-security-best-practices/v/1.0.0",
            "SecurityControlId": "CloudTrail.1",
            "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/CloudTrail.1",
            "AssociationStatus": "DISABLED",
            "RelatedRequirements": [],
            "UpdatedAt": "2023-05-15T19:31:52.671000+00:00",
            "UpdatedReason": "Alternative compensating controls are in place",
            "StandardsControlTitle": "CloudTrail should be enabled and configured
 with at least one multi-Region trail that includes read and write management
 events",
            "StandardsControlDescription": "This AWS control checks that there is
 at least one multi-region AWS CloudTrail trail includes read and write management
 events."
        },
        {
            "StandardsArn": "arn:aws:securityhub:us-east-2::standards/cis-aws-
foundations-benchmark/v/1.4.0",
            "SecurityControlId": "CloudTrail.1",
            "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/CloudTrail.1",
            "AssociationStatus": "ENABLED",
            "RelatedRequirements": [
                "CIS AWS Foundations Benchmark v1.4.0/3.1"
            ],
            "UpdatedAt": "2022-11-10T15:40:36.021000+00:00",
            "StandardsControlTitle": "Ensure CloudTrail is enabled in all regions",
            "StandardsControlDescription": "AWS CloudTrail is a web service that
 records AWS API calls for your account and delivers log files to you. The recorded
 information includes the identity of the API caller, the time of the API call,
 the source IP address of the API caller, the request parameters, and the response
 elements returned by the AWS service. CloudTrail provides a history of AWS API
 calls for an account, including API calls made via the Management Console, SDKs,
 command line tools, and higher-level AWS services (such as CloudFormation)."
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的在特定標準中啟用和停用控制項。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListStandardsControlAssociations。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

擷取指派給資源的標籤

下列list-tags-for-resource範例會傳回指派給指定中樞資源的標籤。

```
aws securityhub list-tags-for-resource \
    --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default"
```

輸出：

```
{
    "Tags": {
        "Department" : "Operations",
        "Area" : "USMidwest"
    }
}
```

如需詳細資訊，請參閱 AWS CloudFormation 使用者指南中的 AWS：：SecurityHub：：Hub。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

start-configuration-policy-association

以下程式碼範例顯示如何使用 start-configuration-policy-association。

AWS CLI

範例 1：建立組態政策的關聯

下列start-configuration-policy-association範例會將指定的組態政策與指定的組織單位建立關聯。組態可能與目標帳戶、組織單位或根帳戶相關聯。

```
aws securityhub start-configuration-policy-association \
    --configuration-policy-identifier "arn:aws:securityhub:eu-
central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333" \
    --target '{"OrganizationalUnitId": "ou-6hi7-8j91kl2m"}'
```

輸出：

```
{
    "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "TargetId": "ou-6hi7-8j91kl2m",
    "TargetType": "ORGANIZATIONAL_UNIT",
    "AssociationType": "APPLIED",
    "UpdatedAt": "2023-11-29T17:40:52.468000+00:00",
    "AssociationStatus": "PENDING"
}
```

如需詳細資訊，請參閱 Security Hub 使用者指南中的建立和關聯 Security Hub 組態政策。 AWS

範例 2：建立自我管理組態的關聯

下列start-configuration-policy-association範例會將自我管理組態與指定的帳戶建立關聯。

```
aws securityhub start-configuration-policy-association \
    --configuration-policy-identifier "SELF_MANAGED_SECURITY_HUB" \
    --target '{"OrganizationalUnitId": "123456789012"}'
```

輸出：

```
{
    "ConfigurationPolicyId": "SELF_MANAGED_SECURITY_HUB",
    "TargetId": "123456789012",
    "TargetType": "ACCOUNT",
    "AssociationType": "APPLIED",
    "UpdatedAt": "2023-11-29T17:40:52.468000+00:00",
    "AssociationStatus": "PENDING"
}
```

如需詳細資訊，請參閱 Security Hub 使用者指南中的建立和關聯 Security Hub 組態政策。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartConfigurationPolicyAssociation。

start-configuration-policy-disassociation

以下程式碼範例顯示如何使用 start-configuration-policy-disassociation。

AWS CLI

範例 1：取消組態政策的關聯

下列start-configuration-policy-disassociation範例會取消組態政策與指定組織單位的關聯。組態可能會與目標帳戶、組織單位或根帳戶取消關聯。

```
aws securityhub start-configuration-policy-disassociation \
    --configuration-policy-identifier "arn:aws:securityhub:eu-
central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333" \
    --target '{"OrganizationalUnitId": "ou-6hi7-8j91kl2m"}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的取消組態與帳戶和 OUs的關聯。

範例 2：取消自我管理組態的關聯

下列start-configuration-policy-disassociation範例會取消自我管理組態與指定帳戶的關聯。

```
aws securityhub start-configuration-policy-disassociation \
    --configuration-policy-identifier "SELF_MANAGED_SECURITY_HUB" \
    --target '{"AccountId": "123456789012"}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的取消組態與帳戶和 OUs的關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartConfigurationPolicyDisassociation。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤指派給資源

下列tag-resource範例會將 Department and Area 標籤的值指派給指定的中樞資源。

```
aws securityhub tag-resource \
    --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default" \
    --tags '{"Department":"Operations", "Area":"USMidwest"}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CloudFormation 使用者指南中的 AWS：：SecurityHub：：Hub。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從資源移除標籤值

下列untag-resource範例會從指定的中樞資源移除 Department 標籤。

```
aws securityhub untag-resource \
    --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default" \
    --tag-keys "Department"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CloudFormation 使用者指南中的 AWS：：SecurityHub：：Hub。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-action-target

以下程式碼範例顯示如何使用 update-action-target。

AWS CLI

更新自訂動作

下列update-action-target範例會更新指定 ARN 所識別的自訂動作名稱。

```
aws securityhub update-action-target \
    --action-target-arn "arn:aws:securityhub:us-west-1:123456789012:action/custom/
Remediation" \
    --name "Send to remediation"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的建立自訂動作並將其與 CloudWatch Events 規則建立關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateActionTarget。

update-configuration-policy

以下程式碼範例顯示如何使用 update-configuration-policy。

AWS CLI

更新組態政策

下列update-configuration-policy範例會更新現有的組態政策，以使用指定的設定。

```
aws securityhub update-configuration-policy \
    --identifier "arn:aws:securityhub:eu-central-1:508236694226:configuration-policy/09f37766-57d8-4ede-9d33-5d8b0fecf70e" \
    --name "SampleConfigurationPolicyUpdated" \
    --description "SampleDescriptionUpdated" \
    --configuration-policy '{"SecurityHub": {"ServiceEnabled": true, "EnabledStandardIdentifiers": ["arn:aws:securityhub:eu-central-1::standards/aws-foundational-security-best-practices/v/1.0.0","arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0"],"SecurityControlsConfiguration":{"DisabledSecurityControlIdentifiers": ["CloudWatch.1"], "SecurityControlCustomParameters": [{"SecurityControlId": "ACM.1", "Parameters": {"daysToExpiration": {"ValueType": "CUSTOM", "Value": {"Integer": 21}}}}]}}}' \
    --updated-reason "Disabling CloudWatch.1 and changing parameter value"
```

輸出：

```
{
    "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Name": "SampleConfigurationPolicyUpdated",
    "Description": "SampleDescriptionUpdated",
    "UpdatedAt": "2023-11-28T20:28:04.494000+00:00",
    "CreatedAt": "2023-11-28T20:28:04.494000+00:00",
    "ConfigurationPolicy": {
        "SecurityHub": {
            "ServiceEnabled": true,
```

```
            "EnabledStandardIdentifiers": [
                "arn:aws:securityhub:eu-central-1::standards/aws-foundational-
security-best-practices/v/1.0.0",
                "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"
            ],
            "SecurityControlsConfiguration": {
                "DisabledSecurityControlIdentifiers": [
                    "CloudWatch.1"
                ],
                "SecurityControlCustomParameters": [
                    {
                        "SecurityControlId": "ACM.1",
                        "Parameters": {
                            "daysToExpiration": {
                                "ValueType": "CUSTOM",
                                "Value": {
                                    "Integer": 21
                                }
                            }
                        }
                    }
                ]
            }
        }
    }
}
```

如需詳細資訊，請參閱 Security Hub 使用者指南中的更新 Security Hub 組態政策。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateConfigurationPolicy。

update-finding-aggregator

以下程式碼範例顯示如何使用 update-finding-aggregator。

AWS CLI

更新目前的調查結果彙總組態

下列update-finding-aggregator範例會將調查結果彙總組態變更為來自所選區域的連結。它從美國東部（維吉尼亞）執行，這是彙總區域。它會選取美國西部（加利佛尼亞北部）和美國西部（奧勒岡）做為連結的區域。

```
aws securityhub update-finding-aggregator \
    --region us-east-1 \
    --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-
aggregator/123e4567-e89b-12d3-a456-426652340000 \
    --region-linking-mode SPECIFIED_REGIONS \
    --regions us-west-1,us-west-2
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的更新調查結果彙總組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateFindingAggregator。

update-insight

以下程式碼範例顯示如何使用 update-insight。

AWS CLI

範例 1：變更自訂洞見的篩選條件

下列update-insight範例會變更自訂洞見的篩選條件。更新的洞見會尋找與 AWS 角色相關的嚴重性較高的問題清單。

```
aws securityhub update-insight \
    --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
    --filters '{"ResourceType": [{ "Comparison": "EQUALS", "Value": "AwsIamRole"}],
 "SeverityLabel": [{"Comparison": "EQUALS", "Value": "HIGH"}]}' \
    --name "High severity role findings"
```

範例 2：變更自訂洞見的分組屬性

下列update-insight範例會使用指定的 ARN 變更自訂洞見的分組屬性。新的分組屬性是資源 ID。

```
aws securityhub update-insight \
    --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
    --group-by-attribute "ResourceId" \
    --name "Critical role findings"
```

輸出：

```
{
    "Insights": [
        {
            "InsightArn": "arn:aws:securityhub:us-
west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",
            "Name": "Critical role findings",
            "Filters": {
                "SeverityLabel": [
                    {
                        "Value": "CRITICAL",
                        "Comparison": "EQUALS"
                    }
                ],
                "ResourceType": [
                    {
                        "Value": "AwsIamRole",
                        "Comparison": "EQUALS"
                    }
                ]
            },
            "GroupByAttribute": "ResourceId"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的管理自訂洞見。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateInsight。

update-organization-configuration

以下程式碼範例顯示如何使用 update-organization-configuration。

AWS CLI

更新為組織設定 Security Hub 的方式

下列update-organization-configuration範例指定 Security Hub 應使用中央組態來設定組織。執行此命令後，委派的 Security Hub 管理員可以建立和管理組態政策來設定組織。委派管理員

也可以使用此命令，從中央切換到本機組態。如果本機組態是組態類型，委派管理員可以選擇是否要在新組織帳戶中自動啟用 Security Hub 和預設安全標準。

```
aws securityhub update-organization-configuration \
    --no-auto-enable \
    --organization-configuration '{"ConfigurationType": "CENTRAL"}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的使用 AWS Organizations 管理帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateOrganizationConfiguration。

update-security-control

以下程式碼範例顯示如何使用 update-security-control。

AWS CLI

更新安全控制屬性

下列update-security-control範例指定 Security Hub 安全控制參數的自訂值。

```
aws securityhub update-security-control \
    --security-control-id ACM.1 \
    --parameters '{"daysToExpiration": {"ValueType": "CUSTOM", "Value": {"Integer":
 15}}}' \
    --last-update-reason "Internal compliance requirement"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的自訂控制參數。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSecurityControl。

update-security-hub-configuration

以下程式碼範例顯示如何使用 update-security-hub-configuration。

AWS CLI

更新 Security Hub 組態

下列update-security-hub-configuration範例會將 Security Hub 設定為自動啟用啟用標準的新控制項。

```
aws securityhub update-security-hub-configuration \
    --auto-enable-controls
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的自動啟用新控制項。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSecurityHubConfiguration。

update-standards-control

以下程式碼範例顯示如何使用 update-standards-control。

AWS CLI

範例 1：停用控制項

下列update-standards-control範例會停用 PCI.AutoScaling.1 控制項。

```
aws securityhub update-standards-control \
    --standards-control-arn "arn:aws:securityhub:us-west-1:123456789012:control/pci-dss/v/3.2.1/PCI.AutoScaling.1" \
    --control-status "DISABLED" \
    --disabled-reason "Not applicable for my service"
```

此命令不會產生輸出。

範例 2：啟用控制項

下列update-standards-control範例會啟用 PCI.AutoScaling.1 控制項。

```
aws securityhub update-standards-control \
    --standards-control-arn "arn:aws:securityhub:us-west-1:123456789012:control/pci-dss/v/3.2.1/PCI.AutoScaling.1" \
    --control-status "ENABLED"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Security Hub 使用者指南中的停用和啟用個別控制項。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateStandardsControl。

使用 的 Security Lake 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Security Lake 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-aws-log-source`

以下程式碼範例顯示如何使用 `create-aws-log-source`。

AWS CLI

新增原生支援的 Amazon Web Service 做為 Amazon Security Lake 來源

下列`create-aws-logsource`範例會將 VPC 流程日誌新增為指定帳戶和區域中的安全湖來源。

```
aws securitylake create-aws-log-source \
    --sources '[{"regions": ["us-east-1"], "accounts": ["123456789012"],
 "sourceName": "SH_FINDINGS", "sourceVersion": "2.0"}]'
```

輸出：

```
{
    "failed": [
        "123456789012"
    ]
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的新增 AWS 服務做為來源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateAwsLogSource。

create-custom-log-source

以下程式碼範例顯示如何使用 create-custom-log-source。

AWS CLI

新增自訂來源做為 Amazon Security Lake 來源

下列create-custom-logsource範例會將自訂來源新增為指定日誌提供者帳戶和指定區域中的 Security Lake 來源。

```
aws securitylake create-custom-log-source \
    --source-name "VPC_FLOW" \
    --event-classes '["DNS_ACTIVITY", "NETWORK_ACTIVITY"]' \
    --configuration '{"crawlerConfiguration": {"roleArn": "arn:aws:glue:eu-
west-2:123456789012:crawler/E1WG1ZNPRXT0D4"},"providerIdentity": {"principal":
 "029189416600","externalId": "123456789012"}}' --region "us-east-1"
```

輸出：

```
{
    "customLogSource": {
        "attributes": {
            "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
E1WG1ZNPRXT0D4",
            "databaseArn": "arn:aws:glue:eu-west-2:123456789012:database/
E1WG1ZNPRXT0D4",
            "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/E1WG1ZNPRXT0D4"
        },
        "provider": {
            "location": "amzn-s3-demo-bucket--usw2-az1--x-s3",
            "roleArn": "arn:aws:iam::123456789012:role/AmazonSecurityLake-Provider-
testCustom2-eu-west-2"
        },
        "sourceName": "testCustom2"
        "sourceVersion": "2.0"
    }
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的新增自訂來源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCustomLogSource。

create-data-lake-exception-subscription

以下程式碼範例顯示如何使用 create-data-lake-exception-subscription。

AWS CLI

傳送 Security Lake 例外狀況的通知

下列create-data-lake-exception-subscription範例會透過簡訊傳遞，將 Security Lake 例外狀況的通知傳送至指定的帳戶。例外狀況訊息會保留在指定的期間內。

```
aws securitylake create-data-lake-exception-subscription \
    --notification-endpoint "123456789012" \
    --exception-time-to-live 30 \
    --subscription-protocol "sms"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的對 Amazon Security Lake 進行故障診斷。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDataLakeExceptionSubscription。

create-data-lake-organization-configuration

以下程式碼範例顯示如何使用 create-data-lake-organization-configuration。

AWS CLI

在新的組織帳戶中設定 Security Lake

下列create-data-lake-organization-configuration範例會啟用 Security Lake 和新組織帳戶中指定來源事件和日誌的集合。

```
aws securitylake create-data-lake-organization-configuration \
    --auto-enable-new-account '[{"region":"us-east-1","sources":
[{"sourceName":"SH_FINDINGS","sourceVersion": "1.0"}]}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的使用 AWS Organizations 管理多個帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDataLakeOrganizationConfiguration。

create-data-lake

以下程式碼範例顯示如何使用 create-data-lake。

AWS CLI

範例 1：在多個區域中設定您的資料湖

下列create-data-lake範例會在多個 AWS 區域中啟用 Amazon Security Lake，並設定您的資料湖。

```
aws securitylake create-data-lake \
    --configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-1","lifecycleConfiguration":
 {"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}}, {"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-2","lifecycleConfiguration":
 {"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}}]' \
    --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"
```

輸出：

```
{
    "dataLakes": [
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-1:522481757177:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "S3_MANAGED_KEY"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 365
                },
```

```
                "transitions": [
                    {
                        "days": 60,
                        "storageClass": "ONEZONE_IA"
                    }
                ]
            },
            "region": "us-east-1",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-1-
gnevt6s8z7bzby8oi3uiaysbr8v2ml",
            "updateStatus": {
                "exception": {},
                "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
                "status": "INITIALIZED"
            }
        },
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "S3_MANAGED_KEY"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 365
                },
                "transitions": [
                    {
                        "days": 60,
                        "storageClass": "ONEZONE_IA"
                    }
                ]
            },
            "region": "us-east-2",
            "replicationConfiguration": {
                "regions": [
```

```
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-2-
cehuifzl5rwmhm6m62h7zhvtseogr9",
            "updateStatus": {
                "exception": {},
                "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
                "status": "INITIALIZED"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的 Amazon Security Lake 入門。

範例 2：在單一區域中設定您的資料湖

下列create-data-lake範例會在單一區域中啟用 Amazon Security Lake， AWS 並設定您的資料湖。

```
aws securitylake create-data-lake \
    --configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"1234abcd-12ab-34cd-56ef-1234567890ab"},"region":"us-
east-2","lifecycleConfiguration": {"expiration":{"days":500},"transitions":
[{"days":30,"storageClass":"GLACIER"}]}}]' \
    --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"
```

輸出：

```
{
    "dataLakes": [
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
            },
            "lifecycleConfiguration": {
```

```
                "expiration": {
                    "days": 500
                },
                "transitions": [
                    {
                        "days": 30,
                        "storageClass": "GLACIER"
                    }
                ]
            },
            "region": "us-east-2",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-2-
cehuifzl5rwmhm6m62h7zhvtseogr9",
            "updateStatus": {
                "exception": {},
                "requestId": "77702a53-dcbf-493e-b8ef-518e362f3003",
                "status": "INITIALIZED"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的 Amazon Security Lake 入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateDataLake。

create-subscriber-notification

以下程式碼範例顯示如何使用 create-subscriber-notification。

AWS CLI

建立訂閱者通知

下列create-subscriber-notification範例顯示如何指定訂閱者通知，以在將新資料寫入資料湖時建立通知。

```
aws securitylake create-subscriber-notification \
    --subscriber-id "12345ab8-1a34-1c34-1bd4-12345ab9012" \
    --configuration '{"httpsNotificationConfiguration":
 {"targetRoleArn":"arn:aws:iam::XXX:role/service-role/RoleName",
 "endpoint":"https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"}}'
```

輸出：

```
{
    "subscriberEndpoint": [
        "https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"
    ]
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的訂閱者管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSubscriberNotification。

create-subscriber

以下程式碼範例顯示如何使用 create-subscriber。

AWS CLI

範例 1：建立具有資料存取權的訂閱者

下列create-subscriber範例會在 Security Lake 中建立訂閱者，以存取目前 AWS 區域中的 AWS 來源指定訂閱者身分的資料。

```
aws securitylake create-subscriber \
    --access-types "S3" \
    --sources '[{"awsLogSource": {"sourceName": "VPC_FLOW","sourceVersion":
 "2.0"}}]' \
    --subscriber-name 'opensearch-s3' \
    --subscriber-identity '{"principal": "029189416600","externalId":
 "123456789012"}'
```

輸出：

```
{
    "subscriber": {
```

```
        "accessTypes": [
            "S3"
        ],
        "createdAt": "2024-07-17T19:08:26.787000+00:00",
        "roleArn": "arn:aws:iam::773172568199:role/AmazonSecurityLake-896f218b-
cfba-40be-a255-8b49a65d0407",
        "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-1-
um632ufwpvxkyz0bc5hkb64atycnf3",
        "sources": [
            {
                "awsLogSource": {
                    "sourceName": "VPC_FLOW",
                    "sourceVersion": "2.0"
                }
            }
        ],
        "subscriberArn": "arn:aws:securitylake:us-
east-1:773172568199:subscriber/896f218b-cfba-40be-a255-8b49a65d0407",
        "subscriberId": "896f218b-cfba-40be-a255-8b49a65d0407",
        "subscriberIdentity": {
            "externalId": "123456789012",
            "principal": "029189416600"
        },
        "subscriberName": "opensearch-s3",
        "subscriberStatus": "ACTIVE",
        "updatedAt": "2024-07-17T19:08:27.133000+00:00"
    }
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的建立具有資料存取權的訂閱者。

範例 2：建立具有查詢存取權的訂閱者

下列create-subscriber範例會在 Security Lake 中建立訂閱者，並在目前 AWS 區域中為指定的訂閱者身分提供查詢存取權。

```
aws securitylake create-subscriber \
    --access-types "LAKEFORMATION" \
    --sources '[{"awsLogSource": {"sourceName": "VPC_FLOW","sourceVersion":
 "2.0"}}]' \
    --subscriber-name 'opensearch-s3' \
    --subscriber-identity '{"principal": "029189416600","externalId":
 "123456789012"}'
```

輸出：

```
{
    "subscriber": {
        "accessTypes": [
            "LAKEFORMATION"
        ],
        "createdAt": "2024-07-18T01:05:55.853000+00:00",
        "resourceShareArn": "arn:aws:ram:us-east-1:123456789012:resource-
share/8c31da49-c224-4f1e-bb12-37ab756d6d8a",
        "resourceShareName": "LakeFormation-V2-NAMENAMENA-123456789012",
        "sources": [
            {
                "awsLogSource": {
                    "sourceName": "VPC_FLOW",
                    "sourceVersion": "2.0"
                }
            }
        ],
        "subscriberArn": "arn:aws:securitylake:us-east-1:123456789012:subscriber/
e762aabb-ce3d-4585-beab-63474597845d",
        "subscriberId": "e762aabb-ce3d-4585-beab-63474597845d",
        "subscriberIdentity": {
            "externalId": "123456789012",
            "principal": "029189416600"
        },
        "subscriberName": "opensearch-s3",
        "subscriberStatus": "ACTIVE",
        "updatedAt": "2024-07-18T01:05:58.393000+00:00"
    }
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的建立具有查詢存取權的訂閱者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSubscriber。

delete-aws-log-source

以下程式碼範例顯示如何使用 delete-aws-log-source。

AWS CLI

移除原生支援的 AWS 服務。

下列delete-aws-logsource範例會將 VPC 流程日誌刪除為指定帳戶和區域中的安全湖來源。

```
aws securitylake delete-aws-log-source \
    --sources '[{"regions": ["us-east-1"], "accounts": ["123456789012"],
 "sourceName": "SH_FINDINGS", "sourceVersion": "2.0"}]'
```

輸出：

```
{
    "failed": [
        "123456789012"
    ]
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的移除 AWS 服務做為來源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAwsLogSource。

delete-custom-log-source

以下程式碼範例顯示如何使用 delete-custom-log-source。

AWS CLI

移除自訂來源。

下列delete-custom-logsource範例會刪除指定區域中指定日誌提供者帳戶中的自訂來源。

```
aws securitylake delete-custom-log-source \
    --source-name "CustomSourceName"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的刪除自訂來源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCustomLogSource。

delete-data-lake-organization-configuration

以下程式碼範例顯示如何使用 delete-data-lake-organization-configuration。

AWS CLI

停止成員帳戶中的自動來源集合

下列delete-data-lake-organization-configuration範例會停止從加入組織的新成員帳戶自動收集 AWS Security Hub 調查結果。只有委派的 Security Lake 管理員才能執行此命令。它可防止新成員帳戶自動將資料提供給資料湖。

```
aws securitylake delete-data-lake-organization-configuration \
    --auto-enable-new-account '[{"region":"us-east-1","sources":
[{"sourceName":"SH_FINDINGS"}]}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的使用 AWS Organizations 管理多個帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDataLakeOrganizationConfiguration。

delete-data-lake

以下程式碼範例顯示如何使用 delete-data-lake。

AWS CLI

停用您的資料湖

下列delete-data-lake範例會停用指定區域中的資料湖 AWS 。在指定的區域中，來源不會再將資料貢獻至資料湖。對於使用 AWS Organizations 的 Security Lake 部署，只有組織的委派 Security Lake 管理員可以停用組織中帳戶的 Security Lake。

```
aws securitylake delete-data-lake \
    --regions "ap-northeast-1" "eu-central-1"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的停用 Amazon Security Lake。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteDataLake。

delete-subscriber-notification

以下程式碼範例顯示如何使用 delete-subscriber-notification。

AWS CLI

刪除訂閱者通知

下列delete-subscriber-notification範例顯示如何刪除特定 Security Lake 訂閱者的訂閱者通知。

```
aws securitylake delete-subscriber-notification \
    --subscriber-id "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的訂閱者管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSubscriberNotification。

delete-subscriber

以下程式碼範例顯示如何使用 delete-subscriber。

AWS CLI

刪除訂閱者

下列delete-subscriber範例顯示，如果您不再希望訂閱者從 Security Lake 取用資料，如何移除訂閱者。

```
aws securitylake delete-subscriber \
    --subscriber-id "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的訂閱者管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSubscriber。

get-data-lake-exception-subscription

以下程式碼範例顯示如何使用 get-data-lake-exception-subscription。

AWS CLI

取得例外狀況訂閱的詳細資訊

下列get-data-lake-exception-subscription範例提供 Security Lake 例外狀況訂閱的詳細資訊。在此範例中，指定 AWS 帳戶的使用者會透過簡訊交付收到錯誤通知。例外狀況訊息會在指定的期間內保留在帳戶中。例外狀況訂閱會透過申請者的偏好通訊協定，將錯誤通知 Security Lake 使用者。

```
aws securitylake get-data-lake-exception-subscription
```

輸出：

```
{
    "exceptionTimeToLive": 30,
    "notificationEndpoint": "123456789012",
    "subscriptionProtocol": "sms"
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的對資料湖狀態進行故障診斷。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDataLakeExceptionSubscription。

get-data-lake-organization-configuration

以下程式碼範例顯示如何使用 get-data-lake-organization-configuration。

AWS CLI

取得新組織帳戶組態的詳細資訊

下列get-data-lake-organization-configuration範例會擷取新組織帳戶在加入 Amazon Security Lake 後將傳送之來源日誌的詳細資訊。

```
aws securitylake get-data-lake-organization-configuration
```

輸出：

```
{
    "autoEnableNewAccount": [
        {
            "region": "us-east-1",
```

```
            "sources": [
                {
                    "sourceName": "VPC_FLOW",
                    "sourceVersion": "1.0"
                },
                {
                    "sourceName": "ROUTE53",
                    "sourceVersion": "1.0"
                },
                {
                    "sourceName": "SH_FINDINGS",
                    "sourceVersion": "1.0"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的使用 AWS Organizations 管理多個帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDataLakeOrganizationConfiguration。

get-data-lake-sources

以下程式碼範例顯示如何使用 get-data-lake-sources。

AWS CLI

取得日誌收集的狀態

下列get-data-lake-sources範例會取得目前 AWS 區域中指定帳戶的日誌集合快照。帳戶已啟用 Amazon Security Lake。

```
aws securitylake get-data-lake-sources \
    --accounts "123456789012"
```

輸出：

```
{
    "dataLakeSources": [
        {
```

```
            "account": "123456789012",
            "sourceName": "SH_FINDINGS",
            "sourceStatuses": [
                {
                    "resource": "vpc-1234567890abcdef0",
                    "status": "COLLECTING"
                }
            ]
        },
        {
            "account": "123456789012",
            "sourceName": "VPC_FLOW",
            "sourceStatuses": [
                {
                    "resource": "vpc-1234567890abcdef0",
                    "status": "NOT_COLLECTING"
                }
            ]
        },
        {
            "account": "123456789012",
            "sourceName": "LAMBDA_EXECUTION",
            "sourceStatuses": [
                {
                    "resource": "vpc-1234567890abcdef0",
                    "status": "COLLECTING"
                }
            ]
        },
        {
            "account": "123456789012",
            "sourceName": "ROUTE53",
            "sourceStatuses": [
                {
                    "resource": "vpc-1234567890abcdef0",
                    "status": "COLLECTING"
                }
            ]
        },
        {
            "account": "123456789012",
            "sourceName": "CLOUD_TRAIL_MGMT",
            "sourceStatuses": [
                {
```

```
                    "resource": "vpc-1234567890abcdef0",
                    "status": "COLLECTING"
                }
            ]
        }
    ],
    "dataLakeArn": null
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的從 AWS 服務收集資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDataLakeSources。

get-subscriber

以下程式碼範例顯示如何使用 get-subscriber。

AWS CLI

擷取訂閱資訊

下列get-subscriber範例會擷取指定 Securiy Lake 訂閱者的訂閱資訊。

```
aws securitylake get-subscriber \
    --subscriber-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "subscriber": {
        "accessTypes": [
            "LAKEFORMATION"
        ],
        "createdAt": "2024-04-19T15:19:44.421803+00:00",
        "resourceShareArn": "arn:aws:ram:eu-west-2:123456789012:resource-share/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "resourceShareName": "LakeFormation-V3-TKJGBHCKTZ-123456789012",
        "sources": [
            {
                "awsLogSource": {
                    "sourceName": "LAMBDA_EXECUTION",
                    "sourceVersion": "1.0"
                }
```

```
            },
            {
                "awsLogSource": {
                    "sourceName": "EKS_AUDIT",
                    "sourceVersion": "2.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "ROUTE53",
                    "sourceVersion": "1.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "SH_FINDINGS",
                    "sourceVersion": "1.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "VPC_FLOW",
                    "sourceVersion": "1.0"
                }
            },
            {
                "customLogSource": {
                    "attributes": {
                        "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
testCustom2",
                        "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/amazon_security_lake_glue_db_eu_west_2",
                        "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
amazon_security_lake_table_eu_west_2_ext_testcustom2"
                    },
                    "provider": {
                        "location": "s3://aws-security-data-lake-eu-
west-2-8ugsus4ztnsfpjbldwbgf4vge98av9/ext/testCustom2/",
                        "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-Provider-testCustom2-eu-west-2"
                    },
                    "sourceName": "testCustom2"
                }
            },
```

```
            {
                "customLogSource": {
                    "attributes": {
                        "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
TestCustom",
                        "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/amazon_security_lake_glue_db_eu_west_2",
                        "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
amazon_security_lake_table_eu_west_2_ext_testcustom"
                    },
                    "provider": {
                        "location": "s3://aws-security-data-lake-eu-
west-2-8ugsus4ztnsfpjbldwbgf4vge98av9/ext/TestCustom/",
                        "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-Provider-TestCustom-eu-west-2"
                    },
                    "sourceName": "TestCustom"
                }
            }
        ],
        "subscriberArn": "arn:aws:securitylake:eu-west-2:123456789012:subscriber/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "subscriberId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "subscriberIdentity": {
            "externalId": "123456789012",
            "principal": "123456789012"
        },
        "subscriberName": "test",
        "subscriberStatus": "ACTIVE",
        "updatedAt": "2024-04-19T15:19:55.230588+00:00"
    }
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的訂閱者管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSubscriber。

list-data-lake-exceptions

以下程式碼範例顯示如何使用 `list-data-lake-exceptions`。

AWS CLI

列出影響資料湖的問題

下列list-data-lake-exceptions範例列出過去 14 天內在指定 AWS 區域中影響資料湖的問題。

```
aws securitylake list-data-lake-exceptions \
    --regions "us-east-1" "eu-west-3"
```

輸出：

```
{
    "exceptions": [
        {
            "exception": "The account does not have the required role permissions.
 Update your role permissions to use the new data source version.",
            "region": "us-east-1",
            "timestamp": "2024-02-29T12:24:15.641725+00:00"
        },
        {
            "exception": "The account does not have the required role permissions.
 Update your role permissions to use the new data source version.",
            "region": "eu-west-3",
            "timestamp": "2024-02-29T12:24:15.641725+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的對 Amazon Security Lake 進行故障診斷。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDataLakeExceptions。

list-data-lakes

以下程式碼範例顯示如何使用 list-data-lakes。

AWS CLI

列出 Security Lake 組態物件

下列list-data-lakes範例列出指定 AWS 區域的 Amazon Security Lake 組態物件。您可以使用此命令來判斷指定區域或區域中是否已啟用 Security Lake。

```
aws securitylake list-data-lakes \
```

```
    --regions "us-east-1"
```

輸出：

```
{
    "dataLakes": [
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-1:123456789012:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "S3_MANAGED_KEY"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 365
                },
                "transitions": [
                    {
                        "days": 60,
                        "storageClass": "ONEZONE_IA"
                    }
                ]
            },
            "region": "us-east-1",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:123456789012:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-
east-1-1234567890abcdef0",
            "updateStatus": {
                "exception": {
                    "code": "software.amazon.awssdk.services.s3.model.S3Exception",
                    "reason": ""
                },
                "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
                "status": "FAILED"
            }
        }
```

```
    ]
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的檢查區域狀態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDataLakes。

`list-log-sources`

以下程式碼範例顯示如何使用 `list-log-sources`。

AWS CLI

擷取 Amazon Security Lake 日誌來源

下列`list-log-sources`範例列出指定帳戶中的 Amazon Security Lake 日誌來源。

```
aws securitylake list-log-sources \
    --accounts "123456789012"
```

輸出：

```
{
    "account": "123456789012",
    "region": "xy-region-1",
    "sources": [
        {
               "awsLogSource": {
                "sourceName": "VPC_FLOW",
                "sourceVersion": "2.0"
            }
        },
        {
            "awsLogSource": {
                "sourceName": "SH_FINDINGS",
                "sourceVersion": "2.0"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的來源管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListLogSources。

`list-subscribers`

以下程式碼範例顯示如何使用 `list-subscribers`。

AWS CLI

擷取 Amazon Security Lake 訂閱者

下列`list-subscribers`範例列出特定帳戶中的所有 Amazon Security Lake 訂閱者。

```
aws securitylake list-subscribers
```

輸出：

```
{
    "subscribers": [
        {
            "accessTypes": [
                "S3"
            ],
            "createdAt": "2024-06-04T15:02:28.921000+00:00",
            "roleArn": "arn:aws:iam::123456789012:role/AmazonSecurityLake-
E1WG1ZNPRXT0D4",
            "s3BucketArn": "amzn-s3-demo-bucket--usw2-az1--x-s3",
            "sources": [
                {
                    "awsLogSource": {
                        "sourceName": "CLOUD_TRAIL_MGMT",
                        "sourceVersion": "2.0"
                    }
                },
                {
                    "awsLogSource": {
                        "sourceName": "LAMBDA_EXECUTION",
                        "sourceVersion": "1.0"
                    }
                },
                {
                    "customLogSource": {
                        "attributes": {
```

```
                            "crawlerArn": "arn:aws:glue:eu-
west-2:123456789012:crawler/E1WG1ZNPRXT0D4",
                            "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/E1WG1ZNPRXT0D4",
                            "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
E1WG1ZNPRXT0D4"
                        },
                        "provider": {
                            "location": "amzn-s3-demo-bucket--usw2-az1--x-s3",
                            "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-E1WG1ZNPRXT0D4"
                        },
                        "sourceName": "testCustom2"
                    }
                }
            ],
            "subscriberArn": "arn:aws:securitylake:eu-
west-2:123456789012:subscriber/E1WG1ZNPRXT0D4",
            "subscriberEndpoint": "arn:aws:sqs:eu-
west-2:123456789012:AmazonSecurityLake-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111-Main-
Queue",
            "subscriberId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "subscriberIdentity": {
                "externalId": "ext123456789012",
                "principal": "123456789012"
            },
            "subscriberName": "Test",
            "subscriberStatus": "ACTIVE",
            "updatedAt": "2024-06-04T15:02:35.617000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的訂閱者管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSubscribers。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

列出現有資源的標籤

下列list-tags-for-resource範例列出指定 Amazon Security Lake 訂閱者的標籤。在此範例中，擁有者標籤金鑰沒有相關聯的標籤值。您也可以使用此操作來列出其他現有 Security Lake 資源的標籤。

```
aws securitylake list-tags-for-resource \
    --resource-arn "arn:aws:securitylake:us-
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab"
```

輸出：

```
{
    "tags": [
        {
            "key": "Environment",
            "value": "Cloud"
        },
        {
            "key": "CostCenter",
            "value": "12345"
        },
        {
            "key": "Owner",
            "value": ""
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的標記 Amazon Security Lake 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

register-data-lake-delegated-administrator

以下程式碼範例顯示如何使用 register-data-lake-delegated-administrator。

AWS CLI

指定委派管理員

下列register-data-lake-delegated-administrator範例會將指定的 AWS 帳戶指定為委派的 Amazon Security Lake 管理員。

```
aws securitylake register-data-lake-delegated-administrator \
    --account-id 123456789012
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的使用 AWS Organizations 管理多個帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterDataLakeDelegatedAdministrator。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤新增至現有資源

下列tag-resource範例會將標籤新增至現有的訂閱者資源。若要建立新的資源並新增一或多個標籤，請勿使用此操作。請改為針對您要建立的資源類型使用適當的建立操作。

```
aws securitylake tag-resource \
    --resource-arn "arn:aws:securitylake:us-
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab" \
    --tags key=Environment,value=Cloud
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的標記 Amazon Security Lake 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從現有資源移除標籤

下列untag-resource範例會從現有的訂閱者資源移除指定的標籤。

```
aws securitylake untag-resource \
    --resource-arn "arn:aws:securitylake:us-
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab" \
    --tags Environment Owner
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的標記 Amazon Security Lake 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-data-lake-exception-subscription

以下程式碼範例顯示如何使用 update-data-lake-exception-subscription。

AWS CLI

更新 Security Lake 例外狀況的通知訂閱

下列update-data-lake-exception-subscription範例會更新通知訂閱，以通知使用者 Security Lake 例外狀況。

```
aws securitylake update-data-lake-exception-subscription \
    --notification-endpoint "123456789012" \
    --exception-time-to-live 30 \
    --subscription-protocol "email"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的對 Amazon Security Lake 進行故障診斷。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDataLakeExceptionSubscription。

update-data-lake

以下程式碼範例顯示如何使用 update-data-lake。

AWS CLI

範例 1：更新資料湖設定

下列update-data-lake範例會更新 Amazon Security Lake 資料湖的設定。您可以使用此操作來指定資料加密、儲存和彙總區域設定。

```
aws securitylake update-data-lake \
    --configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-1","lifecycleConfiguration":
 {"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}}, {"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-2","lifecycleConfiguration":
 {"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}}]' \
    --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"
```

輸出：

```
{
    "dataLakes": [
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-1:522481757177:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "S3_MANAGED_KEY"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 365
                },
                "transitions": [
                    {
                        "days": 60,
                        "storageClass": "ONEZONE_IA"
                    }
                ]
            },
            "region": "us-east-1",
            "replicationConfiguration": {
                "regions": [
```

```
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-1-
gnevt6s8z7bzby8oi3uiaysbr8v2ml",
            "updateStatus": {
                "exception": {},
                "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
                "status": "INITIALIZED"
            }
        },
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "S3_MANAGED_KEY"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 365
                },
                "transitions": [
                    {
                        "days": 60,
                        "storageClass": "ONEZONE_IA"
                    }
                ]
            },
            "region": "us-east-2",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-2-
cehuifzl5rwmhm6m62h7zhvtseogr9",
            "updateStatus": {
                "exception": {},
                "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
```

```
                "status": "INITIALIZED"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的 Amazon Security Lake 入門。

範例 2：在單一區域中設定您的資料湖

下列create-data-lake範例會在單一區域中啟用 Amazon Security Lake， AWS 並設定您的資料湖。

```
aws securitylake create-data-lake \
    --configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"1234abcd-12ab-34cd-56ef-1234567890ab"},"region":"us-
east-2","lifecycleConfiguration": {"expiration":{"days":500},"transitions":
[{"days":30,"storageClass":"GLACIER"}]}}]' \
    --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"
```

輸出：

```
{
    "dataLakes": [
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 500
                },
                "transitions": [
                    {
                        "days": 30,
                        "storageClass": "GLACIER"
                    }
                ]
            },
```

```
            "region": "us-east-2",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-2-
cehuifzl5rwmhm6m62h7zhvtseogr9",
            "updateStatus": {
                "exception": {},
                "requestId": "77702a53-dcbf-493e-b8ef-518e362f3003",
                "status": "INITIALIZED"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的 Amazon Security Lake 入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDataLake。

update-subscriber-notification

以下程式碼範例顯示如何使用 update-subscriber-notification。

AWS CLI

更新訂閱者通知

下列update-subscriber-notification範例顯示如何更新訂閱者的通知方法。

```
aws securitylake update-subscriber-notification \
    --subscriber-id "12345ab8-1a34-1c34-1bd4-12345ab9012" \
    --configuration '{"httpsNotificationConfiguration":
 {"targetRoleArn":"arn:aws:iam::XXX:role/service-role/RoleName",
 "endpoint":"https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"}}'
```

輸出：

```
{
    "subscriberEndpoint": [
```

```
        "https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"
    ]
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的訂閱者管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSubscriberNotification。

update-subscriber

以下程式碼範例顯示如何使用 update-subscriber。

AWS CLI

更新 Amazon Security Lake 訂閱者。

下列update-subscriber範例會更新特定 Security Lake 訂閱者的安全湖資料存取來源。

```
aws securitylake update-subscriber \
    --subscriber-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "subscriber": {
        "accessTypes": [
            "LAKEFORMATION"
        ],
        "createdAt": "2024-04-19T15:19:44.421803+00:00",
        "resourceShareArn": "arn:aws:ram:eu-west-2:123456789012:resource-share/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "resourceShareName": "LakeFormation-V3-TKJGBHCKTZ-123456789012",
        "sources": [
            {
                "awsLogSource": {
                    "sourceName": "LAMBDA_EXECUTION",
                    "sourceVersion": "1.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "EKS_AUDIT",
                    "sourceVersion": "2.0"
```

```
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "ROUTE53",
                    "sourceVersion": "1.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "SH_FINDINGS",
                    "sourceVersion": "1.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "VPC_FLOW",
                    "sourceVersion": "1.0"
                }
            },
            {
                "customLogSource": {
                    "attributes": {
                        "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
E1WG1ZNPRXT0D4",
                        "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/E1WG1ZNPRXT0D4",
                        "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
E1WG1ZNPRXT0D4"
                    },
                    "provider": {
                        "location": "amzn-s3-demo-bucket--usw2-az1--x-s3",
                        "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-E1WG1ZNPRXT0D4"
                    },
                    "sourceName": "testCustom2"
                }
            }
        ],
        "subscriberArn": "arn:aws:securitylake:eu-west-2:123456789012:subscriber/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "subscriberId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "subscriberIdentity": {
            "externalId": "123456789012",
```

```
            "principal": "123456789012"
        },
        "subscriberName": "test",
        "subscriberStatus": "ACTIVE",
        "updatedAt": "2024-07-18T20:47:37.098000+00:00"
    }
}
```

如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的訂閱者管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSubscriber。

AWS Serverless Application Repository 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS Serverless Application Repository。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`put-application-policy`

以下程式碼範例顯示如何使用 `put-application-policy`。

AWS CLI

範例 1：公開共用應用程式

以下內容公開`put-application-policy`共用應用程式，因此任何人都可以在無 AWS 伺服器應用程式儲存庫中尋找和部署您的應用程式。

```
aws serverlessrepo put-application-policy \
    --application-id arn:aws:serverlessrepo:us-east-1:123456789012:applications/my-test-application \
```

```
    --statements Principals='*',Actions=Deploy
```

輸出：

```
{
    "Statements": [
        {
            "Actions": [
                "Deploy"
            ],
            "Principals": [
                ""
            ],
            "StatementId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
        }
    ]
}
```

範例 2：私下共用應用程式

以下私下put-application-policy共用應用程式，因此只有特定 AWS 帳戶可以在無 AWS 伺服器應用程式儲存庫中尋找和部署您的應用程式。

```
aws serverlessrepo put-application-policy \
    --application-id arn:aws:serverlessrepo:us-east-1:123456789012:applications/my-test-application \
    --statements Principals=111111111111,222222222222,Actions=Deploy
```

輸出：

```
{
    "Statements": [
        {
            "Actions": [
                "Deploy"
            ],
            "Principals": [
                "111111111111",
                "222222222222"
            ],
            "StatementId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

```
        }
    ]
}
```

如需詳細資訊，請參閱《 Serverless Application Repository 開發人員指南》中的透過主控台共用應用程式 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutApplicationPolicy。

使用 的 Service Catalog 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Service Catalog 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`accept-portfolio-share`

以下程式碼範例顯示如何使用 `accept-portfolio-share`。

AWS CLI

接受產品組合共享

下列`accept-portfolio-share`範例接受另一個使用者提出的提議，以共用指定的產品組合。

```
aws servicecatalog accept-portfolio-share \
    --portfolio-id port-2s6wuabcdefghijk
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AcceptPortfolioShare。

associate-principal-with-portfolio

以下程式碼範例顯示如何使用 associate-principal-with-portfolio。

AWS CLI

將委託人與產品組合建立關聯

下列associate-principal-with-portfolio範例會將使用者與指定的產品組合建立關聯。

```
aws servicecatalog associate-principal-with-portfolio \
    --portfolio-id port-2s6abcdefwdh4 \
    --principal-arn arn:aws:iam::123456789012:user/usertest \
    --principal-type IAM
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociatePrincipalWithPortfolio。

associate-product-with-portfolio

以下程式碼範例顯示如何使用 associate-product-with-portfolio。

AWS CLI

將產品與產品組合建立關聯

下列associate-product-with-portfolio範例會將指定的產品與指定的產品組合建立關聯。

```
aws servicecatalog associate-product-with-portfolio
     --product-id prod-3p5abcdef3oyk
     --portfolio-id port-2s6abcdef5wdh4
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateProductWithPortfolio。

associate-tag-option-with-resource

以下程式碼範例顯示如何使用 associate-tag-option-with-resource。

AWS CLI

將 TagOption 與資源建立關聯

下列associate-tag-option-with-resource範例會將指定的 TagOption 與指定的資源建立關聯。

```
aws servicecatalog associate-tag-option-with-resource \
    --resource-id port-2s6abcdq5wdh4 \
    --tag-option-id tag-p3abc2pkpz5qc
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateTagOptionWithResource。

copy-product

以下程式碼範例顯示如何使用 copy-product。

AWS CLI

複製產品

下列copy-product範例會使用 JSON 檔案傳遞參數，複製指定的產品。

```
aws servicecatalog copy-product --cli-input-json file://copy-product-input.json
```

copy-product-input.json 的內容：

```
{
    "SourceProductArn": "arn:aws:catalog:us-west-2:123456789012:product/prod-
tcabcd3syn2xy",
    "TargetProductName": "copy-of-myproduct",
    "CopyOptions": [
        "CopyTags"
    ]
}
```

輸出：

```
{
```

```
    "CopyProductToken": "copyproduct-abc5defgjkdji"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CopyProduct。

create-portfolio-share

以下程式碼範例顯示如何使用 create-portfolio-share。

AWS CLI

與 帳戶共用產品組合

下列create-portfolio-share範例與指定的帳戶共用指定的產品組合。

```
aws servicecatalog create-portfolio-share \
    --portfolio-id port-2s6abcdef5wdh4 \
    --account-id 794123456789
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePortfolioShare。

create-portfolio

以下程式碼範例顯示如何使用 create-portfolio。

AWS CLI

建立產品組合

下列create-portfolio範例會建立產品組合。

```
aws servicecatalog create-portfolio  \
    --provider-name my-provider \
    --display-name my-portfolio
```

輸出：

```
{
    "PortfolioDetail": {
```

```
        "ProviderName": "my-provider",
        "DisplayName": "my-portfolio",
        "CreatedTime": 1571337221.555,
        "ARN": "arn:aws:catalog:us-east-2:123456789012:portfolio/
port-2s6xmplq5wdh4",
        "Id": "port-2s6xmplq5wdh4"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePortfolio。

create-product

以下程式碼範例顯示如何使用 create-product。

AWS CLI

建立產品

下列create-product範例會使用 JSON 檔案傳遞參數來建立產品。

```
aws servicecatalog create-product \
    --cli-input-json file://create-product-input.json
```

create-product-input.json 的內容：

```
{
    "AcceptLanguage": "en",
    "Name": "test-product",
    "Owner": "test-owner",
    "Description": "test-description",
    "Distributor": "test-distributor",
    "SupportDescription": "test-support",
    "SupportEmail": "test@amazon.com",
    "SupportUrl": "https://aws.amazon.com",
    "ProductType": "CLOUD_FORMATION_TEMPLATE",
    "Tags": [
        {
            "Key": "region",
            "Value": "us-east-1"
        }
    ],
```

```
    "ProvisioningArtifactParameters": {
        "Name": "test-version-name",
        "Description": "test-version-description",
        "Info": {
            "LoadTemplateFromURL": "https://s3-us-west-1.amazonaws.com/
cloudformation-templates-us-west-1/my-cfn-template.template"
        },
        "Type": "CLOUD_FORMATION_TEMPLATE"
    }
}
```

輸出：

```
{
    "Tags": [
        {
            "Key": "region",
            "Value": "us-east-1"
        }
    ],
    "ProductViewDetail": {
        "CreatedTime": 1576025036.0,
        "ProductARN": "arn:aws:catalog:us-west-2:1234568542028:product/
prod-3p5abcdef3oyk",
        "Status": "CREATED",
        "ProductViewSummary": {
            "Type": "CLOUD_FORMATION_TEMPLATE",
            "Distributor": "test-distributor",
            "SupportUrl": "https://aws.amazon.com",
            "SupportEmail": "test@amazon.com",
            "Id": "prodview-abcd42wvx45um",
            "SupportDescription": "test-support",
            "ShortDescription": "test-description",
            "Owner": "test-owner",
            "Name": "test-product2",
            "HasDefaultPath": false,
            "ProductId": "prod-3p5abcdef3oyk"
        }
    },
    "ProvisioningArtifactDetail": {
        "CreatedTime": 1576025036.0,
        "Active": true,
        "Id": "pa-pq3p5lil12a34",
```

```
        "Description": "test-version-description",
        "Name": "test-version-name",
        "Type": "CLOUD_FORMATION_TEMPLATE"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateProduct。

create-provisioning-artifact

以下程式碼範例顯示如何使用 create-provisioning-artifact。

AWS CLI

建立佈建成品

下列create-provisioning-artifact範例使用 JSON 檔案傳遞參數，建立佈建成品。

```
aws servicecatalog create-provisioning-artifact \
   --cli-input-json file://create-provisioning-artifact-input.json
```

create-provisioning-artifact-input.json 的內容：

```
{
    "ProductId": "prod-nfi2abcdefghi",
    "Parameters": {
        "Name": "test-provisioning-artifact",
        "Description": "test description",
        "Info": {
            "LoadTemplateFromURL": "https://s3-us-west-1.amazonaws.com/
cloudformation-templates-us-west-1/my-cfn-template.template"
        },
        "Type": "CLOUD_FORMATION_TEMPLATE"
    }
}
```

輸出：

```
{
    "Info": {
        "TemplateUrl": "https://s3-us-west-1.amazonaws.com/cloudformation-templates-
us-west-1/my-cfn-template.template"
```

```
    },
    "Status": "CREATING",
    "ProvisioningArtifactDetail": {
        "Id": "pa-bb4abcdefwnaio",
        "Name": "test-provisioning-artifact",
        "Description": "test description",
        "Active": true,
        "Type": "CLOUD_FORMATION_TEMPLATE",
        "CreatedTime": 1576022545.0
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateProvisioningArtifact。

create-tag-option

以下程式碼範例顯示如何使用 create-tag-option。

AWS CLI

建立 TagOption

下列create-tag-option範例會建立 TagOption。

```
aws servicecatalog create-tag-option
    --key 1234
    --value name
```

輸出：

```
{
    "TagOptionDetail": {
    "Id": "tag-iabcdn4fzjjms",
    "Value": "name",
    "Active": true,
    "Key": "1234"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTagOption。

delete-portfolio-share

以下程式碼範例顯示如何使用 delete-portfolio-share。

AWS CLI

停止與 帳戶共用產品組合

下列delete-portfolio-share範例會停止與指定的帳戶共用產品組合。

```
aws servicecatalog delete-portfolio-share \
    --portfolio-id port-2s6abcdq5wdh4 \
    --account-id 123456789012
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePortfolioShare。

delete-portfolio

以下程式碼範例顯示如何使用 delete-portfolio。

AWS CLI

刪除產品組合

下列delete-portfolio範例會刪除指定的產品組合。

```
aws servicecatalog delete-portfolio \
    --id port-abcdlx4gox4do
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePortfolio。

delete-product

以下程式碼範例顯示如何使用 delete-product。

AWS CLI

刪除產品

下列delete-product範例會刪除指定的產品。

```
aws servicecatalog delete-product \
    --id prod-abcdcek6yhbxi
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteProduct。

delete-provisioning-artifact

以下程式碼範例顯示如何使用 delete-provisioning-artifact。

AWS CLI

刪除佈建成品

下列delete-provisioning-artifact範例會刪除指定的佈建成品。

```
aws servicecatalog delete-provisioning-artifact \
    --product-id prod-abc2uebuplcpw \
    --provisioning-artifact-id pa-pqabcddii7ouc
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteProvisioningArtifact。

delete-tag-option

以下程式碼範例顯示如何使用 delete-tag-option。

AWS CLI

刪除 TagOption

下列delete-tag-option範例會刪除指定的 TagOption。

```
aws servicecatalog delete-tag-option \
    --id tag-iabcdn4fzjjms
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTagOption。

describe-copy-product-status

以下程式碼範例顯示如何使用 describe-copy-product-status。

AWS CLI

描述複製產品操作的狀態

下列describe-copy-product-status範例顯示指定非同步複製產品操作的目前狀態。

```
aws servicecatalog describe-copy-product-status \
    --copy-product-token copyproduct-znn5tf5abcd3w
```

輸出：

```
{
    "CopyProductStatus": "SUCCEEDED",
    "TargetProductId": "prod-os6hog7abcdt2"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCopyProductStatus。

describe-portfolio

以下程式碼範例顯示如何使用 describe-portfolio。

AWS CLI

描述產品組合

下列describe-portfolio範例顯示指定產品組合的詳細資訊。

```
aws servicecatalog describe-portfolio \
    --id port-2s6abcdq5wdh4
```

輸出：

```
{
    "TagOptions": [],
```

```
    "PortfolioDetail": {
        "ARN": "arn:aws:catalog:us-west-2:687558541234:portfolio/
port-2s6abcdq5wdh4",
        "Id": "port-2s6wuzyq5wdh4",
        "CreatedTime": 1571337221.555,
        "DisplayName": "my-portfolio",
        "ProviderName": "my-provider"
    },
    "Tags": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePortfolio。

describe-product-as-admin

以下程式碼範例顯示如何使用 describe-product-as-admin。

AWS CLI

將產品描述為管理員

下列describe-product-as-admin範例使用管理員權限顯示指定產品的詳細資訊。

```
aws servicecatalog describe-product-as-admin \
    --id prod-abcdcek6yhbxi
```

輸出：

```
{
    "TagOptions": [],
    "ProductViewDetail": {
        "ProductARN": "arn:aws:catalog:us-west-2:687558542028:product/prod-
abcdcek6yhbxi",
        "ProductViewSummary": {
            "SupportEmail": "test@amazon.com",
            "Type": "CLOUD_FORMATION_TEMPLATE",
            "Distributor": "test-distributor",
            "ShortDescription": "test-description",
            "Owner": "test-owner",
            "Id": "prodview-wi3l2j4abc6vc",
            "SupportDescription": "test-support",
            "ProductId": "prod-abcdcek6yhbxi",
```

```
            "HasDefaultPath": false,
            "Name": "test-product3",
            "SupportUrl": "https://aws.amazon.com"
        },
        "CreatedTime": 1577136715.0,
        "Status": "CREATED"
    },
    "ProvisioningArtifactSummaries": [
        {
            "CreatedTime": 1577136715.0,
            "Description": "test-version-description",
            "ProvisioningArtifactMetadata": {
                "SourceProvisioningArtifactId": "pa-abcdxkkiv5fcm"
            },
            "Name": "test-version-name-3",
            "Id": "pa-abcdxkkiv5fcm"
        }
    ],
    "Tags": [
        {
            "Value": "iad",
            "Key": "region"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeProductAsAdmin。

describe-provisioned-product

以下程式碼範例顯示如何使用 describe-provisioned-product。

AWS CLI

描述佈建的產品

下列describe-provisioned-product範例顯示指定佈建產品的詳細資訊。

```
aws servicecatalog describe-provisioned-product \
    --id pp-dpom27bm4abcd
```

輸出：

```
{
    "ProvisionedProductDetail": {
        "Status": "ERROR",
        "CreatedTime": 1577222793.358,
        "Arn": "arn:aws:servicecatalog:us-west-2:123456789012:stack/mytestppname3/
pp-dpom27bm4abcd",
        "Id": "pp-dpom27bm4abcd",
        "StatusMessage": "AmazonCloudFormationException  Parameters: [KeyName]
 must have values (Service: AmazonCloudFormation; Status Code: 400; Error Code:
 ValidationError; Request ID: 5528602a-a9ef-427c-825c-f82c31b814f5)",
        "IdempotencyToken": "527c5358-2a1a-4b9e-b1b9-7293b0ddff42",
        "LastRecordId": "rec-tfuawdjovzxge",
        "Type": "CFN_STACK",
        "Name": "mytestppname3"
    },
    "CloudWatchDashboards": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeProvisionedProduct。

describe-provisioning-artifact

以下程式碼範例顯示如何使用 describe-provisioning-artifact。

AWS CLI

描述佈建成品

下列describe-provisioning-artifact範例顯示指定佈建成品的詳細資訊。

```
aws servicecatalog describe-provisioning-artifact \
    --provisioning-artifact-id pa-pcz347abcdcfm \
    --product-id prod-abcdfz3syn2rg
```

輸出：

```
{
    "Info": {
        "TemplateUrl": "https://awsdocs.s3.amazonaws.com/servicecatalog/
myexampledevelopment-environment.template"
    },
    "ProvisioningArtifactDetail": {
```

```
        "Id": "pa-pcz347abcdcfm",
        "Active": true,
        "Type": "CLOUD_FORMATION_TEMPLATE",
        "Description": "updated description",
        "CreatedTime": 1562097906.0,
        "Name": "updated name"
    },
    "Status": "AVAILABLE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeProvisioningArtifact。

describe-tag-option

以下程式碼範例顯示如何使用 describe-tag-option。

AWS CLI

描述 TagOption

下列describe-tag-option範例顯示指定 TagOption 的詳細資訊。

```
aws servicecatalog describe-tag-option \
    --id tag-p3tej2abcd5qc
```

輸出：

```
{
    "TagOptionDetail": {
        "Active": true,
        "Id": "tag-p3tej2abcd5qc",
        "Value": "value-3",
        "Key": "1234"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTagOption。

disassociate-principal-from-portfolio

以下程式碼範例顯示如何使用 disassociate-principal-from-portfolio。

AWS CLI

取消委託人與產品組合的關聯

下列disassociate-principal-from-portfolio範例會取消指定委託人與產品組合的關聯。

```
aws servicecatalog disassociate-principal-from-portfolio \
    --portfolio-id port-2s6abcdq5wdh4 \
    --principal-arn arn:aws:iam::123456789012:group/myendusers
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociatePrincipalFromPortfolio。

disassociate-product-from-portfolio

以下程式碼範例顯示如何使用 disassociate-product-from-portfolio。

AWS CLI

取消產品與產品組合的關聯

下列disassociate-product-from-portfolio範例會將指定的產品與產品組合取消關聯。

```
aws servicecatalog disassociate-product-from-portfolio \
    --product-id prod-3p5abcdmu3oyk \
    --portfolio-id port-2s6abcdq5wdh4
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateProductFromPortfolio。

disassociate-tag-option-from-resource

以下程式碼範例顯示如何使用 disassociate-tag-option-from-resource。

AWS CLI

取消 TagOption 與資源的關聯

下列disassociate-tag-option-from-resource範例會取消指定的 TagOption 與 資源的關聯。

```
aws servicecatalog disassociate-tag-option-from-resource \
     --resource-id port-2s6abcdq5wdh4 \
     --tag-option-id tag-p3abc2pkpz5qc
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateTagOptionFromResource。

list-accepted-portfolio-shares

以下程式碼範例顯示如何使用 list-accepted-portfolio-shares。

AWS CLI

列出已接受的產品組合共用

下列list-accepted-portfolio-shares範例列出此帳戶接受共用的所有產品組合，僅包括預設的 Service Catalog 產品組合。

```
aws servicecatalog list-accepted-portfolio-shares \
    --portfolio-share-type "AWS_SERVICECATALOG"
```

輸出：

```
{
    "PortfolioDetails": [
        {
            "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/port-
d2abcd5dpkuma",
            "Description": "AWS Service Catalog Reference blueprints for often-used
 AWS services such as EC2, S3, RDS, VPC and EMR.",
            "CreatedTime": 1574456190.687,
            "ProviderName": "AWS Service Catalog",
            "DisplayName": "Reference Architectures",
            "Id": "port-d2abcd5dpkuma"
        },
        {
            "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/port-
abcdefaua7zpu",
            "Description": "AWS well-architected blueprints for high reliability
 applications.",
```

```
            "CreatedTime": 1574461496.092,
            "ProviderName": "AWS Service Catalog",
            "DisplayName": "High Reliability Architectures",
            "Id": "port-abcdefaua7zpu"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAcceptedPortfolioShares。

`list-portfolio-access`

以下程式碼範例顯示如何使用 `list-portfolio-access`。

AWS CLI

列出可存取產品組合的帳戶

下列`list-portfolio-access`範例列出可存取指定產品組合 AWS 的帳戶。

```
aws servicecatalog list-portfolio-access \
    --portfolio-id port-2s6abcdq5wdh4
```

輸出：

```
{
    "AccountIds": [
        "123456789012"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPortfolioAccess。

`list-portfolios-for-product`

以下程式碼範例顯示如何使用 `list-portfolios-for-product`。

AWS CLI

列出與產品相關聯的產品組合

下列list-portfolios-for-product範例列出與指定產品相關聯的產品組合。

```
aws servicecatalog list-portfolios-for-product \
    --product-id prod-abcdfz3syn2rg
```

輸出：

```
{
    "PortfolioDetails": [
        {
            "CreatedTime": 1571337221.555,
            "Id": "port-2s6abcdq5wdh4",
            "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/
port-2s6abcdq5wdh4",
            "DisplayName": "my-portfolio",
            "ProviderName": "my-provider"
        },
        {
            "CreatedTime": 1559665256.348,
            "Id": "port-5abcd3e5st4ei",
            "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/
port-5abcd3e5st4ei",
            "DisplayName": "test",
            "ProviderName": "provider-name"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPortfoliosForProduct。

list-portfolios

以下程式碼範例顯示如何使用 list-portfolios。

AWS CLI

列出產品組合

下列list-portfolios範例列出目前區域中的 Service Catalog 產品組合。

```
aws servicecatalog list-portfolios
```

輸出：

```
{
    "PortfolioDetails": [
        {
            "CreatedTime": 1559665256.348,
            "ARN": "arn:aws:catalog:us-east-2:123456789012:portfolio/
port-5pzcxmplst4ei",
            "DisplayName": "my-portfolio",
            "Id": "port-5pzcxmplst4ei",
            "ProviderName": "my-user"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPortfolios。

list-principals-for-portfolio

以下程式碼範例顯示如何使用 list-principals-for-portfolio。

AWS CLI

列出產品組合的所有主體

下列list-principals-for-portfolio範例列出指定產品組合的所有主體。

```
aws servicecatalog list-principals-for-portfolio \
    --portfolio-id port-2s6abcdq5wdh4
```

輸出：

```
{
    "Principals": [
        {
            "PrincipalARN": "arn:aws:iam::123456789012:user/usertest",
            "PrincipalType": "IAM"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPrincipalsForPortfolio。

list-provisioning-artifacts

以下程式碼範例顯示如何使用 list-provisioning-artifacts。

AWS CLI

列出產品的所有佈建成品

下列list-provisioning-artifacts範例列出指定產品的所有佈建成品。

```
aws servicecatalog list-provisioning-artifacts \
    --product-id prod-nfi2abcdefgcpw
```

輸出：

```
{
    "ProvisioningArtifactDetails": [
        {
            "Id": "pa-abcdef54ipm6z",
            "Description": "test-version-description",
            "Type": "CLOUD_FORMATION_TEMPLATE",
            "CreatedTime": 1576021147.0,
            "Active": true,
            "Name": "test-version-name"
        },
        {
            "Id": "pa-bb4zyxwwnaio",
            "Description": "test description",
            "Type": "CLOUD_FORMATION_TEMPLATE",
            "CreatedTime": 1576022545.0,
            "Active": true,
            "Name": "test-provisioning-artifact-2"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListProvisioningArtifacts。

list-resources-for-tag-option

以下程式碼範例顯示如何使用 list-resources-for-tag-option。

AWS CLI

列出與 TagOption 相關聯的資源

下列`list-resources-for-tag-option`範例列出與指定 相關聯的資源`TagOption`。

```
aws servicecatalog list-resources-for-tag-option \
    --tag-option-id tag-p3tej2abcd5qc
```

輸出：

```
{
    "ResourceDetails": [
        {
            "ARN": "arn:aws:catalog:us-west-2:123456789012:product/prod-
abcdfz3syn2rg",
            "Name": "my product",
            "Description": "description",
            "CreatedTime": 1562097906.0,
            "Id": "prod-abcdfz3syn2rg"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourcesForTagOption。

`list-tag-options`

以下程式碼範例顯示如何使用 `list-tag-options`。

AWS CLI

下列`list-tag-options`範例列出 的所有值`TagOptions`。

```
aws servicecatalog list-tag-options
```

輸出：

```
{
    "TagOptionDetails": [
        {
```

```
            "Value": "newvalue",
            "Active": true,
            "Id": "tag-iabcdn4fzjjms",
            "Key": "1234"
        },
        {
            "Value": "value1",
            "Active": true,
            "Id": "tag-e3abcdvmwvrzy",
            "Key": "key"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTagOptions。

provision-product

以下程式碼範例顯示如何使用 provision-product。

AWS CLI

佈建產品

下列provision-product範例使用指定的佈建成品佈建指定的產品。

```
aws servicecatalog provision-product \
    --product-id prod-abcdfz3syn2rg \
    --provisioning-artifact-id pa-abc347pcsccfm \
    --provisioned-product-name "mytestppname3"
```

輸出：

```
{
    "RecordDetail": {
        "RecordId": "rec-tfuawdabcdege",
        "CreatedTime": 1577222793.362,
        "ProvisionedProductId": "pp-abcd27bm4mldq",
        "PathId": "lpv2-abcdg3jp6t5k6",
        "RecordErrors": [],
        "ProductId": "prod-abcdfz3syn2rg",
        "UpdatedTime": 1577222793.362,
```

```
        "RecordType": "PROVISION_PRODUCT",
        "ProvisionedProductName": "mytestppname3",
        "ProvisioningArtifactId": "pa-pcz347abcdcfm",
        "RecordTags": [],
        "Status": "CREATED",
        "ProvisionedProductType": "CFN_STACK"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ProvisionProduct。

reject-portfolio-share

以下程式碼範例顯示如何使用 reject-portfolio-share。

AWS CLI

拒絕產品組合共用

下列reject-portfolio-share範例會拒絕指定產品組合的產品組合共用。

```
aws servicecatalog reject-portfolio-share \
    --portfolio-id port-2s6wuabcdefghijk
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RejectPortfolioShare。

scan-provisioned-products

以下程式碼範例顯示如何使用 scan-provisioned-products。

AWS CLI

列出所有可用的佈建產品

下列scan-provisioned-products範例列出可用的佈建產品。

```
aws servicecatalog scan-provisioned-products
```

輸出：

```
{
    "ProvisionedProducts": [
        {
            "Status": "ERROR",
            "Arn": "arn:aws:servicecatalog:us-west-2:123456789012:stack/
mytestppname3/pp-abcd27bm4mldq",
            "StatusMessage": "AmazonCloudFormationException  Parameters: [KeyName]
 must have values (Service: AmazonCloudFormation; Status Code: 400; Error Code:
 ValidationError; Request ID: 5528602a-a9ef-427c-825c-f82c31b814f5)",
            "Id": "pp-abcd27bm4mldq",
            "Type": "CFN_STACK",
            "IdempotencyToken": "527c5358-2a1a-4b9e-b1b9-7293b0ddff42",
            "CreatedTime": 1577222793.358,
            "Name": "mytestppname3",
            "LastRecordId": "rec-tfuawdabcdxge"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ScanProvisionedProducts。

search-products-as-admin

以下程式碼範例顯示如何使用 search-products-as-admin。

AWS CLI

使用管理員權限搜尋產品

下列search-products-as-admin範例會使用產品組合 ID 做為篩選條件，搜尋具有管理員權限的產品。

```
aws servicecatalog search-products-as-admin \
    --portfolio-id port-5abcd3e5st4ei
```

輸出：

```
{
    "ProductViewDetails": [
        {
            "ProductViewSummary": {
```

```
                "Name": "my product",
                "Owner": "owner name",
                "Type": "CLOUD_FORMATION_TEMPLATE",
                "ProductId": "prod-abcdfz3syn2rg",
                "HasDefaultPath": false,
                "Id": "prodview-abcdmyuzv2dlu",
                "ShortDescription": "description"
            },
            "ProductARN": "arn:aws:catalog:us-west-2:123456789012:product/prod-
abcdfz3syn2rg",
            "CreatedTime": 1562097906.0,
            "Status": "CREATED"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchProductsAsAdmin。

search-provisioned-products

以下程式碼範例顯示如何使用 search-provisioned-products。

AWS CLI

搜尋佈建的產品

下列search-provisioned-products範例會使用 JSON 檔案傳遞參數，搜尋符合指定產品 ID 的佈建產品。

```
aws servicecatalog search-provisioned-products \
    --cli-input-json file://search-provisioned-products-input.json
```

search-provisioned-products-input.json 的內容：

```
{
    "Filters": {
        "SearchQuery": [
            "prod-tcjvfz3syn2rg"
        ]
    }
}
```

輸出：

```
{
    "ProvisionedProducts": [
        {
            "ProvisioningArtifactId": "pa-pcz347abcdcfm",
            "Name": "mytestppname3",
            "CreatedTime": 1577222793.358,
            "Id": "pp-abcd27bm4mldq",
            "Status": "ERROR",
            "UserArn": "arn:aws:iam::123456789012:user/cliuser",
            "StatusMessage": "AmazonCloudFormationException  Parameters: [KeyName]
 must have values (Service: AmazonCloudFormation; Status Code: 400; Error Code:
 ValidationError; Request ID: 5528602a-a9ef-427c-825c-f82c31b814f5)",
            "Arn": "arn:aws:servicecatalog:us-west-2:123456789012:stack/
mytestppname3/pp-abcd27bm4mldq",
            "Tags": [
                {
                    "Value": "arn:aws:catalog:us-west-2:123456789012:product/prod-
abcdfz3syn2rg",
                    "Key": "aws:servicecatalog:productArn"
                },
                {
                    "Value": "arn:aws:iam::123456789012:user/cliuser",
                    "Key": "aws:servicecatalog:provisioningPrincipalArn"
                },
                {
                    "Value": "value-3",
                    "Key": "1234"
                },
                {
                    "Value": "pa-pcz347abcdcfm",
                    "Key": "aws:servicecatalog:provisioningArtifactIdentifier"
                },
                {
                    "Value": "arn:aws:catalog:us-west-2:123456789012:portfolio/
port-2s6abcdq5wdh4",
                    "Key": "aws:servicecatalog:portfolioArn"
                },
                {
                    "Value": "arn:aws:servicecatalog:us-west-2:123456789012:stack/
mytestppname3/pp-abcd27bm4mldq",
                    "Key": "aws:servicecatalog:provisionedProductArn"
                }
```

```
            ],
            "IdempotencyToken": "527c5358-2a1a-4b9e-b1b9-7293b0ddff42",
            "UserArnSession": "arn:aws:iam::123456789012:user/cliuser",
            "Type": "CFN_STACK",
            "LastRecordId": "rec-tfuawdabcdxge",
            "ProductId": "prod-abcdfz3syn2rg"
        }
    ],
    "TotalResultsCount": 1
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SearchProvisionedProducts。

update-portfolio

以下程式碼範例顯示如何使用 update-portfolio。

AWS CLI

更新產品組合

下列update-portfolio範例會更新指定產品組合的名稱。

```
aws servicecatalog update-portfolio \
    --id port-5abcd3e5st4ei \
    --display-name "New portfolio name"
```

輸出：

```
{
    "PortfolioDetail": {
        "DisplayName": "New portfolio name",
        "ProviderName": "provider",
        "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/
port-5abcd3e5st4ei",
        "Id": "port-5abcd3e5st4ei",
        "CreatedTime": 1559665256.348
    },
    "Tags": []
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePortfolio。

update-product

以下程式碼範例顯示如何使用 update-product。

AWS CLI

更新產品

下列update-product範例會更新指定產品的名稱和擁有者。

```
aws servicecatalog update-product \
    --id prod-os6abc7drqlt2 \
    --name "New product name" \
    --owner "Updated product owner"
```

輸出：

```
{
    "Tags": [
        {
            "Value": "iad",
            "Key": "region"
        }
    ],
    "ProductViewDetail": {
        "ProductViewSummary": {
            "Owner": "Updated product owner",
            "ProductId": "prod-os6abc7drqlt2",
            "Distributor": "test-distributor",
            "SupportUrl": "https://aws.amazon.com",
            "Name": "New product name",
            "ShortDescription": "test-description",
            "HasDefaultPath": false,
            "Id": "prodview-6abcdgrfhvidy",
            "SupportDescription": "test-support",
            "SupportEmail": "test@amazon.com",
            "Type": "CLOUD_FORMATION_TEMPLATE"
        },
        "Status": "CREATED",
        "ProductARN": "arn:aws:catalog:us-west-2:123456789012:product/prod-
os6abc7drqlt2",
        "CreatedTime": 1577136255.0
    }
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateProduct。

update-provisioning-artifact

以下程式碼範例顯示如何使用 update-provisioning-artifact。

AWS CLI

更新佈建成品

下列update-provisioning-artifact範例會使用 JSON 檔案傳遞參數，更新指定佈建成品的名稱和描述。

```
aws servicecatalog update-provisioning-artifact \
    --cli-input-json file://update-provisioning-artifact-input.json
```

update-provisioning-artifact-input.json 的內容：

```
{
    "ProductId": "prod-abcdfz3syn2rg",
    "ProvisioningArtifactId": "pa-pcz347abcdcfm",
    "Name": "updated name",
    "Description": "updated description"
}
```

輸出：

```
{
    "Info": {
        "TemplateUrl": "https://awsdocs.s3.amazonaws.com/servicecatalog/
myexampledevelopment-environment.template"
    },
    "Status": "AVAILABLE",
    "ProvisioningArtifactDetail": {
        "Active": true,
        "Description": "updated description",
        "Id": "pa-pcz347abcdcfm",
        "Name": "updated name",
```

```
        "Type": "CLOUD_FORMATION_TEMPLATE",
        "CreatedTime": 1562097906.0
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateProvisioningArtifact。

update-tag-option

以下程式碼範例顯示如何使用 update-tag-option。

AWS CLI

更新 TagOption

下列update-tag-option範例會使用指定的 JSON 檔案TagOption更新 的值。

```
aws servicecatalog update-tag-option --cli-input-json file://update-tag-option-input.json
```

update-tag-option-input.json 的內容：

```
{
    "Id": "tag-iabcdn4fzjjms",
    "Value": "newvalue",
    "Active": true
}
```

輸出：

```
{
    "TagOptionDetail": {
        "Value": "newvalue",
        "Key": "1234",
        "Active": true,
        "Id": "tag-iabcdn4fzjjms"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateTagOption。

使用 Service Quotas範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Service Quotas 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`get-aws-default-service-quota`

以下程式碼範例顯示如何使用 `get-aws-default-service-quota`。

AWS CLI

描述預設服務配額

下列`get-aws-default-service-quota`範例顯示指定配額的詳細資訊。

```
aws service-quotas get-aws-default-service-quota \
    --service-code ec2 \
    --quota-code L-1216C47A
```

輸出：

```
{
    "Quota": {
        "ServiceCode": "ec2",
        "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",
        "QuotaArn": "arn:aws:servicequotas:us-east-2::ec2/L-1216C47A",
        "QuotaCode": "L-1216C47A",
        "QuotaName": "Running On-Demand Standard (A, C, D, H, I, M, R, T, Z)
 instances",
        "Value": 5.0,
        "Unit": "None",
        "Adjustable": true,
```

```
        "GlobalQuota": false,
        "UsageMetric": {
            "MetricNamespace": "AWS/Usage",
            "MetricName": "ResourceCount",
            "MetricDimensions": {
                "Class": "Standard/OnDemand",
                "Resource": "vCPU",
                "Service": "EC2",
                "Type": "Resource"
            },
            "MetricStatisticRecommendation": "Maximum"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAwsDefaultServiceQuota。

get-requested-service-quota-change

以下程式碼範例顯示如何使用 get-requested-service-quota-change。

AWS CLI

描述增加服務配額請求

下列get-requested-service-quota-change範例說明指定的配額增加請求。

```
aws service-quotas get-requested-service-quota-change \
    --request-id d187537d15254312a9609aa51bbf7624u7W49tPO
```

輸出：

```
{
    "RequestedQuota": {
        "Id": "d187537d15254312a9609aa51bbf7624u7W49tPO",
        "CaseId": "6780195351",
        "ServiceCode": "ec2",
        "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",
        "QuotaCode": "L-20F13EBD",
        "QuotaName": "Running Dedicated c5n Hosts",
        "DesiredValue": 2.0,
        "Status": "CASE_OPENED",
```

```
        "Created": 1580446904.067,
        "LastUpdated": 1580446953.265,
        "Requester": "{\"accountId\":\"123456789012\",\"callerArn\":
\"arn:aws:iam::123456789012:root\"}",
        "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/L-20F13EBD",
        "GlobalQuota": false,
        "Unit": "None"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRequestedServiceQuotaChange。

get-service-quota

以下程式碼範例顯示如何使用 get-service-quota。

AWS CLI

描述服務配額

下列get-service-quota範例顯示指定配額的詳細資訊。

```
aws service-quotas get-service-quota \
    --service-code ec2 \
    --quota-code L-1216C47A
```

輸出：

```
{
    "Quota": {
        "ServiceCode": "ec2",
        "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",
        "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/L-1216C47A",
        "QuotaCode": "L-1216C47A",
        "QuotaName": "Running On-Demand Standard (A, C, D, H, I, M, R, T, Z)
 instances",
        "Value": 1920.0,
        "Unit": "None",
        "Adjustable": true,
        "GlobalQuota": false,
        "UsageMetric": {
            "MetricNamespace": "AWS/Usage",
```

```
            "MetricName": "ResourceCount",
            "MetricDimensions": {
                "Class": "Standard/OnDemand",
                "Resource": "vCPU",
                "Service": "EC2",
                "Type": "Resource"
            },
            "MetricStatisticRecommendation": "Maximum"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetServiceQuota。

list-aws-default-service-quotas

以下程式碼範例顯示如何使用 list-aws-default-service-quotas。

AWS CLI

列出服務的預設配額

下列list-aws-default-service-quotas範例列出指定服務配額的預設值。

```
aws service-quotas list-aws-default-service-quotas \
    --service-code xray
```

輸出：

```
{
    "Quotas": [
        {
            "ServiceCode": "xray",
            "ServiceName": "AWS X-Ray",
            "QuotaArn": "arn:aws:servicequotas:us-west-2::xray/L-C6B6F05D",
            "QuotaCode": "L-C6B6F05D",
            "QuotaName": "Indexed annotations per trace",
            "Value": 50.0,
            "Unit": "None",
            "Adjustable": false,
            "GlobalQuota": false
        },
```

```
        {
            "ServiceCode": "xray",
            "ServiceName": "AWS X-Ray",
            "QuotaArn": "arn:aws:servicequotas:us-west-2::xray/L-D781C0FD",
            "QuotaCode": "L-D781C0FD",
            "QuotaName": "Segment document size",
            "Value": 64.0,
            "Unit": "Kilobytes",
            "Adjustable": false,
            "GlobalQuota": false
        },
        {
            "ServiceCode": "xray",
            "ServiceName": "AWS X-Ray",
            "QuotaArn": "arn:aws:servicequotas:us-west-2::xray/L-998BFF16",
            "QuotaCode": "L-998BFF16",
            "QuotaName": "Trace and service graph retention in days",
            "Value": 30.0,
            "Unit": "None",
            "Adjustable": false,
            "GlobalQuota": false
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAwsDefaultServiceQuotas。

list-requested-service-quota-change-history-by-quota

以下程式碼範例顯示如何使用 list-requested-service-quota-change-history-by-quota。

AWS CLI

列出您的配額增加請求

下列list-requested-service-quota-change-history-by-quota範例列出指定配額的配額增加請求。

```
aws service-quotas list-requested-service-quota-change-history-by-quota \
    --service-code ec2 \
    --quota-code L-20F13EBD
```

輸出：

```
{
    "RequestedQuotas": [
        {
            "Id": "d187537d15254312a9609aa51bbf7624u7W49tPO",
            "CaseId": "6780195351",
            "ServiceCode": "ec2",
            "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",
            "QuotaCode": "L-20F13EBD",
            "QuotaName": "Running Dedicated c5n Hosts",
            "DesiredValue": 2.0,
            "Status": "CASE_OPENED",
            "Created": 1580446904.067,
            "LastUpdated": 1580446953.265,
            "Requester": "{\"accountId\":\"123456789012\",\"callerArn\":
\"arn:aws:iam::123456789012:root\"}",
            "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/
L-20F13EBD",
            "GlobalQuota": false,
            "Unit": "None"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRequestedServiceQuotaChangeHistoryByQuota。

list-requested-service-quota-change-history

以下程式碼範例顯示如何使用 list-requested-service-quota-change-history。

AWS CLI

列出您的配額增加請求

下列list-requested-service-quota-change-history範例列出指定服務的配額增加請求。

```
aws service-quotas list-requested-service-quota-change-history \
    --service-code ec2
```

輸出：

```
{
    "RequestedQuotas": [
        {
            "Id": "d187537d15254312a9609aa51bbf7624u7W49tPO",
            "CaseId": "6780195351",
            "ServiceCode": "ec2",
            "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",
            "QuotaCode": "L-20F13EBD",
            "QuotaName": "Running Dedicated c5n Hosts",
            "DesiredValue": 2.0,
            "Status": "CASE_OPENED",
            "Created": 1580446904.067,
            "LastUpdated": 1580446953.265,
            "Requester": "{\"accountId\":\"123456789012\",\"callerArn\":
\"arn:aws:iam::123456789012:root\"}",
            "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/
L-20F13EBD",
            "GlobalQuota": false,
            "Unit": "None"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRequestedServiceQuotaChangeHistory。

`list-service-quotas`

以下程式碼範例顯示如何使用 `list-service-quotas`。

AWS CLI

列出服務的配額

下列`list-service-quotas`範例顯示 AWS CloudFormation 配額的詳細資訊。

```
aws service-quotas list-service-quotas \
    --service-code cloudformation
```

輸出：

```
{
    "Quotas": [
        {
            "ServiceCode": "cloudformation",
            "ServiceName": "AWS CloudFormation",
            "QuotaArn": "arn:aws:servicequotas:us-
east-2:123456789012:cloudformation/L-87D14FB7",
            "QuotaCode": "L-87D14FB7",
            "QuotaName": "Output count in CloudFormation template",
            "Value": 60.0,
            "Unit": "None",
            "Adjustable": false,
            "GlobalQuota": false
        },
        {
            "ServiceCode": "cloudformation",
            "ServiceName": "AWS CloudFormation",
            "QuotaArn": "arn:aws:servicequotas:us-
east-2:123456789012:cloudformation/L-0485CB21",
            "QuotaCode": "L-0485CB21",
            "QuotaName": "Stack count",
            "Value": 200.0,
            "Unit": "None",
            "Adjustable": true,
            "GlobalQuota": false
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListServiceQuotas。

list-services

以下程式碼範例顯示如何使用 `list-services`。

AWS CLI

列出可用的服務

下列命令列出 Service Quotas 中可用的服務。

```
aws service-quotas list-services
```

輸出：

```
{
    "Services": [
        {
            "ServiceCode": "AWSCloudMap",
            "ServiceName": "AWS Cloud Map"
        },
        {
            "ServiceCode": "access-analyzer",
            "ServiceName": "Access Analyzer"
        },
        {
            "ServiceCode": "acm",
            "ServiceName": "AWS Certificate Manager (ACM)"
        },

        ...truncated...

        {
            "ServiceCode": "xray",
            "ServiceName": "AWS X-Ray"
        }
    ]
}
```

您可以新增 `--query` 參數，將顯示篩選為您感興趣的資訊。下列範例只會顯示服務代碼。

```
aws service-quotas list-services \
    --query Services[*].ServiceCode
```

輸出：

```
[
    "AWSCloudMap",
    "access-analyzer",
    "acm",
    "acm-pca",
    "amplify",
    "apigateway",
    "application-autoscaling",
        ...truncated...
```

```
    "xray"
]
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListServices。

request-service-quota-increase

以下程式碼範例顯示如何使用 request-service-quota-increase。

AWS CLI

請求提高服務配額

下列request-service-quota-increase範例會請求增加指定的服務配額。

```
aws service-quotas request-service-quota-increase \
    --service-code ec2 \
    --quota-code L-20F13EBD \
    --desired-value 2
```

輸出：

```
{
    "RequestedQuota": {
        "Id": "d187537d15254312a9609aa51bbf7624u7W49tPO",
        "ServiceCode": "ec2",
        "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",
        "QuotaCode": "L-20F13EBD",
        "QuotaName": "Running Dedicated c5n Hosts",
        "DesiredValue": 2.0,
        "Status": "PENDING",
        "Created": 1580446904.067,
        "Requester": "{\"accountId\":\"123456789012\",\"callerArn\":
\"arn:aws:iam::123456789012:root\"}",
        "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/L-20F13EBD",
        "GlobalQuota": false,
        "Unit": "None"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RequestServiceQuotaIncrease。

使用 的 Amazon SES 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon SES 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`delete-identity`

以下程式碼範例顯示如何使用 `delete-identity`。

AWS CLI

刪除身分

下列範例使用 `delete-identity` 命令從透過 Amazon SES 驗證的身分清單中刪除身分：

```
aws ses delete-identity --identity user@example.com
```

如需驗證的身分詳細資訊，請參閱《Amazon Simple Email Service 開發人員指南》中的「在 Amazon SES 中驗證電子郵件地址和網域」。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteIdentity。

`get-identity-dkim-attributes`

以下程式碼範例顯示如何使用 `get-identity-dkim-attributes`。

AWS CLI

取得身分清單的 Amazon SES Easy DKIM 屬性

下列範例使用 get-identity-dkim-attributes命令來擷取身分清單的 Amazon SES Easy DKIM 屬性：

```
aws ses get-identity-dkim-attributes --identities "example.com" "user@example.com"
```

輸出：

```
{
   "DkimAttributes": {
       "example.com": {
           "DkimTokens": [
               "EXAMPLEjcs5xoyqytjsotsijas7236gr",
               "EXAMPLEjr76cvoc6mysspnioorxsn6ep",
               "EXAMPLEkbmkqkhlm2lyz77ppkulerm4k"
           ],
           "DkimEnabled": true,
           "DkimVerificationStatus": "Success"
       },
       "user@example.com": {
           "DkimEnabled": false,
           "DkimVerificationStatus": "NotStarted"
       }
   }
}
```

如果您使用從未提交進行驗證的身分來呼叫此命令，則該身分不會出現在輸出中。

如需 Easy DKIM 的詳細資訊，請參閱《Amazon Simple Email Service 開發人員指南》中的 Amazon SES 中的 Easy DKIM。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIdentityDkimAttributes。

get-identity-notification-attributes

以下程式碼範例顯示如何使用 get-identity-notification-attributes。

AWS CLI

取得身分清單的 Amazon SES 通知屬性

下列範例使用 get-identity-notification-attributes命令來擷取身分清單的 Amazon SES 通知屬性：

```
aws ses get-identity-notification-attributes --
identities "user1@example.com" "user2@example.com"
```

輸出：

```
{
   "NotificationAttributes": {
       "user1@example.com": {
           "ForwardingEnabled": false,
           "ComplaintTopic": "arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic",
           "BounceTopic": "arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic",
           "DeliveryTopic": "arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic"
       },
       "user2@example.com": {
           "ForwardingEnabled": true
       }
   }
}
```

此命令會傳回電子郵件意見回饋轉送的狀態，如果適用，也會傳回退信、投訴和交付通知所傳送之 Amazon SNS 主題的 Amazon Resource Name (ARNs)。

如果您使用從未提交進行驗證的身分來呼叫此命令，則該身分不會出現在輸出中。

如需通知的詳細資訊，請參閱《Amazon Simple Email Service 開發人員指南》中的搭配 Amazon SES 使用通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIdentityNotificationAttributes。

get-identity-verification-attributes

以下程式碼範例顯示如何使用 get-identity-verification-attributes。

AWS CLI

取得身分清單的 Amazon SES 驗證狀態

下列範例使用 get-identity-verification-attributes 命令來擷取身分清單的 Amazon SES 驗證狀態：

```
aws ses get-identity-verification-attributes --
identities "user1@example.com" "user2@example.com"
```

輸出：

```
{
   "VerificationAttributes": {
       "user1@example.com": {
           "VerificationStatus": "Success"
       },
       "user2@example.com": {
           "VerificationStatus": "Pending"
       }
   }
}
```

如果您使用從未提交進行驗證的身分來呼叫此命令，則該身分不會出現在輸出中。

如需驗證的身分詳細資訊，請參閱《Amazon Simple Email Service 開發人員指南》中的「在 Amazon SES 中驗證電子郵件地址和網域」。

- 如需 API 詳細資訊，請參閱《AWS CLI API 參考》中的 GetIdentityVerificationAttributes。

get-send-quota

以下程式碼範例顯示如何使用 get-send-quota。

AWS CLI

取得您的 Amazon SES 傳送限制

下列範例使用 get-send-quota 命令來傳回 Amazon SES 傳送限制：

```
aws ses get-send-quota
```

輸出：

```
{
   "Max24HourSend": 200.0,
   "SentLast24Hours": 1.0,
   "MaxSendRate": 1.0
}
```

Max24HourSend 是您的傳送配額，也就是您在 24 小時期間內可傳送的電子郵件數量上限。傳送配額所反映的是一段時間內的配額。每當您嘗試傳送電子郵件時，Amazon SES 會檢查您在過去的

24 小時內傳送的電子郵件數量。只要您已傳送的電子郵件總數量低於您的配額，您的傳送請求將被接受並將傳送您的電子郵件。

SentLast24Hours 是您在過去 24 小時內傳送的電子郵件數量。

MaxSendRate 是您每秒可傳送的電子郵件數量上限。

請注意，傳送限制依據收件人而定，而非訊息。例如，一封電子郵件中有 10 個收件人，就會佔用 10 個您的傳送配額。

如需詳細資訊，請參閱《Amazon Simple Email Service 開發人員指南》中的「管理您的 Amazon SES 傳送限制」。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetSendQuota。

get-send-statistics

以下程式碼範例顯示如何使用 get-send-statistics。

AWS CLI

若要取得 Amazon SES 傳送統計資料

下列範例使用 get-send-statistics命令來傳回 Amazon SES 傳送統計資料

```
aws ses get-send-statistics
```

輸出：

```
{
   "SendDataPoints": [
       {
           "Complaints": 0,
           "Timestamp": "2013-06-12T19:32:00Z",
           "DeliveryAttempts": 2,
           "Bounces": 0,
           "Rejects": 0
       },
       {
           "Complaints": 0,
           "Timestamp": "2013-06-12T00:47:00Z",
           "DeliveryAttempts": 1,
```

```
            "Bounces": 0,
            "Rejects": 0
        }
    ]
}
```

結果是資料點的清單，代表過去兩週的傳送活動。清單中的每個資料點都包含間隔 15 分鐘的統計資料。

在此範例中，只有兩個資料點，因為使用者在過去兩週內傳送的唯一電子郵件在兩個 15 分鐘內下降。

如需詳細資訊，請參閱《Amazon SES Simple Email Service 開發人員指南》中的監控您的 Amazon SES 用量統計資料。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSendStatistics。

`list-identities`

以下程式碼範例顯示如何使用 `list-identities`。

AWS CLI

列出特定 AWS 帳戶的所有身分 （電子郵件地址和網域）

下列範例使用 `list-identities` 命令來列出已提交給 Amazon SES 驗證的所有身分：

```
aws ses list-identities
```

輸出：

```
{
    "Identities": [
      "user@example.com",
      "example.com"
    ]
}
```

傳回的清單包含所有身分，無論驗證狀態為何 (已驗證、等待驗證、失敗等)。

在此範例中，因為我們未指定 identity-type 參數，所以會傳回電子郵件地址 和 網域。

如需驗證詳細資訊，請參閱《Amazon Simple Email Service 開發人員指南》中的「在 Amazon SES 中驗證電子郵件地址和網域」。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListIdentities。

`send-email`

以下程式碼範例顯示如何使用 `send-email`。

AWS CLI

使用 Amazon SES 傳送格式化電子郵件

下列範例使用 `send-email` 命令來傳送格式化電子郵件：

```
aws ses send-email --from sender@example.com --destination file://destination.json
 --message file://message.json
```

輸出：

```
{
    "MessageId": "EXAMPLEf3a5efcd1-51adec81-d2a4-4e3f-9fe2-5d85c1b23783-000000"
}
```

目的地和訊息是在目前目錄中以 .json 檔案形式儲存的 JSON 資料結構。這些檔案如下：

`destination.json`:

```
{
  "ToAddresses":  ["recipient1@example.com", "recipient2@example.com"],
  "CcAddresses":  ["recipient3@example.com"],
  "BccAddresses": []
}
```

`message.json`:

```
{
    "Subject": {
        "Data": "Test email sent using the AWS CLI",
        "Charset": "UTF-8"
    },
    "Body": {
```

```
        "Text": {
            "Data": "This is the message body in text format.",
            "Charset": "UTF-8"
        },
        "Html": {
            "Data": "This message body contains HTML formatting. It can, for example,
 contain links like this one: <a class=\"ulink\" href=\"http://docs.aws.amazon.com/
ses/latest/DeveloperGuide\" target=\"_blank\">Amazon SES Developer Guide</a>.",
            "Charset": "UTF-8"
        }
    }
}
```

將寄件者和收件者的電子郵件地址取代為您要使用的地址。請注意，必須透過 Amazon SES 驗證寄件者的電子郵件地址。在您取得 Amazon SES 生產存取權之前，除非收件者是 Amazon SES 信箱模擬器，否則您還必須驗證每個收件者的電子郵件地址。如需驗證詳細資訊，請參閱《Amazon Simple Email Service 開發人員指南》中的「在 Amazon SES 中驗證電子郵件地址和網域」。

輸出中的訊息 ID 表示對 send-email 的呼叫成功。

如果您沒有收到電子郵件，請檢查您的垃圾郵件匣。

如需傳送格式化電子郵件的詳細資訊，請參閱《Amazon Simple Email Service 開發人員指南》中「使用 Amazon SES API 傳送格式化電子郵件」。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 SendEmail。

send-raw-email

以下程式碼範例顯示如何使用 `send-raw-email`。

AWS CLI

使用 Amazon SES 傳送電子郵件原始碼

下列範例使用 `send-raw-email` 命令來傳送包含 TXT 附件的電子郵件：

```
aws ses send-raw-email --raw-message file://message.json
```

輸出：

```
{
```

```
    "MessageId": "EXAMPLEf3f73d99b-c63fb06f-d263-41f8-a0fb-d0dc67d56c07-000000"
}
```

原始訊息是在目前目錄中以名為 message.json 的檔案儲存的 JSON 資料結構。其中包含下列各項：

```
{
    "Data": "From: sender@example.com\nTo: recipient@example.com\nSubject: Test email
 sent using the AWS CLI (contains an attachment)\nMIME-Version: 1.0\nContent-type:
 Multipart/Mixed; boundary=\"NextPart\"\n\n--NextPart\nContent-Type: text/plain
\n\nThis is the message body.\n\n--NextPart\nContent-Type: text/plain;\nContent-
Disposition: attachment; filename=\"attachment.txt\"\n\nThis is the text in the
 attachment.\n\n--NextPart--"
}
```

如您所見，「資料」是一個長字串，其中包含 MIME 格式的整個原始電子郵件內容，包括名為 attachment.txt 的附件。

將 sender@example.com 和 recipient@example.com 取代為您要使用的地址。請注意，必須透過 Amazon SES 驗證寄件者的電子郵件地址。在您取得 Amazon SES 生產存取權之前，除非收件者是 Amazon SES 信箱模擬器，否則您還必須驗證收件者的電子郵件地址。如需驗證詳細資訊，請參閱《Amazon Simple Email Service 開發人員指南》中的「在 Amazon SES 中驗證電子郵件地址和網域」。

輸出中的訊息 ID 表示對 send-raw-email 的呼叫成功。

如果您沒有收到電子郵件，請檢查您的垃圾郵件匣。

如需傳送電子郵件原始碼的詳細資訊，請參閱《Amazon Simple Email Service 開發人員指南》中的「使用 Amazon SES API 傳送電子郵件原始碼」。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 SendRawEmail。

set-identity-dkim-enabled

以下程式碼範例顯示如何使用 set-identity-dkim-enabled。

AWS CLI

啟用或停用 Amazon SES 驗證身分的 Easy DKIM

下列範例使用 set-identity-dkim-enabled命令來停用已驗證電子郵件地址的 DKIM：

```
aws ses set-identity-dkim-enabled --identity user@example.com --no-dkim-enabled
```

如需 Easy DKIM 的詳細資訊，請參閱《Amazon Simple Email Service 開發人員指南》中的 Amazon SES 中的 Easy DKIM。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetIdentityDkimEnabled。

set-identity-feedback-forwarding-enabled

以下程式碼範例顯示如何使用 set-identity-feedback-forwarding-enabled。

AWS CLI

啟用或停用 Amazon SES 驗證身分的退信和投訴電子郵件意見回饋轉送

下列範例使用 set-identity-feedback-forwarding-enabled命令來啟用已驗證的電子郵件地址，以透過電子郵件接收退信和投訴通知：

```
aws ses set-identity-feedback-forwarding-enabled --identity user@example.com --
forwarding-enabled
```

您必須透過 Amazon SNS 或電子郵件意見回饋轉送來接收退信和投訴通知，因此只有在您同時為退信和投訴通知選取 Amazon SNS 主題時，才能停用電子郵件意見回饋轉送。

如需通知的詳細資訊，請參閱《Amazon Simple Email Service 開發人員指南》中的搭配 Amazon SES 使用通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetIdentityFeedbackForwardingEnabled。

set-identity-notification-topic

以下程式碼範例顯示如何使用 set-identity-notification-topic。

AWS CLI

設定 Amazon SES 將發佈已驗證身分的退信、投訴和/或交付通知的 Amazon SNS 主題 Amazon SES

下列範例使用 set-identity-notification-topic命令來指定已驗證電子郵件地址將收到退信通知的 Amazon SNS 主題：

```
aws ses set-identity-notification-topic --identity user@example.com --notification-
type Bounce --sns-topic arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic
```

如需通知的詳細資訊，請參閱《Amazon Simple Email Service 開發人員指南》中的搭配 Amazon SES 使用通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetIdentityNotificationTopic。

verify-domain-dkim

以下程式碼範例顯示如何使用 verify-domain-dkim。

AWS CLI

產生已驗證網域的 DKIM 權杖以使用 Amazon SES 進行 DKIM 簽署

下列範例使用 verify-domain-dkim命令為已使用 Amazon SES 驗證的網域產生 DKIM 字符：

```
aws ses verify-domain-dkim --domain example.com
```

輸出：

```
{
   "DkimTokens": [
       "EXAMPLEq76owjnks3lnluwg65scbemvw",
       "EXAMPLEi3dnsj67hstzaj673klariwx2",
       "EXAMPLEwfbtcukvimehexktmdtaz6naj"
   ]
}
```

若要設定 DKIM，您必須使用傳回的 DKIM 權杖，以指向 Amazon SES 託管的 DKIM 公有金鑰的 CNAME 記錄來更新網域的 DNS 設定。如需詳細資訊，請參閱《Amazon Simple Email Service 開發人員指南》中的 Amazon SES 中的 Easy DKIM。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 VerifyDomainDkim。

verify-domain-identity

以下程式碼範例顯示如何使用 verify-domain-identity。

AWS CLI

透過 Amazon SES 驗證網域

下列範例會使用 verify-domain-identity 命令來驗證網域：

```
aws ses verify-domain-identity --domain example.com
```

輸出：

```
{
   "VerificationToken": "eoEmxw+YaYhb3h3iVJHuXMJXqeu1q1/wwmvjuEXAMPLE"
}
```

若要完成網域驗證，您必須將帶有傳回驗證 Token 的 TXT 記錄新增至網域的 DNS 設定。如需詳細資訊，請參閱《Amazon Simple Email Service 開發人員指南》中的「在 Amazon SES 中驗證網域」。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 VerifyDomainIdentity。

verify-email-identity

以下程式碼範例顯示如何使用 verify-email-identity。

AWS CLI

透過 Amazon SES 驗證電子郵件地址

下列範例會使用 verify-email-identity 命令來驗證網域：

```
aws ses verify-email-identity --email-address user@example.com
```

您必須先驗證您要用於傳送電子郵件的寄件地址或網域來證明您擁有該地址或網域，才可使用 Amazon SES 傳送電子郵件。如果您尚未擁有生產存取權，除了由 Amazon SES 信箱模擬器提供的電子郵件地址外，您仍需驗證任何您傳送電子郵件的收件電子郵件地址。

呼叫 verify-email-identity 之後，電子郵件地址會收到一封驗證電子郵件。使用者必須按一下電子郵件中的連結，以完成驗證程序。

如需詳細資訊，請參閱《Amazon Simple Email Service 開發人員指南》中的「在 Amazon SES 中驗證電子郵件地址」。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 VerifyEmailIdentity。

使用 的 Shield 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Shield 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-drt-log-bucket`

以下程式碼範例顯示如何使用 `associate-drt-log-bucket`。

AWS CLI

授權 DRT 存取 Amazon S3 儲存貯體

下列`associate-drt-log-bucket`範例會在 DRT 與指定的 S3 儲存貯體之間建立關聯。這可讓 DRT 代表帳戶存取儲存貯體。：

```
aws shield associate-drt-log-bucket \
    --log-bucket flow-logs-for-website-lb
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Shield 進階開發人員指南》中的授權 DDoS 回應團隊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateDrtLogBucket。

`associate-drt-role`

以下程式碼範例顯示如何使用 `associate-drt-role`。

AWS CLI

授權 DRT 代表您緩解潛在的攻擊

下列associate-drt-role範例會建立 DRT 與指定角色之間的關聯。DRT 可以使用 角色來存取和管理帳戶。

```
aws shield associate-drt-role \
    --role-arn arn:aws:iam::123456789012:role/service-role/DrtRole
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Shield 進階開發人員指南》中的授權 DDoS 回應團隊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateDrtRole。

create-protection

以下程式碼範例顯示如何使用 create-protection。

AWS CLI

為單一 AWS 資源啟用 AWS Shield Advanced 保護

下列create-protection範例會啟用指定 AWS CloudFront 分佈的 Shield Advanced 保護。

```
aws shield create-protection \
    --name "Protection for CloudFront distribution" \
    --resource-arn arn:aws:cloudfront::123456789012:distribution/E198WC25FXOWY8
```

輸出：

```
{
    "ProtectionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

如需詳細資訊，請參閱《AWS Shield 進階開發人員指南》中的指定要保護的資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateProtection。

create-subscription

以下程式碼範例顯示如何使用 create-subscription。

AWS CLI

為 帳戶啟用 AWS Shield Advanced 保護

下列create-subscription範例會啟用帳戶的 Shield Advanced 保護。

```
aws shield create-subscription
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Shield Advanced AWS 開發人員指南》AWS 中的 Shield Advanced 入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSubscription。

delete-protection

以下程式碼範例顯示如何使用 delete-protection。

AWS CLI

從 AWS 資源移除 AWS Shield Advanced 保護

下列delete-protection範例會移除指定的 AWS Shield Advanced 保護。

```
aws shield delete-protection \
    --protection-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Shield Advanced AWS 開發人員指南》中的從 AWS 資源移除 Shield Advanced。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteProtection。

describe-attack

以下程式碼範例顯示如何使用 describe-attack。

AWS CLI

擷取攻擊的詳細說明

下列describe-attack範例顯示具有指定攻擊 ID 的 DDoS 攻擊詳細資訊。您可以執行 list-attacks命令來取得攻擊 IDs。

```
aws shield describe-attack --attack-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

輸出：

```
{
    "Attack": {
        "AttackId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "ResourceArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/testElb",
        "SubResources": [
            {
                "Type": "IP",
                "Id": "192.0.2.2",
                "AttackVectors": [
                    {
                        "VectorType": "SYN_FLOOD",
                        "VectorCounters": [
                            {
                                "Name": "SYN_FLOOD_BPS",
                                "Max": 982184.0,
                                "Average": 982184.0,
                                "Sum": 11786208.0,
                                "N": 12,
                                "Unit": "BPS"
                            }
                        ]
                    }
                ],
                "Counters": []
            },
            {
                "Type": "IP",
                "Id": "192.0.2.3",
                "AttackVectors": [
                    {
                        "VectorType": "SYN_FLOOD",
                        "VectorCounters": [
                            {
                                "Name": "SYN_FLOOD_BPS",
                                "Max": 982184.0,
```

```
                        "Average": 982184.0,
                        "Sum": 9821840.0,
                        "N": 10,
                        "Unit": "BPS"
                    }
                ]
            }
        ],
        "Counters": []
    },
    {
        "Type": "IP",
        "Id": "192.0.2.4",
        "AttackVectors": [
            {
                "VectorType": "SYN_FLOOD",
                "VectorCounters": [
                    {
                        "Name": "SYN_FLOOD_BPS",
                        "Max": 982184.0,
                        "Average": 982184.0,
                        "Sum": 7857472.0,
                        "N": 8,
                        "Unit": "BPS"
                    }
                ]
            }
        ],
        "Counters": []
    },
    {
        "Type": "IP",
        "Id": "192.0.2.5",
        "AttackVectors": [
            {
                "VectorType": "SYN_FLOOD",
                "VectorCounters": [
                    {
                        "Name": "SYN_FLOOD_BPS",
                        "Max": 982184.0,
                        "Average": 982184.0,
                        "Sum": 1964368.0,
                        "N": 2,
                        "Unit": "BPS"
```

```
                        }
                    ]
                }
            ],
            "Counters": []
        },
        {
            "Type": "IP",
            "Id": "2001:DB8::bcde:4321:8765:0:0",
            "AttackVectors": [
                {
                    "VectorType": "SYN_FLOOD",
                    "VectorCounters": [
                        {
                            "Name": "SYN_FLOOD_BPS",
                            "Max": 982184.0,
                            "Average": 982184.0,
                            "Sum": 1964368.0,
                            "N": 2,
                            "Unit": "BPS"
                        }
                    ]
                }
            ],
            "Counters": []
        },
        {
            "Type": "IP",
            "Id": "192.0.2.6",
            "AttackVectors": [
                {
                    "VectorType": "SYN_FLOOD",
                    "VectorCounters": [
                        {
                            "Name": "SYN_FLOOD_BPS",
                            "Max": 982184.0,
                            "Average": 982184.0,
                            "Sum": 1964368.0,
                            "N": 2,
                            "Unit": "BPS"
                        }
                    ]
                }
            ],
```

```
                "Counters": []
            }
        ],
        "StartTime": 1576024927.457,
        "EndTime": 1576025647.457,
        "AttackCounters": [],
        "AttackProperties": [
            {
                "AttackLayer": "NETWORK",
                "AttackPropertyIdentifier": "SOURCE_IP_ADDRESS",
                "TopContributors": [
                    {
                        "Name": "198.51.100.5",
                        "Value": 2024475682
                    },
                    {
                        "Name": "198.51.100.8",
                        "Value": 1311380863
                    },
                    {
                        "Name": "203.0.113.4",
                        "Value": 900599855
                    },
                    {
                        "Name": "198.51.100.4",
                        "Value": 769417366
                    },
                    {
                        "Name": "203.1.113.13",
                        "Value": 757992847
                    }
                ],
                "Unit": "BYTES",
                "Total": 92773354841
            },
            {
                "AttackLayer": "NETWORK",
                "AttackPropertyIdentifier": "SOURCE_COUNTRY",
                "TopContributors": [
                    {
                        "Name": "United States",
                        "Value": 80938161764
                    },
                    {
```

```
                    "Name": "Brazil",
                    "Value": 9929864330
                },
                {
                    "Name": "Netherlands",
                    "Value": 1635009446
                },
                {
                    "Name": "Mexico",
                    "Value": 144832971
                },
                {
                    "Name": "Japan",
                    "Value": 45369000
                }
            ],
            "Unit": "BYTES",
            "Total": 92773354841
        },
        {
            "AttackLayer": "NETWORK",
            "AttackPropertyIdentifier": "SOURCE_ASN",
            "TopContributors": [
                {
                    "Name": "12345",
                    "Value": 74953625841
                },
                {
                    "Name": "12346",
                    "Value": 4440087595
                },
                {
                    "Name": "12347",
                    "Value": 1635009446
                },
                {
                    "Name": "12348",
                    "Value": 1221230000
                },
                {
                    "Name": "12349",
                    "Value": 1199425294
                }
            ],
```

```
                "Unit": "BYTES",
                "Total": 92755479921
            }
        ],
        "Mitigations": []
    }
}
```

如需詳細資訊，請參閱《AWS Shield 進階開發人員指南》中的檢閱 DDoS 事件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAttack。

describe-drt-access

以下程式碼範例顯示如何使用 describe-drt-access。

AWS CLI

若要擷取 DRT 必須代表您緩解攻擊的授權描述

下列describe-drt-access範例會擷取 DRT 擁有的角色和 S3 儲存貯體授權，讓它代表您回應潛在的攻擊。

```
aws shield describe-drt-access
```

輸出：

```
{
    "RoleArn": "arn:aws:iam::123456789012:role/service-role/DrtRole",
    "LogBucketList": [
        "flow-logs-for-website-lb"
    ]
}
```

如需詳細資訊，請參閱《AWS Shield 進階開發人員指南》中的授權 DDoS 回應團隊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDrtAccess。

describe-emergency-contact-settings

以下程式碼範例顯示如何使用 describe-emergency-contact-settings。

AWS CLI

擷取您使用 DRT 存檔的緊急電子郵件地址

下列`describe-emergency-contact-settings`範例會擷取 帳戶使用 DRT 存檔的電子郵件地址。這些是 DRT 在回應可疑攻擊時應聯絡的地址。

```
aws shield describe-emergency-contact-settings
```

輸出：

```
{
    "EmergencyContactList": [
        {
            "EmailAddress": "ops@example.com"
        },
        {
            "EmailAddress": "ddos-notifications@example.com"
       }
    ]
}
```

如需詳細資訊，請參閱《Shield AWS AWS 進階開發人員指南》中的 Shield 如何運作<https://docs.aws.amazon.com/waf/latest/developerguide/ddos-overview.html>。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeEmergencyContactSettings。

`describe-protection`

以下程式碼範例顯示如何使用 `describe-protection`。

AWS CLI

擷取 Shield Advanced AWS 保護的詳細資訊

下列`describe-protection`範例顯示具有指定 ID 的 Shield Advanced 保護的詳細資訊。您可以執行 `list-protections`命令來取得保護 IDs。

```
aws shield describe-protection \
    --protection-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "Protection": {
        "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Name": "1.2.3.4",
        "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:eip-allocation/
eipalloc-0ac1537af40742a6d"
    }
}
```

如需詳細資訊，請參閱《AWS Shield 進階開發人員指南》中的指定要保護的資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeProtection。

describe-subscription

以下程式碼範例顯示如何使用 describe-subscription。

AWS CLI

擷取帳戶 Shield Advanced AWS 保護的詳細資訊

下列describe-subscription範例顯示針對帳戶提供的 Shield Advanced 保護的詳細資訊：

```
aws shield describe-subscription
```

輸出：

```
{
    "Subscription": {
        "StartTime": 1534368978.0,
        "EndTime": 1597613778.0,
        "TimeCommitmentInSeconds": 63244800,
        "AutoRenew": "ENABLED",
        "Limits": [
            {
                "Type": "GLOBAL_ACCELERATOR",
                "Max": 1000
            },
            {
                "Type": "ROUTE53_HOSTED_ZONE",
                "Max": 1000
            },
```

```
            {
                "Type": "CF_DISTRIBUTION",
                "Max": 1000
            },
            {
                "Type": "ELB_LOAD_BALANCER",
                "Max": 1000
            },
            {
                "Type": "EC2_ELASTIC_IP_ALLOCATION",
                "Max": 1000
            }
        ]
    }
}
```

如需詳細資訊，請參閱《Shield AWS 進階開發人員指南》中的 Shield 如何運作。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSubscription。

disassociate-drt-log-bucket

以下程式碼範例顯示如何使用 disassociate-drt-log-bucket。

AWS CLI

移除 DRT 代表您存取 Amazon S3 儲存貯體的授權

下列disassociate-drt-log-bucket範例會移除 DRT 與指定 S3 儲存貯體之間的關聯。此命令完成後，DRT 就無法再代表帳戶存取儲存貯體。

```
aws shield disassociate-drt-log-bucket \
    --log-bucket flow-logs-for-website-lb
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Shield 進階開發人員指南》中的授權 DDoS 回應團隊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateDrtLogBucket。

disassociate-drt-role

以下程式碼範例顯示如何使用 disassociate-drt-role。

AWS CLI

移除 DRT 的授權，以代表您緩解潛在的攻擊

下列disassociate-drt-role範例會移除 DRT 與 帳戶之間的關聯。在此呼叫之後，DRT 就無法再存取或管理您的帳戶。

```
aws shield disassociate-drt-role
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Shield 進階開發人員指南》中的授權 DDoS 回應團隊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateDrtRole。

get-subscription-state

以下程式碼範例顯示如何使用 get-subscription-state。

AWS CLI

擷取帳戶 Shield Advanced AWS 訂閱的目前狀態

下列get-subscription-state範例會擷取帳戶的 Shield Advanced 保護狀態。

```
aws shield get-subscription-state
```

輸出：

```
{
    "SubscriptionState": "ACTIVE"
}
```

如需詳細資訊，請參閱《Shield AWS 進階開發人員指南》中的 Shield 如何運作。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSubscriptionState。

list-attacks

以下程式碼範例顯示如何使用 list-attacks。

AWS CLI

從 Shield Advanced AWS 擷取攻擊摘要

下列`list-attacks`範例會擷取指定期間內指定 AWS CloudFront 分佈的攻擊摘要。回應包含您可以提供給 `describe-attack`命令IDs，以取得攻擊的詳細資訊。

```
aws shield list-attacks \
    --resource-arns arn:aws:cloudfront::12345678910:distribution/E1PXMP22ZVFAOR \
    --start-time FromInclusive=1529280000,ToExclusive=1529300000
```

輸出：

```
{
    "AttackSummaries": [
        {
            "AttackId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "ResourceArn": "arn:aws:cloudfront::123456789012:distribution/
E1PXMP22ZVFAOR",
            "StartTime": 1529280000.0,
            "EndTime": 1529449200.0,
            "AttackVectors": [
                {
                    "VectorType": "SYN_FLOOD"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Shield 進階開發人員指南》中的檢閱 DDoS 事件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAttacks。

`list-protections`

以下程式碼範例顯示如何使用 `list-protections`。

AWS CLI

從 Shield Advanced AWS 擷取保護摘要

下列list-protections範例會擷取為帳戶啟用的保護摘要。

```
aws shield list-protections
```

輸出：

```
{
    "Protections": [
        {
            "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Name": "Protection for CloudFront distribution",
            "ResourceArn": "arn:aws:cloudfront::123456789012:distribution/
E198WC25FXOWY8"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Shield 進階開發人員指南》中的指定要保護的資源。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListProtections。

update-emergency-contact-settings

以下程式碼範例顯示如何使用 update-emergency-contact-settings。

AWS CLI

使用 DRT 定義存檔的緊急電子郵件地址

下列update-emergency-contact-settings範例定義兩個電子郵件地址，DRT 應在回應可疑攻擊時聯絡該地址。

```
aws shield update-emergency-contact-settings \
        --emergency-contact-list EmailAddress=ops@example.com EmailAddress=ddos-
notifications@example.com
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Shield AWS 進階開發人員指南》中的 Shield 如何運作。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateEmergencyContactSettings。

update-subscription

以下程式碼範例顯示如何使用 update-subscription。

AWS CLI

修改帳戶的 AWS Shield Advanced 訂閱

下列update-subscription範例會啟用帳戶的 Shield Advanced AWS 訂閱自動續約。

```
aws shield update-subscription \
    --auto-renew ENABLED
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Shield AWS 進階開發人員指南》中的 Shield 如何運作。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSubscription。

使用 的簽署者範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Signer 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

cancel-signing-profile

以下程式碼範例顯示如何使用 cancel-signing-profile。

AWS CLI

刪除簽署設定檔

下列cancel-signing-profile範例會從 AWS Signer 移除現有的簽署設定檔。

```
aws signer cancel-signing-profile \
    --profile-name MyProfile1
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelSigningProfile。

describe-signing-job

以下程式碼範例顯示如何使用 describe-signing-job。

AWS CLI

顯示簽署任務的詳細資訊

下列describe-signing-job範例顯示指定簽署任務的詳細資訊。

```
aws signer describe-signing-job \
    --job-id 2065c468-73e2-4385-a6c9-0123456789abc
```

輸出：

```
{
    "status": "Succeeded",
    "completedAt": 1568412037,
    "platformId": "AmazonFreeRTOS-Default",
    "signingMaterial": {
        "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
    },
    "statusReason": "Signing Succeeded",
    "jobId": "2065c468-73e2-4385-a6c9-0123456789abc",
    "source": {
        "s3": {
            "version": "PNyFaUTgsQh5ZdMCcoCe6pT1gOpgB_M4",
            "bucketName": "signer-source",
            "key": "MyCode.rb"
        }
    },
    "profileName": "MyProfile2",
```

```
    "signedObject": {
        "s3": {
            "bucketName": "signer-destination",
            "key": "signed-2065c468-73e2-4385-a6c9-0123456789abc"
        }
    },
    "requestedBy": "arn:aws:iam::123456789012:user/maria",
    "createdAt": 1568412036
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSigningJob。

get-signing-platform

以下程式碼範例顯示如何使用 get-signing-platform。

AWS CLI

顯示簽署平台的詳細資訊

下列get-signing-platform範例顯示指定簽署平台的詳細資訊。

```
aws signer get-signing-platform \
    --platform-id AmazonFreeRTOS-TI-CC3220SF
```

輸出：

```
{
    "category": "AWS",
    "displayName": "Amazon FreeRTOS SHA1-RSA CC3220SF-Format",
    "target": "SHA1-RSA-TISHA1",
    "platformId": "AmazonFreeRTOS-TI-CC3220SF",
    "signingConfiguration": {
        "encryptionAlgorithmOptions": {
            "defaultValue": "RSA",
            "allowedValues": [
                "RSA"
            ]
        },
        "hashAlgorithmOptions": {
            "defaultValue": "SHA1",
            "allowedValues": [
```

```
                "SHA1"
            ]
        }
    },
    "maxSizeInMB": 16,
    "partner": "AmazonFreeRTOS",
    "signingImageFormat": {
        "defaultFormat": "JSONEmbedded",
        "supportedFormats": [
            "JSONEmbedded"
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSigningPlatform。

get-signing-profile

以下程式碼範例顯示如何使用 get-signing-profile。

AWS CLI

顯示簽署設定檔的詳細資訊

下列get-signing-profile範例顯示指定簽署設定檔的詳細資訊。

```
aws signer get-signing-profile \
    --profile-name MyProfile3
```

輸出：

```
{
    "platformId": "AmazonFreeRTOS-TI-CC3220SF",
    "profileName": "MyProfile3",
    "status": "Active",
    "signingMaterial": {
        "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSigningProfile。

list-signing-jobs

以下程式碼範例顯示如何使用 list-signing-jobs。

AWS CLI

列出所有簽署任務

下列list-signing-jobs範例顯示帳戶所有簽署任務的詳細資訊。

```
aws signer list-signing-jobs
```

在此範例中，會傳回兩個任務，一個成功，另一個失敗。

```
{
    "jobs": [
        {
            "status": "Succeeded",
            "signingMaterial": {
                "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
            },
            "jobId": "2065c468-73e2-4385-a6c9-0123456789abc",
            "source": {
                "s3": {
                    "version": "PNyFaUTgsQh5ZdMCcoCe6pT1gOpgB_M4",
                    "bucketName": "signer-source",
                    "key": "MyCode.rb"
                }
            },
            "signedObject": {
                "s3": {
                    "bucketName": "signer-destination",
                    "key": "signed-2065c468-73e2-4385-a6c9-0123456789abc"
                }
            },
            "createdAt": 1568412036
        },
        {
            "status": "Failed",
            "source": {
                "s3": {
                    "version": "PNyFaUTgsQh5ZdMCcoCe6pT1gOpgB_M4",
                    "bucketName": "signer-source",
```

```
                    "key": "MyOtherCode.rb"
                }
            },
            "signingMaterial": {
                "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
            },
            "createdAt": 1568402690,
            "jobId": "74d9825e-22fc-4a0d-b962-0123456789abc"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSigningJobs。

list-signing-platforms

以下程式碼範例顯示如何使用 list-signing-platforms。

AWS CLI

列出所有簽署平台

下列list-signing-platforms範例顯示所有可用簽署平台的詳細資訊。

```
aws signer list-signing-platforms
```

輸出：

```
{
    "platforms": [
        {
            "category": "AWS",
            "displayName": "AWS IoT Device Management SHA256-ECDSA ",
            "target": "SHA256-ECDSA",
            "platformId": "AWSIoTDeviceManagement-SHA256-ECDSA",
            "signingConfiguration": {
                "encryptionAlgorithmOptions": {
                    "defaultValue": "ECDSA",
                    "allowedValues": [
                        "ECDSA"
                    ]
                },
```

```
                "hashAlgorithmOptions": {
                    "defaultValue": "SHA256",
                    "allowedValues": [
                        "SHA256"
                    ]
                }
            },
            "maxSizeInMB": 2048,
            "partner": "AWSIoTDeviceManagement",
            "signingImageFormat": {
                "defaultFormat": "JSONDetached",
                "supportedFormats": [
                    "JSONDetached"
                ]
            }
        },
        {
            "category": "AWS",
            "displayName": "Amazon FreeRTOS SHA1-RSA CC3220SF-Format",
            "target": "SHA1-RSA-TISHA1",
            "platformId": "AmazonFreeRTOS-TI-CC3220SF",
            "signingConfiguration": {
                "encryptionAlgorithmOptions": {
                    "defaultValue": "RSA",
                    "allowedValues": [
                        "RSA"
                    ]
                },
                "hashAlgorithmOptions": {
                    "defaultValue": "SHA1",
                    "allowedValues": [
                        "SHA1"
                    ]
                }
            },
            "maxSizeInMB": 16,
            "partner": "AmazonFreeRTOS",
            "signingImageFormat": {
                "defaultFormat": "JSONEmbedded",
                "supportedFormats": [
                    "JSONEmbedded"
                ]
            }
        },
```

```
        {
            "category": "AWS",
            "displayName": "Amazon FreeRTOS SHA256-ECDSA",
            "target": "SHA256-ECDSA",
            "platformId": "AmazonFreeRTOS-Default",
            "signingConfiguration": {
                "encryptionAlgorithmOptions": {
                    "defaultValue": "ECDSA",
                    "allowedValues": [
                        "ECDSA"
                    ]
                },
                "hashAlgorithmOptions": {
                    "defaultValue": "SHA256",
                    "allowedValues": [
                        "SHA256"
                    ]
                }
            },
            "maxSizeInMB": 16,
            "partner": "AmazonFreeRTOS",
            "signingImageFormat": {
                "defaultFormat": "JSONEmbedded",
                "supportedFormats": [
                    "JSONEmbedded"
                ]
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSigningPlatforms。

list-signing-profiles

以下程式碼範例顯示如何使用 list-signing-profiles。

AWS CLI

列出所有簽署設定檔

下列list-signing-profiles範例顯示帳戶所有簽署設定檔的詳細資訊。

```
aws signer list-signing-profiles
```

輸出：

```
{
    "profiles": [
        {
            "platformId": "AmazonFreeRTOS-TI-CC3220SF",
            "profileName": "MyProfile4",
            "status": "Active",
            "signingMaterial": {
                "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
            }
        },
        {
            "platformId": "AWSIoTDeviceManagement-SHA256-ECDSA",
            "profileName": "MyProfile5",
            "status": "Active",
            "signingMaterial": {
                "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSigningProfiles。

put-signing-profile

以下程式碼範例顯示如何使用 put-signing-profile。

AWS CLI

建立簽署設定檔

下列put-signing-profile範例會使用指定的憑證和平台建立簽署設定檔。

```
aws signer put-signing-profile \
    --profile-name MyProfile6 \
```

```
    --signing-material certificateArn=arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc \
    --platform AmazonFreeRTOS-TI-CC3220SF
```

輸出：

```
{
    "arn": "arn:aws:signer:us-west-2:123456789012:/signing-profiles/MyProfile6"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutSigningProfile。

start-signing-job

以下程式碼範例顯示如何使用 start-signing-job。

AWS CLI

啟動簽署任務

下列start-signing-job範例會在指定來源找到的程式碼上啟動簽署任務。它使用指定的設定檔來執行簽署，並將簽署的程式碼放在指定的目的地。

```
aws signer start-signing-job \
    --source 's3={bucketName=signer-
source,key=MyCode.rb,version=PNyFaUTgsQh5ZdMCcoCe6pT1gOpgB_M4}' \
    --destination 's3={bucketName=signer-destination,prefix=signed-}' \
    --profile-name MyProfile7
```

輸出是簽署任務的 ID。

```
{
    "jobId": "2065c468-73e2-4385-a6c9-0123456789abc"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartSigningJob。

使用 的 Snowball Edge 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Snowball Edge 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`get-snowball-usage`

以下程式碼範例顯示如何使用 `get-snowball-usage`。

AWS CLI

取得您帳戶 Snowball 服務限制的相關資訊

下列`get-snowball-usage`範例顯示您帳戶 Snowball 服務限制的相關資訊，以及您的帳戶使用中的 Snowball 數量。

```
aws snowball get-snowball-usage
```

輸出：

```
{
    "SnowballLimit": 1,
    "SnowballsInUse": 0
}
```

如需詳細資訊，請參閱《AWS Snowball 開發人員指南》中的 Snowball Edge 限制。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSnowballUsage。

`list-jobs`

以下程式碼範例顯示如何使用 `list-jobs`。

AWS CLI

列出您帳戶中目前的 Snowball 任務

下列`list-jobs`範例顯示 `JobListEntry` 物件陣列。在此範例中，會列出單一任務。

```
aws snowball list-jobs
```

輸出：

```
{
    "JobListEntries": [
        {
            "CreationDate": 2016-09-27T14:50Z,
             "Description": "Important Photos 2016-08-11",
             "IsMaster": TRUE,
             "JobId": "ABCd1e324fe-022f-488e-a98b-3b0566063db1",
             "JobState": "Complete",
             "JobType": "IMPORT",
             "SnowballType": "EDGE"
        }
   ]
}
```

如需詳細資訊，請參閱《AWS Snowball 開發人員指南》中的 Snowball Edge 裝置任務。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListJobs。

使用 的 Amazon SNS 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon SNS 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

案例是向您展示如何呼叫服務中的多個函數或與其他 AWS 服務組合來完成特定任務的程式碼範例。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作
- 案例

動作

`add-permission`

以下程式碼範例顯示如何使用 add-permission。

AWS CLI

將許可新增至主題

下列add-permission範例會新增 AWS 帳戶 的許可987654321098，以將 Publish動作與 AWS 帳戶 下指定的主題搭配使用123456789012。

```
aws sns add-permission \
    --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --label Publish-Permission \
    --aws-account-id 987654321098 \
    --action-name Publish
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddPermission。

`check-if-phone-number-is-opted-out`

以下程式碼範例顯示如何使用 check-if-phone-number-is-opted-out。

AWS CLI

檢查電話號碼是否停止接收簡訊

下列check-if-phone-number-is-opted-out範例會檢查指定的電話號碼是否選擇退出接收來自目前 AWS 帳戶的簡訊。

```
aws sns check-if-phone-number-is-opted-out \
    --phone-number +1555550100
```

輸出：

```
{
    "isOptedOut": false
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CheckIfPhoneNumberIsOptedOut。

confirm-subscription

以下程式碼範例顯示如何使用 confirm-subscription。

AWS CLI

確認訂閱

下列 confirm-subscription 命令會完成您訂閱名為 my-topic 的 SNS 主題時啟動的確認程序。-token 參數來自傳送到訂閱呼叫中所指定通知端點的確認訊息。

```
aws sns confirm-subscription \
    --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic \
    --
token 2336412f37fb687f5d51e6e241d7700ae02f7124d8268910b858cb4db727ceeb2474bb937929d3bdd7ce5d
```

輸出：

```
{
    "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-
topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ConfirmSubscription。

create-platform-application

以下程式碼範例顯示如何使用 create-platform-application。

AWS CLI

建立平台應用程式

下列create-platform-application範例會使用指定的平台登入資料建立 Google Firebase 平台應用程式。

```
aws sns create-platform-application \
    --name MyApplication \
    --platform GCM \
    --attributes PlatformCredential=EXAMPLEabcd12345jklm67890stuv12345bcdef
```

輸出：

```
{
    "PlatformApplicationArn": "arn:aws:sns:us-west-2:123456789012:app/GCM/
MyApplication"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePlatformApplication。

create-topic

以下程式碼範例顯示如何使用 create-topic。

AWS CLI

建立 SNS 主題

下列 create-topic 範例會建立名為 my-topic 的 SNS 主題。

```
aws sns create-topic \
    --name my-topic
```

輸出：

```
{
    "ResponseMetadata": {
        "RequestId": "1469e8d7-1642-564e-b85d-a19b4b341f83"
    },
```

```
    "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
}
```

如需詳細資訊，請參閱《 AWS 命令列界面使用者指南》中的搭配 Amazon SQS 和 Amazon SNS 使用命令列界面。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateTopic。

delete-endpoint

以下程式碼範例顯示如何使用 delete-endpoint。

AWS CLI

刪除平台應用程式端點

下列delete-endpoint範例會刪除指定的平台應用程式端點。

```
aws sns delete-endpoint \
    --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/
MyApplication/12345678-abcd-9012-efgh-345678901234
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteEndpoint。

delete-platform-application

以下程式碼範例顯示如何使用 delete-platform-application。

AWS CLI

刪除平台應用程式

下列delete-platform-application範例會刪除指定的平台應用程式。

```
aws sns delete-platform-application \
    --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/ADM/
MyApplication
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePlatformApplication。

delete-topic

以下程式碼範例顯示如何使用 delete-topic。

AWS CLI

刪除 SNS 主題

下列 delete-topic 範例會刪除指定的 SNS 主題。

```
aws sns delete-topic \
    --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteTopic。

get-endpoint-attributes

以下程式碼範例顯示如何使用 get-endpoint-attributes。

AWS CLI

列出平台應用程式端點屬性

下列get-endpoint-attributes範例列出指定平台應用程式端點的屬性。

```
aws sns get-endpoint-attributes \
    --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/
MyApplication/12345678-abcd-9012-efgh-345678901234
```

輸出：

```
{
    "Attributes": {
        "Enabled": "true",
        "Token": "EXAMPLE12345..."
    }
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetEndpointAttributes。

get-platform-application-attributes

以下程式碼範例顯示如何使用 get-platform-application-attributes。

AWS CLI

列出平台應用程式屬性

下列get-platform-application-attributes範例列出指定平台應用程式的屬性。

```
aws sns get-platform-application-attributes \
    --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/MPNS/
MyApplication
```

輸出：

```
{
    "Attributes": {
        "Enabled": "true",
        "SuccessFeedbackSampleRate": "100"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPlatformApplicationAttributes。

get-sms-attributes

以下程式碼範例顯示如何使用 get-sms-attributes。

AWS CLI

列出預設簡訊屬性

下列 get-sms-attributes 範例會列出傳送簡訊的預設屬性。

```
aws sns get-sms-attributes
```

輸出：

```
{
    "attributes": {
        "DefaultSenderID": "MyName"
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetSMSAttributes。

get-subscription-attributes

以下程式碼範例顯示如何使用 get-subscription-attributes。

AWS CLI

擷取主題的訂閱屬性

以下get-subscription-attributes顯示指定訂閱的屬性。您可以從 list-subscriptions命令的subscription-arn輸出取得 。

```
aws sns get-subscription-attributes \
    --subscription-arn "arn:aws:sns:us-west-2:123456789012:my-
topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
```

輸出：

```
{
    "Attributes": {
        "Endpoint": "my-email@example.com",
        "Protocol": "email",
        "RawMessageDelivery": "false",
        "ConfirmationWasAuthenticated": "false",
        "Owner": "123456789012",
        "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-
topic:8a21d249-4329-4871-acc6-7be709c6ea7f",
        "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSubscriptionAttributes。

get-topic-attributes

以下程式碼範例顯示如何使用 get-topic-attributes。

AWS CLI

擷取主題的屬性

下列 get-topic-attributes 範例會顯示指定主題的屬性。

```
aws sns get-topic-attributes \
    --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

輸出：

```
{
    "Attributes": {
        "SubscriptionsConfirmed": "1",
        "DisplayName": "my-topic",
        "SubscriptionsDeleted": "0",
        "EffectiveDeliveryPolicy": "{\"http\":{\"defaultHealthyRetryPolicy\":
{\"minDelayTarget\":20,\"maxDelayTarget\":20,\"numRetries\":3,\"numMaxDelayRetries
\":0,\"numNoDelayRetries\":0,\"numMinDelayRetries\":0,\"backoffFunction\":\"linear
\"},\"disableSubscriptionOverrides\":false}}",
        "Owner": "123456789012",
        "Policy": "{\"Version\":\"2008-10-17\",\"Id\":\"__default_policy_ID\",
\"Statement\":[{\"Sid\":\"__default_statement_ID\",\"Effect\":\"Allow\",\"Principal
\":{\"AWS\":\"*\"},\"Action\":[\"SNS:Subscribe\",\"SNS:ListSubscriptionsByTopic
\",\"SNS:DeleteTopic\",\"SNS:GetTopicAttributes\",\"SNS:Publish\",
\"SNS:RemovePermission\",\"SNS:AddPermission\",\"SNS:SetTopicAttributes\"],
\"Resource\":\"arn:aws:sns:us-west-2:123456789012:my-topic\",\"Condition\":
{\"StringEquals\":{\"AWS:SourceOwner\":\"0123456789012\"}}}]}",
        "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic",
        "SubscriptionsPending": "0"
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetTopicAttributes。

list-endpoints-by-platform-application

以下程式碼範例顯示如何使用 list-endpoints-by-platform-application。

AWS CLI

列出平台應用程式的端點

下列list-endpoints-by-platform-application範例列出指定平台應用程式的端點和端點屬性。

```
aws sns list-endpoints-by-platform-application \
    --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/
MyApplication
```

輸出：

```
{
    "Endpoints": [
        {
            "Attributes": {
                "Token": "EXAMPLE12345...,
                "Enabled": "true"
            },
            "EndpointArn": "arn:aws:sns:us-west-2:123456789012:endpoint/GCM/
MyApplication/12345678-abcd-9012-efgh-345678901234"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListEndpointsByPlatformApplication。

list-phone-numbers-opted-out

以下程式碼範例顯示如何使用 list-phone-numbers-opted-out。

AWS CLI

列出停止接收簡訊

下列 list-phone-numbers-opted-out 範例會列出選擇停止接收簡訊的電話號碼。

```
aws sns list-phone-numbers-opted-out
```

輸出：

```
{
    "phoneNumbers": [
        "+15555550100"
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListPhoneNumbersOptedOut。

list-platform-applications

以下程式碼範例顯示如何使用 list-platform-applications。

AWS CLI

列出平台應用程式

下列list-platform-applications範例列出 ADM 和 MPNS 的平台應用程式。

```
aws sns list-platform-applications
```

輸出：

```
{
    "PlatformApplications": [
        {
            "PlatformApplicationArn": "arn:aws:sns:us-west-2:123456789012:app/ADM/
MyApplication",
            "Attributes": {
                "SuccessFeedbackSampleRate": "100",
                "Enabled": "true"
            }
        },
        {
            "PlatformApplicationArn": "arn:aws:sns:us-west-2:123456789012:app/MPNS/
MyOtherApplication",
            "Attributes": {
                "SuccessFeedbackSampleRate": "100",
                "Enabled": "true"
            }
        }
    ]
```

```
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPlatformApplications。

`list-subscriptions-by-topic`

以下程式碼範例顯示如何使用 `list-subscriptions-by-topic`。

AWS CLI

列出與主題相關聯的訂閱

以下內容會`list-subscriptions-by-topic`擷取與指定主題相關聯的 SNS 訂閱清單。

```
aws sns list-subscriptions-by-topic \
    --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

輸出：

```
{
    "Subscriptions": [
        {
            "Owner": "123456789012",
            "Endpoint": "my-email@example.com",
            "Protocol": "email",
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic",
            "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-
topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListSubscriptionsByTopic。

`list-subscriptions`

以下程式碼範例顯示如何使用 `list-subscriptions`。

AWS CLI

列出您的 SNS 訂閱

下列list-subscriptions範例顯示您 AWS 帳戶中 SNS 訂閱的清單。

```
aws sns list-subscriptions
```

輸出：

```
{
    "Subscriptions": [
        {
            "Owner": "123456789012",
            "Endpoint": "my-email@example.com",
            "Protocol": "email",
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic",
            "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-
topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListSubscriptions。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出主題的標籤

下列list-tags-for-resource範例列出指定 Amazon SNS 主題的標籤。

```
aws sns list-tags-for-resource \
    --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic
```

輸出：

```
{
    "Tags": [
        {
            "Key": "Team",
```

```
            "Value": "Alpha"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`list-topics`

以下程式碼範例顯示如何使用 `list-topics`。

AWS CLI

列出您的 SNS 主題

下列`list-topics`範例列出您 AWS 帳戶中的所有 SNS 主題。

```
aws sns list-topics
```

輸出：

```
{
    "Topics": [
        {
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTopics。

`opt-in-phone-number`

以下程式碼範例顯示如何使用 `opt-in-phone-number`。

AWS CLI

選擇加入簡訊

下列`opt-in-phone-number`範例選擇指定的電話號碼來接收簡訊。

```
aws sns opt-in-phone-number \
    --phone-number +15555550100
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 OptInPhoneNumber。

publish

以下程式碼範例顯示如何使用 publish。

AWS CLI

範例 1：將訊息發布至主題

下列 publish 範例會將指定的訊息發佈到指定的 SNS 主題。訊息來自文字檔案，可讓您包含換行符號。

```
aws sns publish \
    --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic" \
    --message file://message.txt
```

message.txt 的內容：

```
Hello World
Second Line
```

輸出：

```
{
    "MessageId": "123a45b6-7890-12c3-45d6-111122223333"
}
```

範例 2：將簡訊發布至電話號碼

下列 publish 範例會將訊息 Hello world! 發佈至電話號碼 +1-555-555-0100。

```
aws sns publish \
    --message "Hello world!" \
```

```
    --phone-number +1-555-555-0100
```

輸出：

```
{
    "MessageId": "123a45b6-7890-12c3-45d6-333322221111"
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 Publish。

put-data-protection-policy

以下程式碼範例顯示如何使用 put-data-protection-policy。

AWS CLI

設定資料保護政策

範例 1：拒絕發佈者使用 CreditCardNumber 發佈訊息

下列put-data-protection-policy範例會拒絕發佈者使用 CreditCardNumber 發佈訊息。

```
aws sns put-data-protection-policy \
    --resource-arn arn:aws:sns:us-east-1:123456789012:mytopic \
    --data-protection-policy "{\"Name\":\"data_protection_policy\",\"Description
\":\"Example data protection policy\",\"Version\":\"2021-06-01\",\"Statement
\":[{\"DataDirection\":\"Inbound\",\"Principal\":[\"*\"],\"DataIdentifier\":
[\"arn:aws:dataprotection::aws:data-identifier/CreditCardNumber\"],\"Operation\":
{\"Deny\":{}}}]}"
```

此命令不會產生輸出。

範例 2：從檔案載入參數

下列會從檔案put-data-protection-policy載入參數。

```
aws sns put-data-protection-policy \
    --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --data-protection-policy file://policy.json
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutDataProtectionPolicy。

remove-permission

以下程式碼範例顯示如何使用 remove-permission。

AWS CLI

從主題移除許可

下列remove-permission範例Publish-Permission會從指定的主題移除 許可。

```
aws sns remove-permission \
    --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --label Publish-Permission
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 RemovePermission。

set-endpoint-attributes

以下程式碼範例顯示如何使用 set-endpoint-attributes。

AWS CLI

設定端點屬性

下列set-endpoint-attributes範例會停用指定的平台應用程式端點。

```
aws sns set-endpoint-attributes \
    --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/
MyApplication/12345678-abcd-9012-efgh-345678901234 \
    --attributes Enabled=false
```

輸出：

```
{
    "Attributes": {
        "Enabled": "false",
        "Token": "EXAMPLE12345..."
```

```
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetEndpointAttributes。

set-platform-application-attributes

以下程式碼範例顯示如何使用 set-platform-application-attributes。

AWS CLI

設定平台應用程式屬性

下列set-platform-application-attributes範例會將指定平台應用程式的 EventDeliveryFailure 屬性設定為指定 Amazon SNS 主題的 ARN。

```
aws sns set-platform-application-attributes \
    --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/
MyApplication \
    --attributes EventDeliveryFailure=arn:aws:sns:us-
west-2:123456789012:AnotherTopic
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetPlatformApplicationAttributes。

set-sms-attributes

以下程式碼範例顯示如何使用 set-sms-attributes。

AWS CLI

若要設定簡訊屬性

下列 set-sms-attributes 範例會將簡訊的預設寄件者 ID 設定為 MyName。

```
aws sns set-sms-attributes \
    --attributes DefaultSenderID=MyName
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 SetSMSAttributes。

set-subscription-attributes

以下程式碼範例顯示如何使用 set-subscription-attributes。

AWS CLI

設定訂閱屬性

下列 set-subscription-attributes 範例會將 RawMessageDelivery 屬性設定為 SQS 訂閱。

```
aws sns set-subscription-attributes \
    --subscription-arn arn:aws:sns:us-
east-1:123456789012:mytopic:f248de18-2cf6-578c-8592-b6f1eaa877dc \
    --attribute-name RawMessageDelivery \
    --attribute-value true
```

此命令不會產生輸出。

下列 set-subscription-attributes 範例會將 FilterPolicy 屬性設定為 SQS 訂閱。

```
aws sns set-subscription-attributes \
    --subscription-arn arn:aws:sns:us-
east-1:123456789012:mytopic:f248de18-2cf6-578c-8592-b6f1eaa877dc \
    --attribute-name FilterPolicy \
    --attribute-value "{ \"anyMandatoryKey\": [\"any\", \"of\", \"these\"] }"
```

此命令不會產生輸出。

下列 set-subscription-attributes 範例會從 SQS 訂閱移除 FilterPolicy 屬性。

```
aws sns set-subscription-attributes \
    --subscription-arn arn:aws:sns:us-
east-1:123456789012:mytopic:f248de18-2cf6-578c-8592-b6f1eaa877dc \
    --attribute-name FilterPolicy \
    --attribute-value "{}"
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 SetSubscriptionAttributes。

set-topic-attributes

以下程式碼範例顯示如何使用 set-topic-attributes。

AWS CLI

設定主題的屬性

下列 set-topic-attributes 範例會設定指定主題的 DisplayName 屬性。

```
aws sns set-topic-attributes \
    --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --attribute-name DisplayName \
    --attribute-value MyTopicDisplayName
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 SetTopicAttributes。

subscribe

以下程式碼範例顯示如何使用 subscribe。

AWS CLI

訂閱主題

下列 subscribe 命令會將電子郵件地址訂閱至指定的主題。

```
aws sns subscribe \
    --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic \
    --protocol email \
    --notification-endpoint my-email@example.com
```

輸出：

```
{
    "SubscriptionArn": "pending confirmation"
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 Subscribe。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤新增至主題

下列 tag-resource 範例會將中繼資料標籤新增到指定的 Amazon SNS 主題。

```
aws sns tag-resource \
    --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --tags Key=Team,Value=Alpha
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

unsubscribe

以下程式碼範例顯示如何使用 unsubscribe。

AWS CLI

取消訂閱主題

下列 unsubscribe 範例會從主題中刪除指定的訂閱。

```
aws sns unsubscribe \
    --subscription-arn arn:aws:sns:us-west-2:0123456789012:my-
topic:8a21d249-4329-4871-acc6-7be709c6ea7f
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 Unsubscribe。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從主題移除標籤

下列untag-resource範例會從指定的 Amazon SNS 主題中移除具有指定金鑰的任何標籤。

```
aws sns untag-resource \
    --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --tag-keys Team
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

案例

為推播通知建立平台端點

下列程式碼範例示範如何為 Amazon SNS 推播通知建立平台端點。

AWS CLI

建立平台應用程式端點

下列 create-platform-endpoint 範例會使用指定的字符，為指定的平台應用程式建立端點。

```
aws sns create-platform-endpoint \
    --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/
MyApplication \
    --token EXAMPLE12345...
```

輸出：

```
{
      "EndpointArn": "arn:aws:sns:us-west-2:1234567890:endpoint/GCM/
MyApplication/12345678-abcd-9012-efgh-345678901234"
}
```

使用 的 Amazon SQS 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon SQS 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`add-permission`

以下程式碼範例顯示如何使用 `add-permission`。

AWS CLI

將許可新增至佇列

此範例可讓指定的 AWS 帳戶傳送訊息至指定的佇列。

命令：

```
aws sqs add-permission --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --label SendMessagesFromMyQueue --aws-account-ids 12345EXAMPLE --actions SendMessage
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddPermission。

`cancel-message-move-task`

以下程式碼範例顯示如何使用 `cancel-message-move-task`。

AWS CLI

取消訊息移動任務

下列`cancel-message-move-task`範例會取消指定的訊息移動任務。

```
aws sqs cancel-message-move-task \
```

```
    --task-handle AQEB6nR4...HzlvZQ==
```

輸出：

```
{
    "ApproximateNumberOfMessagesMoved": 102
}
```

如需詳細資訊，請參閱《 開發人員指南》中的 Amazon SQS API 許可：動作和資源參考。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelMessageMoveTask。

change-message-visibility-batch

以下程式碼範例顯示如何使用 change-message-visibility-batch。

AWS CLI

以批次形式變更多則訊息的逾時可見性

此範例會將 2 個指定訊息的逾時可見性變更為 10 小時 (10 小時 * 60 分鐘 * 60 秒) 。

命令：

```
aws sqs change-message-visibility-batch --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://change-message-
visibility-batch.json
```

輸入檔案 (change-message-visibility-batch.json)：

```
[
  {
    "Id": "FirstMessage",
        "ReceiptHandle": "AQEBhz2q...Jf3kaw==",
        "VisibilityTimeout": 36000
  },
  {
    "Id": "SecondMessage",
        "ReceiptHandle": "AQEBkTUH...HifSnw==",
        "VisibilityTimeout": 36000
  }
]
```

輸出：

```
{
  "Successful": [
    {
      "Id": "SecondMessage"
    },
    {
      "Id": "FirstMessage"
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ChangeMessageVisibilityBatch。

change-message-visibility

以下程式碼範例顯示如何使用 change-message-visibility。

AWS CLI

變更訊息的逾時可見性

此範例會將指定訊息的逾時可見性變更為 10 小時 (10 小時 * 60 分鐘 * 60 秒) 。

命令：

```
aws sqs change-message-visibility --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyQueue --receipt-handle AQEBTpyI...t6HyQg== --
visibility-timeout 36000
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ChangeMessageVisibility。

create-queue

以下程式碼範例顯示如何使用 create-queue。

AWS CLI

建立佇列

此範例會建立具有指定名稱的佇列、將訊息保留期間設定為 3 天 (3 天 * 24 小時 * 60 分鐘 * 60 秒），並將佇列的無效字母佇列設定為接收計數上限為 1，000 則訊息的指定佇列。

命令：

```
aws sqs create-queue --queue-name MyQueue --attributes file://create-queue.json
```

輸入檔案 (create-queue.json)：

```
{
  "RedrivePolicy": "{\"deadLetterTargetArn\":\"arn:aws:sqs:us-
east-1:80398EXAMPLE:MyDeadLetterQueue\",\"maxReceiveCount\":\"1000\"}",
  "MessageRetentionPeriod": "259200"
}
```

輸出：

```
{
  "QueueUrl": "https://queue.amazonaws.com/80398EXAMPLE/MyQueue"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateQueue。

delete-message-batch

以下程式碼範例顯示如何使用 delete-message-batch。

AWS CLI

刪除批次形式的多則訊息

此範例會刪除指定的訊息。

命令：

```
aws sqs delete-message-batch --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://delete-message-batch.json
```

輸入檔案 (delete-message-batch.json)：

```
[
  {
        "Id": "FirstMessage",
        "ReceiptHandle": "AQEB1mgl...Z4GuLw=="
  },
  {
    "Id": "SecondMessage",
        "ReceiptHandle": "AQEBLsYM...VQubAA=="
  }
]
```

輸出：

```
{
  "Successful": [
    {
      "Id": "FirstMessage"
    },
    {
      "Id": "SecondMessage"
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteMessageBatch。

delete-message

以下程式碼範例顯示如何使用 delete-message。

AWS CLI

刪除訊息

此範例會刪除指定的訊息。

命令：

```
aws sqs delete-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --receipt-handle AQEBRXTo...q2doVA==
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteMessage。

delete-queue

以下程式碼範例顯示如何使用 delete-queue。

AWS CLI

刪除佇列

此範例會刪除指定的佇列。

命令：

```
aws sqs delete-queue --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewerQueue
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteQueue。

get-queue-attributes

以下程式碼範例顯示如何使用 get-queue-attributes。

AWS CLI

取得佇列的屬性

此範例會取得所有指定的佇列屬性。

命令：

```
aws sqs get-queue-attributes --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names All
```

輸出：

```
{
  "Attributes": {
    "ApproximateNumberOfMessagesNotVisible": "0",
    "RedrivePolicy": "{\"deadLetterTargetArn\":\"arn:aws:sqs:us-
east-1:80398EXAMPLE:MyDeadLetterQueue\",\"maxReceiveCount\":1000}",
    "MessageRetentionPeriod": "345600",
    "ApproximateNumberOfMessagesDelayed": "0",
    "MaximumMessageSize": "262144",
    "CreatedTimestamp": "1442426968",
    "ApproximateNumberOfMessages": "0",
    "ReceiveMessageWaitTimeSeconds": "0",
    "DelaySeconds": "0",
    "VisibilityTimeout": "30",
    "LastModifiedTimestamp": "1442426968",
    "QueueArn": "arn:aws:sqs:us-east-1:80398EXAMPLE:MyNewQueue"
  }
}
```

此範例只會取得指定佇列的訊息大小上限和可見性逾時屬性。

命令：

```
aws sqs get-queue-attributes --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyNewQueue --attribute-
names MaximumMessageSize VisibilityTimeout
```

輸出：

```
{
  "Attributes": {
    "VisibilityTimeout": "30",
    "MaximumMessageSize": "262144"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetQueueAttributes。

get-queue-url

以下程式碼範例顯示如何使用 get-queue-url。

AWS CLI

取得佇列 URL

此範例會取得指定佇列的 URL。

命令：

```
aws sqs get-queue-url --queue-name MyQueue
```

輸出：

```
{
  "QueueUrl": "https://queue.amazonaws.com/80398EXAMPLE/MyQueue"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetQueueUrl。

`list-dead-letter-source-queues`

以下程式碼範例顯示如何使用 `list-dead-letter-source-queues`。

AWS CLI

列出無效字母來源佇列

此範例列出與指定的無效字母來源佇列相關聯的佇列。

命令：

```
aws sqs list-dead-letter-source-queues --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue
```

輸出：

```
{
  "queueUrls": [
    "https://queue.amazonaws.com/80398EXAMPLE/MyQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue"
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDeadLetterSourceQueues。

list-message-move-tasks

以下程式碼範例顯示如何使用 list-message-move-tasks。

AWS CLI

列出訊息移動任務

下列list-message-move-tasks範例列出指定佇列中 2 個最新的訊息移動任務。

```
aws sqs list-message-move-tasks \
    --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue \
    --max-results 2
```

輸出：

```
{
    "Results": [
        {
            "TaskHandle": "AQEB6nR4...HzlvZQ==",
            "Status": "RUNNING",
            "SourceArn": "arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue1",
            "DestinationArn": "arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue2",
            "MaxNumberOfMessagesPerSecond": 50,
            "ApproximateNumberOfMessagesMoved": 203,
            "ApproximateNumberOfMessagesToMove": 30,
            "StartedTimestamp": 1442428276921
        },

        {
            "Status": "COMPLETED",
            "SourceArn": "arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue1",
            "DestinationArn": "arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue2",
            "ApproximateNumberOfMessagesMoved": 29,
            "ApproximateNumberOfMessagesToMove": 0,
            "StartedTimestamp": 1342428272093
        }
    ]
}
```

如需詳細資訊，請參閱《 開發人員指南》中的 Amazon SQS API 許可：動作和資源參考。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMessageMoveTasks。

list-queue-tags

以下程式碼範例顯示如何使用 list-queue-tags。

AWS CLI

列出佇列的所有成本分配標籤

下列list-queue-tags範例顯示與指定佇列相關聯的所有成本分配標籤。

```
aws sqs list-queue-tags \
    --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue
```

輸出：

```
{
    "Tags": {
        "Team": "Alpha"
    }
}
```

如需詳細資訊，請參閱《Amazon Simple Queue Service 開發人員指南》中的列出成本分配標籤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListQueueTags。

list-queues

以下程式碼範例顯示如何使用 list-queues。

AWS CLI

列出佇列

此範例列出所有佇列。

命令：

```
aws sqs list-queues
```

輸出：

```
{
  "QueueUrls": [
    "https://queue.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/TestQueue1",
        "https://queue.amazonaws.com/80398EXAMPLE/TestQueue2"
  ]
}
```

此範例只會列出以 "My" 開頭的佇列。

命令：

```
aws sqs list-queues --queue-name-prefix My
```

輸出：

```
{
  "QueueUrls": [
    "https://queue.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue"
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListQueues。

purge-queue

以下程式碼範例顯示如何使用 purge-queue。

AWS CLI

清除佇列

此範例會刪除指定佇列中的所有訊息。

命令：

```
aws sqs purge-queue --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewQueue
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PurgeQueue。

receive-message

以下程式碼範例顯示如何使用 receive-message。

AWS CLI

接收訊息

此範例最多可接收 10 則可用訊息，並傳回所有可用的屬性。

命令：

```
aws sqs receive-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names All --message-attribute-names All --max-number-of-messages 10
```

輸出：

```
{
  "Messages": [
    {
      "Body": "My first message.",
      "ReceiptHandle": "AQEBzbVv...fqNzFw==",
      "MD5OfBody": "1000f835...a35411fa",
      "MD5OfMessageAttributes": "9424c491...26bc3ae7",
      "MessageId": "d6790f8d-d575-4f01-bc51-40122EXAMPLE",
      "Attributes": {
        "ApproximateFirstReceiveTimestamp": "1442428276921",
        "SenderId": "AIDAIAZKMSNQ7TEXAMPLE",
        "ApproximateReceiveCount": "5",
        "SentTimestamp": "1442428276921"
      },
```

```
      "MessageAttributes": {
        "PostalCode": {
          "DataType": "String",
          "StringValue": "ABC123"
        },
        "City": {
          "DataType": "String",
          "StringValue": "Any City"
        }
      }
    }
  ]
}
```

此範例會接收下一個可用的訊息，僅傳回 SenderId 和 SentTimestamp 屬性，以及 PostalCode 訊息屬性。

命令：

```
aws sqs receive-message --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names SenderId SentTimestamp
 --message-attribute-names PostalCode
```

輸出：

```
{
  "Messages": [
    {
      "Body": "My first message.",
      "ReceiptHandle": "AQEB6nR4...HzlvZQ==",
      "MD5OfBody": "1000f835...a35411fa",
      "MD5OfMessageAttributes": "b8e89563...e088e74f",
      "MessageId": "d6790f8d-d575-4f01-bc51-40122EXAMPLE",
      "Attributes": {
        "SenderId": "AIDAIAZKMSNQ7TEXAMPLE",
        "SentTimestamp": "1442428276921"
      },
      "MessageAttributes": {
        "PostalCode": {
          "DataType": "String",
          "StringValue": "ABC123"
        }
      }
```

```
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ReceiveMessage。

`remove-permission`

以下程式碼範例顯示如何使用 `remove-permission`。

AWS CLI

移除許可

此範例會從指定的佇列中移除具有指定標籤的許可。

命令：

```
aws sqs remove-permission --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyQueue --label SendMessagesFromMyQueue
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 RemovePermission。

`send-message-batch`

以下程式碼範例顯示如何使用 `send-message-batch`。

AWS CLI

以批次形式傳送多則訊息

此範例會將具有指定訊息內文、延遲期間和訊息屬性的 2 則訊息傳送至指定的佇列。

命令：

```
aws sqs send-message-batch --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://send-message-batch.json
```

輸入檔案 (send-message-batch.json)：

```
[
  {
    "Id": "FuelReport-0001-2015-09-16T140731Z",
        "MessageBody": "Fuel report for account 0001 on 2015-09-16 at 02:07:31 PM.",
        "DelaySeconds": 10,
        "MessageAttributes": {
          "SellerName": {
            "DataType": "String",
                "StringValue": "Example Store"
      },
          "City": {
        "DataType": "String",
        "StringValue": "Any City"
      },
          "Region": {
            "DataType": "String",
                "StringValue": "WA"
      },
          "PostalCode": {
            "DataType": "String",
                "StringValue": "99065"
          },
          "PricePerGallon": {
            "DataType": "Number",
                "StringValue": "1.99"
      }
        }
  },
  {
    "Id": "FuelReport-0002-2015-09-16T140930Z",
        "MessageBody": "Fuel report for account 0002 on 2015-09-16 at 02:09:30 PM.",
        "DelaySeconds": 10,
        "MessageAttributes": {
          "SellerName": {
            "DataType": "String",
                "StringValue": "Example Fuels"
      },
          "City": {
        "DataType": "String",
        "StringValue": "North Town"
      },
          "Region": {
```

```
            "DataType": "String",
                "StringValue": "WA"
        },
            "PostalCode": {
            "DataType": "String",
                "StringValue": "99123"
            },
            "PricePerGallon": {
            "DataType": "Number",
                "StringValue": "1.87"
        }
          }
    }
]
```

輸出：

```
{
  "Successful": [
    {
      "MD5OfMessageBody": "203c4a38...7943237e",
      "MD5OfMessageAttributes": "10809b55...baf283ef",
      "Id": "FuelReport-0001-2015-09-16T140731Z",
      "MessageId": "d175070c-d6b8-4101-861d-adeb3EXAMPLE"
    },
    {
      "MD5OfMessageBody": "2cf0159a...c1980595",
      "MD5OfMessageAttributes": "55623928...ae354a25",
      "Id": "FuelReport-0002-2015-09-16T140930Z",
      "MessageId": "f9b7d55d-0570-413e-b9c5-a9264EXAMPLE"
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SendMessageBatch。

send-message

以下程式碼範例顯示如何使用 send-message。

AWS CLI

傳送訊息

此範例會將具有指定訊息內文、延遲期間和訊息屬性的訊息傳送至指定的佇列。

命令：

```
aws sqs send-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --message-body "Information about the largest city in Any Region." --delay-seconds 10 --message-attributes file://send-message.json
```

輸入檔案 (send-message.json)：

```
{
  "City": {
    "DataType": "String",
    "StringValue": "Any City"
  },
  "Greeting": {
    "DataType": "Binary",
    "BinaryValue": "Hello, World!"
  },
  "Population": {
    "DataType": "Number",
    "StringValue": "1250800"
  }
}
```

輸出：

```
{
  "MD5OfMessageBody": "51b0a325...39163aa0",
  "MD5OfMessageAttributes": "00484c68...59e48f06",
  "MessageId": "da68f62c-0c07-4bee-bf5f-7e856EXAMPLE"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SendMessage。

set-queue-attributes

以下程式碼範例顯示如何使用 set-queue-attributes。

AWS CLI

設定佇列屬性

此範例會將指定的佇列設定為 10 秒的交付延遲、128 KB (128 KB * 1，024 位元組） 的訊息大小上限、3 天 (3 天 * 24 小時 * 60 分鐘 * 60 秒） 的訊息保留期間、20 秒的接收訊息等待時間，以及 60 秒的預設可見性逾時。此範例也會將指定的無效字母佇列與 1，000 則訊息的最大接收計數建立關聯。

命令：

```
aws sqs set-queue-attributes --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewQueue --attributes file://set-queue-attributes.json
```

輸入檔案 (set-queue-attributes.json)：

```
{
  "DelaySeconds": "10",
  "MaximumMessageSize": "131072",
  "MessageRetentionPeriod": "259200",
  "ReceiveMessageWaitTimeSeconds": "20",
  "RedrivePolicy": "{\"deadLetterTargetArn\":\"arn:aws:sqs:us-east-1:80398EXAMPLE:MyDeadLetterQueue\",\"maxReceiveCount\":\"1000\"}",
  "VisibilityTimeout": "60"
}
```

輸出：

```
None.
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SetQueueAttributes。

start-message-move-task

以下程式碼範例顯示如何使用 start-message-move-task。

AWS CLI

範例 1：*若要開始訊息移動任務*

下列start-message-move-task範例會啟動訊息移動任務，將訊息從指定的無效字母佇列重新驅動到來源佇列。

```
aws sqs start-message-move-task \
```

```
    --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue
```

輸出：

```
{
    "TaskHandle": "AQEB6nR4...HzlvZQ=="
}
```

如需詳細資訊，請參閱指南名稱中的這是主題標題。

範例 2：*以最高速率開始訊息移動任務*

下列start-message-move-task範例會啟動訊息移動任務，將訊息從指定的無效字母佇列重新驅動到指定的目的地佇列，速率上限為每秒 50 則訊息。

```
aws sqs start-message-move-task \
    --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue1 \
    --destination-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue2 \
    --max-number-of-messages-per-second 50
```

輸出：

```
{
    "TaskHandle": "AQEB6nR4...HzlvZQ=="
}
```

如需詳細資訊，請參閱《 開發人員指南》中的 Amazon SQS API 許可：動作和資源參考。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartMessageMoveTask。

tag-queue

以下程式碼範例顯示如何使用 tag-queue。

AWS CLI

將成本分配標籤新增至佇列

下列tag-queue範例會將成本分配標籤新增至指定的 Amazon SQS 佇列。

```
aws sqs tag-queue \
```

```
    --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue \
    --tags Priority=Highest
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Simple Queue Service 開發人員指南》中的新增成本分配標籤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TagQueue。

untag-queue

以下程式碼範例顯示如何使用 untag-queue。

AWS CLI

從佇列中移除成本分配標籤

下列untag-queue範例會從指定的 Amazon SQS 佇列中移除成本分配標籤。

```
aws sqs untag-queue \
    --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue \
    --tag-keys "Priority"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Simple Queue Service 開發人員指南》中的新增成本分配標籤。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UntagQueue。

使用 的 Storage Gateway 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Storage Gateway 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`describe-gateway-information`

以下程式碼範例顯示如何使用 `describe-gateway-information`。

AWS CLI

描述閘道

下列`describe-gateway-information`命令會傳回指定閘道的相關中繼資料。若要指定要描述的閘道，請在 命令中使用閘道的 Amazon Resource Name (ARN)。

此範例指定帳戶 `sgw-12A3456B`中 ID 為 的閘道`123456789012`：

```
aws storagegateway describe-gateway-information --gateway-
arn "arn:aws:storagegateway:us-west-2:123456789012:gateway/sgw-12A3456B"
```

此命令會輸出 JSON 區塊，其中包含有關閘道的中繼資料，例如其名稱、網路介面、設定的時區和狀態（閘道是否正在執行）。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeGatewayInformation。

`list-file-shares`

以下程式碼範例顯示如何使用 `list-file-shares`。

AWS CLI

列出檔案共享

下列`command-name`範例列出您 AWS 帳戶中可用的小工具。

```
aws storagegateway list-file-shares \
    --gateway-arn arn:aws:storagegateway:us-east-1:209870788375:gateway/sgw-FB02E292
```

輸出：

```
{
    "FileShareInfoList": [
        {
            "FileShareType": "NFS",
```

```
            "FileShareARN": "arn:aws:storagegateway:us-east-1:111122223333:share/
share-2FA12345",
            "FileShareId": "share-2FA12345",
            "FileShareStatus": "AVAILABLE",
            "GatewayARN": "arn:aws:storagegateway:us-east-1:111122223333:gateway/
sgw-FB0AAAAA"
        }
    ],
    "Marker": null
}
```

如需詳細資訊，請參閱 AWS Storage Gateway Service API 參考中的 ListFileShares。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListFileShares。

`list-gateways`

以下程式碼範例顯示如何使用 `list-gateways`。

AWS CLI

列出帳戶的閘道

下列`list-gateways`命令會列出為 帳戶定義的所有閘道：

```
aws storagegateway list-gateways
```

此命令會輸出 JSON 區塊，其中包含閘道 Amazon Resource Name (ARNs的清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListGateways。

`list-volumes`

以下程式碼範例顯示如何使用 `list-volumes`。

AWS CLI

列出為閘道設定的磁碟區

下列`list-volumes`命令會傳回為指定閘道設定的磁碟區清單。若要指定要描述的閘道，請在 命令中使用閘道的 Amazon Resource Name (ARN)。

此範例指定帳戶 `sgw-12A3456B`中 ID 為 的閘道`123456789012`：

```
aws storagegateway list-volumes --gateway-arn "arn:aws:storagegateway:us-
west-2:123456789012:gateway/sgw-12A3456B"
```

此命令會輸出 JSON 區塊，其中包含每個磁碟區的類型和 ARN 的磁碟區清單。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListVolumes。

`refresh-cache`

以下程式碼範例顯示如何使用 `refresh-cache`。

AWS CLI

重新整理檔案共享快取

下列`refresh-cache`範例會重新整理指定檔案共享的快取。

```
aws storagegateway refresh-cache \
    --file-share-arn arn:aws:storagegateway:us-east-1:111122223333:share/
share-2FA12345
```

輸出：

```
{
    "FileShareARN": "arn:aws:storagegateway:us-east-1:111122223333:share/
share-2FA12345",
    "NotificationId": "4954d4b1-abcd-ef01-1234-97950a7d3483"
}
```

如需詳細資訊，請參閱 AWS Storage Gateway Service API 參考中的 ListFileShares。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RefreshCache。

AWS STS 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS STS。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

assume-role-with-saml

以下程式碼範例顯示如何使用 assume-role-with-saml。

AWS CLI

取得使用 SAML 驗證之角色的短期登入資料

下列 assume-role-with-saml 命令會為 IAM 角色 TestSaml 擷取一組短期憑證。此範例中的請求是使用身分提供者在進行身分驗證時提供的 SAML 聲明進行身分驗證。

```
aws sts assume-role-with-saml \
    --role-arn arn:aws:iam::123456789012:role/TestSaml \
    --principal-arn arn:aws:iam::123456789012:saml-provider/SAML-test \
    --saml-
assertion "VERYLONGENCODEDASSERTIONEXAMPLExzYW1sOkF1ZGllbmNlPmJsYW5rPC9zYW1sOkF1ZGllbmNlPjwv
+PHNhbWw6TmFtZUlEIEZvcm1hdD0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOm5hbWVpZC1mb3JtYXQ6dHJhbnN
+PHNhbWw6U3ViamVjdENvbmZpcm1hdGlvbiBNZXRob2Q9InVybjpvYXNpczpuYW1lczp0YzpTQU1MOjIuMDpjbTpiZWF
```

輸出：

```
{
    "Issuer": "https://integ.example.com/idp/shibboleth</Issuer",
    "AssumedRoleUser": {
        "Arn": "arn:aws:sts::123456789012:assumed-role/TestSaml",
        "AssumedRoleId": "ARO456EXAMPLE789:TestSaml"
    },
    "Credentials": {
        "AccessKeyId": "ASIAV3ZUEFP6EXAMPLE",
        "SecretAccessKey": "8P+SQvWIuLnKhh8d++jpw0nNmQRBZvNEXAMPLEKEY",
        "SessionToken": "IQoJb3JpZ2luX2VjEOz////////////////////
wEXAMPLEtMSJHMEUCIDoKK3JH9uGQE1z0sINr5M4jk
+Na8KHDcCYRVjJCZEvOAiEA3OvJGtw1EcViOleS2vhs8VdCKFJQWPQrmGdeehM4IC1NtBmUpp2wUE8phUZampKsburED
+xo0rKwT38xVqr7ZD0u0iPPkUL64lIZbqBAz
+scqKmlzm8FDrypNC9Yjc8fPOLn9FX9KSYvKTr4rvx3iSIlTJabIQwj2ICCR/oLxBA==",
        "Expiration": "2019-11-01T20:26:47Z"
    },
```

```
    "Audience": "https://signin.aws.amazon.com/saml",
    "SubjectType": "transient",
    "PackedPolicySize": "6",
    "NameQualifier": "SbdGOnUkh1i4+EXAMPLExL/jEvs=",
    "Subject": "SamlExample"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的請求臨時安全憑證。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssumeRoleWithSaml。

assume-role-with-web-identity

以下程式碼範例顯示如何使用 assume-role-with-web-identity。

AWS CLI

取得使用 Web Identity (OAuth 2."0) 驗證之角色的短期憑證

下列 assume-role-with-web-identity 命令會為 IAM 角色 app1 擷取一組短期憑證。使用指定 Web 身分提供程式提供的 Web 身分權杖來驗證身分。兩個額外的政策會套用至工作階段，進一步限制使用者可以執行的動作。傳回的憑證會在產生後一小時過期。

```
aws sts assume-role-with-web-identity \
    --duration-seconds 3600 \
    --role-session-name "app1" \
    --provider-id "www.amazon.com" \
    --policy-arns "arn:aws:iam::123456789012:policy/
q=webidentitydemopolicy1","arn:aws:iam::123456789012:policy/webidentitydemopolicy2"
 \
    --role-arn arn:aws:iam::123456789012:role/FederatedWebIdentityRole \
    --web-identity-token "Atza
%7CIQEBLjAsAhRFiXuWpUXuRvQ9PZL3GMFcYevydwIUFAHZwXZXXXXXXXXJnrulxKDHwy87oGKPznh0D6bEQZTSCzyoC
CrKqjG7nPBjNIL016GGvuS5gSvPRUxWES3VYfm1wl7WTI7jn-Pcb6M-
buCgHhFOzTQxod27L9CqnOLio7N3gZAGpsp6n1-
AJBOCJckcyXe2c6uD0srOJeZlKUm2eTDVMf8IehDVI0r1QOnTV6KzzAI3OY87Vd_cVMQ"
```

輸出：

```
{
    "SubjectFromWebIdentityToken": "amzn1.account.AF6RHO7KZU5XRVQJGXK6HB56KR2A",
    "Audience": "client.5498841531868486423.1548@apps.example.com",
```

```
    "AssumedRoleUser": {
        "Arn": "arn:aws:sts::123456789012:assumed-role/FederatedWebIdentityRole/
app1",
        "AssumedRoleId": "AROACLKWSDQRAOEXAMPLE:app1"
    },
    "Credentials": {
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJOgQs8IZZaIv2BXIa2R4OlgkBN9bkUDNCJiBeb/
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wnOE49mRl/+OtkIKGO7fAE",
        "Expiration": "2020-05-19T18:06:10+00:00"
    },
    "Provider": "www.amazon.com"
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的請求臨時安全憑證。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 AssumeRoleWithWebIdentity。

assume-role

以下程式碼範例顯示如何使用 assume-role。

AWS CLI

擔任角色

下列 assume-role 命令會為 IAM 角色 s3-access-example 擷取一組短期憑證。

```
aws sts assume-role \
    --role-arn arn:aws:iam::123456789012:role/xaccounts3access \
    --role-session-name s3-access-example
```

輸出：

```
{
    "AssumedRoleUser": {
        "AssumedRoleId": "AROA3XFRBF535PLBIFPI4:s3-access-example",
        "Arn": "arn:aws:sts::123456789012:assumed-role/xaccounts3access/s3-access-
example"
```

```
    },
    "Credentials": {
        "SecretAccessKey": "9drTJvcXLB89EXAMPLELB8923FB892xMFI",
        "SessionToken": "AQoXdzELDDY//////////
wEaoAK1wvxJY12r2IrDFT2IvAzTCn3zHoZ7YNtpiQLF0MqZye/qwjzP2iEXAMPLEbw/
m3hsj8VBTkPORGvr9jM5sgP+w9IZWZnU+LWhmg
+a5fDi2oTGUYcdg9uexQ4mtCHIHfi4citgqZTgco40Yqr4lIlo4V2b2Dyauk0eYFNebHtYlFVgAUj
+7Indz3LU0aTWk1WKIjHmmMCIoTkyYp/k7kUG7moeEYKSitwQIi6Gjn+nyzM
+PtoA3685ixzv0R7i5rjQi0YE0lf1oeie3bDiNHncmzosRM6SFiPzSvp6h/32xQuZsjcypmwsPSDtTPYcs0+YN/8BRi2
IcrxSpnWEXAMPLEXSDFTAQAM6Dl9zR0tXoybnlrZIwMLlMi1Kcgo5OytwU=",
        "Expiration": "2016-03-15T00:05:07Z",
        "AccessKeyId": "ASIAJEXAMPLEXEG2JICEA"
    }
}
```

該命令的輸出包含存取金鑰、私密金鑰以及可用來向 AWS進行驗證的工作階段字符。

對於 AWS CLI 使用，您可以設定與角色相關聯的具名設定檔。當您使用設定檔時，CLI AWS 會呼叫 assume-role 並為您管理登入資料。如需詳細資訊，請參閱《CLI 使用者指南》中的在 AWS CLI 中使用 IAM 角色。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AssumeRole。

`assume-root`

以下程式碼範例顯示如何使用 `assume-root`。

AWS CLI

啟動特權工作階段

下列`assume-root`命令會擷取一組短期登入資料，可用來移除組織中成員帳戶設定錯誤的 Amazon S3 儲存貯體政策。

```
aws sts assume-root \
    --duration-seconds 900 \
    --target-principal 111122223333 \
    --task-policy-arn arn=arn:aws:iam::aws:policy/root-task/S3UnlockBucketPolicy
```

輸出：

```
{
```

```
    "Credentials": {
        "SecretAccessKey": "9drTJvcXLB89EXAMPLELB8923FB892xMFI",
        "SessionToken": "AQoXdzELDDY//////////
wEaoAK1wvxJY12r2IrDFT2IvAzTCn3zHoZ7YNtpiQLF0MqZye/qwjzP2iEXAMPLEbw/
m3hsj8VBTkPORGvr9jM5sgP+w9IZWZnU+LWhmg
+a5fDi2oTGUYcdg9uexQ4mtCHIHfi4citgqZTgco40Yqr4lIlo4V2b2Dyauk0eYFNebHtYlFVgAUj
+7Indz3LU0aTWk1WKIjHmmMCIoTkyYp/k7kUG7moeEYKSitwQIi6Gjn+nyzM
+PtoA3685ixzv0R7i5rjQi0YE0lf1oeie3bDiNHncmzosRM6SFiPzSvp6h/32xQuZsjcypmwsPSDtTPYcs0+YN/8BRi2
IcrxSpnWEXAMPLEXSDFTAQAM6Dl9zR0tXoybnlrZIwMLlMi1Kcgo5OytwU=",
        "Expiration": "2024-11-15T00:05:07Z",
        "AccessKeyId": "ASIAJEXAMPLEXEG2JICEA"
    },
    "SourceIdentity": "Alice",
}
```

命令的輸出包含存取金鑰、私密金鑰和工作階段字符，您可以用來在成員帳戶中執行特權動作。如需詳細資訊，請參閱《AWS IAM 使用者指南》中的在 AWS Organizations 成員帳戶上執行特權任務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssumeRoot。

decode-authorization-message

以下程式碼範例顯示如何使用 decode-authorization-message。

AWS CLI

對回應請求時傳回的編碼授權訊息進行解碼

以下 decode-authorization-message 範例將編碼訊息中與請求授權狀態相關的更多資訊進行解碼，以回應 Amazon Web Services 請求。

```
aws sts decode-authorization-message \
    --encoded-message EXAMPLEWodyRNrtlQARDip-
eTA6i6DrlUhHhPQrLWB_lAbl5pAKxl9mPDLexYcGBreyIKQC1BGBIpBKr3dFDkwqeO7e2NMk5j_hmzAiChJN-8oy3Ewi
Ojau7BMjOTWwOtHPhV_Zaz87yENdipr745EjQwRd5LaoL3vN8_5ZfA9UiBMKDgVh1gjqZJFUiQoubv78V1RbHNYnK44E
pOu3FZjwYStfvTb3GHs3-6rLribGO9jZOktkfE6vqxlFzLyeDr4P2ihC1wty9tArCvvGzIAUNmARQJ2VVWPxioqgoqCz
JWP5pwe_mAyqh0NLw-r1S56YC_90onj9A80sNrHlI-
tIiNd7tgNTYzDuPQYD2FMDBnp82V9eVmYGtPp5NIeSpuf3fOHanFuBZgENxZQZ2dlH3xJGMTtYayzZrRXjiq_SfX9zeB
FaoPIb8LmmKVBLpIB0iFhU9sEHPqKHVPi6jdxXqKaZaFGvYVmVOiuQdNQKuyk0p067POFrZECLjjOtNPBOZCcuEKEXAM
```

輸出：

```
{
    "DecodedMessage": "{\"allowed\":false,\"explicitDeny\":true,\"matchedStatements
\":{\"items\":[{\"statementId\":\"VisualEditor0\",\"effect\":\"DENY\",\"principals
\":{\"items\":[{\"value\":\"AROA123456789EXAMPLE\"}]},\"principalGroups
\":{\"items\":[]},\"actions\":{\"items\":[{\"value\":\"ec2:RunInstances
\"}]},\"resources\":{\"items\":[{\"value\":\"*\"}]},\"conditions\":{\"items
\":[]}}]},\"failures\":{\"items\":[]},\"context\":{\"principal\":{\"id\":
\"AROA123456789EXAMPLE:Ana\",\"arn\":\"arn:aws:sts::111122223333:assumed-role/
Developer/Ana\"},\"action\":\"RunInstances\",\"resource\":\"arn:aws:ec2:us-
east-1:111122223333:instance/*\",\"conditions\":{\"items\":[{\"key\":
\"ec2:MetadataHttpPutResponseHopLimit\",\"values\":{\"items\":[{\"value\":
\"2\"}]}},{\"key\":\"ec2:InstanceMarketType\",\"values\":{\"items\":[{\"value
\":\"on-demand\"}]}},{\"key\":\"aws:Resource\",\"values\":{\"items\":[{\"value
\":\"instance/*\"}]}},{\"key\":\"aws:Account\",\"values\":{\"items\":[{\"value
\":\"111122223333\"}]}},{\"key\":\"ec2:AvailabilityZone\",\"values\":{\"items\":
[{\"value\":\"us-east-1f\"}]}},{\"key\":\"ec2:ebsOptimized\",\"values\":{\"items
\":[{\"value\":\"false\"}]}},{\"key\":\"ec2:IsLaunchTemplateResource\",\"values
\":{\"items\":[{\"value\":\"false\"}]}},{\"key\":\"ec2:InstanceType\",\"values\":
{\"items\":[{\"value\":\"t2.micro\"}]}},{\"key\":\"ec2:RootDeviceType\",\"values
\":{\"items\":[{\"value\":\"ebs\"}]}},{\"key\":\"aws:Region\",\"values\":{\"items
\":[{\"value\":\"us-east-1\"}]}},{\"key\":\"ec2:MetadataHttpEndpoint\",\"values
\":{\"items\":[{\"value\":\"enabled\"}]}},{\"key\":\"aws:Service\",\"values\":
{\"items\":[{\"value\":\"ec2\"}]}},{\"key\":\"ec2:InstanceID\",\"values\":{\"items
\":[{\"value\":\"*\"}]}},{\"key\":\"ec2:MetadataHttpTokens\",\"values\":{\"items
\":[{\"value\":\"required\"}]}},{\"key\":\"aws:Type\",\"values\":{\"items\":
[{\"value\":\"instance\"}]}},{\"key\":\"ec2:Tenancy\",\"values\":{\"items\":
[{\"value\":\"default\"}]}},{\"key\":\"ec2:Region\",\"values\":{\"items\":[{\"value
\":\"us-east-1\"}]}},{\"key\":\"aws:ARN\",\"values\":{\"items\":[{\"value\":
\"arn:aws:ec2:us-east-1:111122223333:instance/*\"}]}}]}}}"
}
```

如需詳細資訊，請參閱 AWS IAM User Guide 中的 Policy evaluation logic。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 DecodeAuthorizationMessage。

get-caller-identity

以下程式碼範例顯示如何使用 `get-caller-identity`。

AWS CLI

取得目前 IAM 身分的詳細資訊

下列get-caller-identity命令會顯示用於驗證請求之 IAM 身分的相關資訊。發起人是 IAM 使用者。

```
aws sts get-caller-identity
```

輸出：

```
{
    "UserId": "AIDASAMPLEUSERID",
    "Account": "123456789012",
    "Arn": "arn:aws:iam::123456789012:user/DevAdmin"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCallerIdentity。

get-federation-token

以下程式碼範例顯示如何使用 get-federation-token。

AWS CLI

使用 IAM 使用者存取金鑰憑證傳回一組臨時安全憑證

以下 get-federation-token 範例會為使用者傳回一組臨時安全憑證 (包括存取金鑰 ID、私密存取金鑰和安全權杖)。必須使用 IAM 使用者的長期安全憑證來呼叫 GetFederationToken 操作。

```
aws sts get-federation-token \
    --name Bob \
    --policy file://myfile.json \
    --policy-arns arn=arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess \
    --duration-seconds 900
```

myfile.json 的內容：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:Describe*",
```

```
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "elasticloadbalancing:Describe*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:ListMetrics",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "autoscaling:Describe*",
            "Resource": "*"
        }
    ]
}
```

輸出：

```
{
    "Credentials": {
        "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "SessionToken": "EXAMPLEpZ2luX2VjEGoaCXVzLXdlc3QtMiJIMEYCIQC/
W9pL5ArQyDD5JwFL3/h5+WGopQ24GEXweNctwhi9sgIhAMkg
+MZE35iWM8s4r5Lr25f9rSTVPFH98G42QQunWMTfKq0DCOP/////////
wEQAxoMNDUyOTI1MTcwNTA3Igxuy3AOpuuoLsk3MJwqgQPg8QOd9HuoClUxq26wnc/nm
+eZLjHDyGf2KUAHK2DuaS/nrGSEXAMPLE",
        "Expiration": "2023-12-20T02:06:07+00:00"
    },
    "FederatedUser": {
        "FederatedUserId": "111122223333:Bob",
        "Arn": "arn:aws:sts::111122223333:federated-user/Bob"
    },
    "PackedPolicySize": 36
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的請求臨時安全憑證。

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetFederationToken。

`get-session-token`

以下程式碼範例顯示如何使用 `get-session-token`。

AWS CLI

為 IAM 身分取得一組短期憑證

下列 `get-session-token` 命令會為進行呼叫的 IAM 身分擷取一組短期憑證。產生的憑證可用於政策要求多重要素驗證 (MFA) 的請求。憑證會在產生後的 15 分鐘過期。

```
aws sts get-session-token \
    --duration-seconds 900 \
    --serial-number "YourMFADeviceSerialNumber" \
    --token-code 123456
```

輸出：

```
{
    "Credentials": {
        "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJOgQs8IZZaIv2BXIa2R4OlgkBN9bkUDNCJiBeb/
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wnOE49mRl/+OtkIKGO7fAE",
        "Expiration": "2020-05-19T18:06:10+00:00"
    }
}
```

如需詳細資訊，請參閱《AWS IAM 使用者指南》中的請求臨時安全憑證。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetSessionToken。

支援 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 支援。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`add-attachments-to-set`

以下程式碼範例顯示如何使用 `add-attachments-to-set`。

AWS CLI

將附件新增至集合

下列`add-attachments-to-set`範例會將映像新增至您可以為 AWS 帳戶中的支援案例指定的集合。

```
aws support add-attachments-to-set \
    --attachment-set-id "as-2f5a6faa2a4a1e600-mu-nk5xQlBr70-
G1cUos5LZkd38KOAHZa9BMDVzNEXAMPLE" \
    --attachments fileName=troubleshoot-screenshot.png,data=base64-encoded-string
```

輸出：

```
{
    "attachmentSetId": "as-2f5a6faa2a4a1e600-mu-nk5xQlBr70-
G1cUos5LZkd38KOAHZa9BMDVzNEXAMPLE",
    "expiryTime": "2020-05-14T17:04:40.790+0000"
}
```

如需詳細資訊，請參閱 AWS 支援使用者指南中的案例管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddAttachmentsToSet。

`add-communication-to-case`

以下程式碼範例顯示如何使用 `add-communication-to-case`。

AWS CLI

將通訊新增至案例

下列add-communication-to-case範例會將通訊新增至您 AWS 帳戶中的支援案例。

```
aws support add-communication-to-case \
    --case-id "case-12345678910-2013-c4c1d2bf33c5cf47" \
    --communication-body "I'm attaching a set of images to this case." \
    --cc-email-addresses "myemail@example.com" \
    --attachment-set-id "as-2f5a6faa2a4a1e600-mu-nk5xQlBr70-
G1cUos5LZkd38KOAHZa9BMDVzNEXAMPLE"
```

輸出：

```
{
    "result": true
}
```

如需詳細資訊，請參閱 AWS 支援使用者指南中的案例管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddCommunicationToCase。

create-case

以下程式碼範例顯示如何使用 create-case。

AWS CLI

建立案例

下列create-case範例會為您的 AWS 帳戶建立支援案例。

```
aws support create-case \
    --category-code "using-aws" \
    --cc-email-addresses "myemail@example.com" \
    --communication-body "I want to learn more about an AWS service." \
    --issue-type "technical" \
    --language "en" \
    --service-code "general-info" \
    --severity-code "low" \
    --subject "Question about my account"
```

輸出：

```
{
    "caseId": "case-12345678910-2013-c4c1d2bf33c5cf47"
}
```

如需詳細資訊，請參閱 AWS 支援使用者指南中的案例管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCase。

describe-attachment

以下程式碼範例顯示如何使用 describe-attachment。

AWS CLI

描述附件

下列describe-attachment範例會傳回具有指定 ID 之附件的相關資訊。

```
aws support describe-attachment \
    --attachment-id "attachment-KBnjRNrePd9D6Jx0-Mm00xZuDEaL2JAj_0-
gJv9qqDooTipsz3V1Nb19rCfkZneeQeDPgp8X1iVJyHH7UuhZDdNeqGoduZsPrAhyMakqlc60-
iJjL5HqyYGiT1FG8EXAMPLE"
```

輸出：

```
{
    "attachment": {
        "fileName": "troubleshoot-screenshot.png",
        "data": "base64-blob"
    }
}
```

如需詳細資訊，請參閱 AWS 支援使用者指南中的案例管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeAttachment。

describe-cases

以下程式碼範例顯示如何使用 describe-cases。

AWS CLI

描述案例

下列describe-cases範例會傳回您 AWS 帳戶中指定支援案例的相關資訊。

```
aws support describe-cases \
    --display-id "1234567890" \
    --after-time "2020-03-23T21:31:47.774Z" \
    --include-resolved-cases \
    --language "en" \
    --no-include-communications \
    --max-item 1
```

輸出：

```
{
    "cases": [
        {
            "status": "resolved",
            "ccEmailAddresses": [],
            "timeCreated": "2020-03-23T21:31:47.774Z",
            "caseId": "case-12345678910-2013-c4c1d2bf33c5cf47",
            "severityCode": "low",
            "language": "en",
            "categoryCode": "using-aws",
            "serviceCode": "general-info",
            "submittedBy": "myemail@example.com",
            "displayId": "1234567890",
            "subject": "Question about my account"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS 支援使用者指南中的案例管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCases。

describe-communications

以下程式碼範例顯示如何使用 describe-communications。

AWS CLI

描述案例的最新通訊

下列describe-communications範例會傳回您 AWS 帳戶中指定支援案例的最新通訊。

```
aws support describe-communications \
    --case-id "case-12345678910-2013-c4c1d2bf33c5cf47" \
    --after-time "2020-03-23T21:31:47.774Z" \
    --max-item 1
```

輸出：

```
{
    "communications": [
        {
            "body": "I want to learn more about an AWS service.",
            "attachmentSet": [],
            "caseId": "case-12345678910-2013-c4c1d2bf33c5cf47",
            "timeCreated": "2020-05-12T23:12:35.000Z",
            "submittedBy": "Amazon Web Services"
        }
    ],
    "NextToken": "eyJuZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQEXAMPLE=="
}
```

如需詳細資訊，請參閱 AWS 支援使用者指南中的案例管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeCommunications。

describe-services

以下程式碼範例顯示如何使用 describe-services。

AWS CLI

列出 AWS 服務和 服務類別

下列describe-services範例列出請求一般資訊的可用服務類別。

```
aws support describe-services \
```

```
    --service-code-list "general-info"
```

輸出：

```
{
    "services": [
        {
            "code": "general-info",
            "name": "General Info and Getting Started",
            "categories": [
                {
                    "code": "charges",
                    "name": "How Will I Be Charged?"
                },
                {
                    "code": "gdpr-queries",
                    "name": "Data Privacy Query"
                },
                {
                    "code": "reserved-instances",
                    "name": "Reserved Instances"
                },
                {
                    "code": "resource",
                    "name": "Where is my Resource?"
                },
                {
                    "code": "using-aws",
                    "name": "Using AWS & Services"
                },
                {
                    "code": "free-tier",
                    "name": "Free Tier"
                },
                {
                    "code": "security-and-compliance",
                    "name": "Security & Compliance"
                },
                {
                    "code": "account-structure",
                    "name": "Account Structure"
                }
            ]
```

```
        }
    ]
}
```

如需詳細資訊，請參閱 AWS 支援使用者指南中的案例管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeServices。

describe-severity-levels

以下程式碼範例顯示如何使用 `describe-severity-levels`。

AWS CLI

列出可用的嚴重性等級

下列`describe-severity-levels`範例列出支援案例的可用嚴重性等級。

```
aws support describe-severity-levels
```

輸出：

```
{
    "severityLevels": [
        {
            "code": "low",
            "name": "Low"
        },
        {
            "code": "normal",
            "name": "Normal"
        },
        {
            "code": "high",
            "name": "High"
        },
        {
            "code": "urgent",
            "name": "Urgent"
        },
        {
            "code": "critical",
```

```
            "name": "Critical"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS 支援使用者指南中的選擇嚴重性。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSeverityLevels。

describe-trusted-advisor-check-refresh-statuses

以下程式碼範例顯示如何使用 describe-trusted-advisor-check-refresh-statuses。

AWS CLI

列出 AWS Trusted Advisor 檢查的重新整理狀態

下列describe-trusted-advisor-check-refresh-statuses範例列出兩個 Trusted Advisor 檢查的重新整理狀態：Amazon S3 儲存貯體許可和 IAM 使用。

```
aws support describe-trusted-advisor-check-refresh-statuses \
    --check-id "Pfx0RwqBli" "zXCkfM1nI3"
```

輸出：

```
{
    "statuses": [
        {
            "checkId": "Pfx0RwqBli",
            "status": "none",
            "millisUntilNextRefreshable": 0
        },
        {
            "checkId": "zXCkfM1nI3",
            "status": "none",
            "millisUntilNextRefreshable": 0
        }
    ]
}
```

如需詳細資訊，請參閱 AWS 支援使用者指南中的 AWS Trusted Advisor。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTrustedAdvisorCheckRefreshStatuses。

describe-trusted-advisor-check-result

以下程式碼範例顯示如何使用 describe-trusted-advisor-check-result。

AWS CLI

列出 an AWS Trusted Advisor 檢查的結果

下列describe-trusted-advisor-check-result範例列出 IAM 使用檢查的結果。

```
aws support describe-trusted-advisor-check-result \
    --check-id "zXCkfM1nI3"
```

輸出：

```
{
    "result": {
        "checkId": "zXCkfM1nI3",
        "timestamp": "2020-05-13T21:38:05Z",
        "status": "ok",
        "resourcesSummary": {
            "resourcesProcessed": 1,
            "resourcesFlagged": 0,
            "resourcesIgnored": 0,
            "resourcesSuppressed": 0
        },
        "categorySpecificSummary": {
            "costOptimizing": {
                "estimatedMonthlySavings": 0.0,
                "estimatedPercentMonthlySavings": 0.0
            }
        },
        "flaggedResources": [
            {
                "status": "ok",
                "resourceId": "47DEQpj8HBSa-_TImW-5JCeuQeRkm5NMpJWZEXAMPLE",
                "isSuppressed": false
            }
        ]
```

```
    }
}
```

如需詳細資訊，請參閱 AWS 支援使用者指南中的 AWS Trusted Advisor。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTrustedAdvisorCheckResult。

`describe-trusted-advisor-check-summaries`

以下程式碼範例顯示如何使用 `describe-trusted-advisor-check-summaries`。

AWS CLI

列出 AWS Trusted Advisor 檢查的摘要

下列`describe-trusted-advisor-check-summaries`範例列出兩個 Trusted Advisor 檢查的結果：Amazon S3 儲存貯體許可和 IAM 使用。

```
aws support describe-trusted-advisor-check-summaries \
    --check-ids "Pfx0RwqBli" "zXCkfM1nI3"
```

輸出：

```
{
    "summaries": [
        {
            "checkId": "Pfx0RwqBli",
            "timestamp": "2020-05-13T21:38:12Z",
            "status": "ok",
            "hasFlaggedResources": true,
            "resourcesSummary": {
                "resourcesProcessed": 44,
                "resourcesFlagged": 0,
                "resourcesIgnored": 0,
                "resourcesSuppressed": 0
            },
            "categorySpecificSummary": {
                "costOptimizing": {
                    "estimatedMonthlySavings": 0.0,
                    "estimatedPercentMonthlySavings": 0.0
                }
            }
        },
```

```
        {
            "checkId": "zXCkfM1nI3",
            "timestamp": "2020-05-13T21:38:05Z",
            "status": "ok",
            "hasFlaggedResources": true,
            "resourcesSummary": {
                "resourcesProcessed": 1,
                "resourcesFlagged": 0,
                "resourcesIgnored": 0,
                "resourcesSuppressed": 0
            },
            "categorySpecificSummary": {
                "costOptimizing": {
                    "estimatedMonthlySavings": 0.0,
                    "estimatedPercentMonthlySavings": 0.0
                }
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS 支援使用者指南中的 AWS Trusted Advisor。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTrustedAdvisorCheckSummaries。

describe-trusted-advisor-checks

以下程式碼範例顯示如何使用 `describe-trusted-advisor-checks`。

AWS CLI

列出可用的 AWS Trusted Advisor 檢查

下列`describe-trusted-advisor-checks`範例列出您 AWS 帳戶中可用的 Trusted Advisor 檢查。此資訊包括檢查名稱、ID、描述、類別和中繼資料。請注意，輸出會縮短以供讀取。

```
aws support describe-trusted-advisor-checks \
    --language "en"
```

輸出：

```
{
```

```
    "checks": [
        {
            "id": "zXCkfM1nI3",
            "name": "IAM Use",
            "description": "Checks for your use of AWS Identity and Access
 Management (IAM). You can use IAM to create users, groups, and roles in AWS, and
 you can use permissions to control access to AWS resources. \n<br>\n<br>\n<b>Alert
 Criteria</b><br>\nYellow: No IAM users have been created for this account.\n<br>
\n<br>\n<b>Recommended Action</b><br>\nCreate one or more IAM users and groups in
 your account. You can then create additional users whose permissions are limited
 to perform specific tasks in your AWS environment. For more information, see <a
 href=\"https://docs.aws.amazon.com/IAM/latest/UserGuide/IAMGettingStarted.html\"
 target=\"_blank\">Getting Started</a>. \n<br><br>\n<b>Additional Resources</b><br>
\n<a href=\"https://docs.aws.amazon.com/IAM/latest/UserGuide/IAM_Introduction.html\"
 target=\"_blank\">What Is IAM?</a>",
            "category": "security",
            "metadata": []
        }
    ]
}
```

如需詳細資訊，請參閱 AWS 支援使用者指南中的 AWS Trusted Advisor。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTrustedAdvisorChecks。

`refresh-trusted-advisor-check`

以下程式碼範例顯示如何使用 `refresh-trusted-advisor-check`。

AWS CLI

重新整理 an AWS Trusted Advisor 檢查

下列`refresh-trusted-advisor-check`範例會重新整理您 AWS 帳戶中的 Amazon S3 儲存貯體許可 Trusted Advisor 檢查。

```
aws support refresh-trusted-advisor-check \
    --check-id "Pfx0RwqBli"
```

輸出：

```
{
    "status": {
```

```
        "checkId": "Pfx0RwqBli",
        "status": "enqueued",
        "millisUntilNextRefreshable": 3599992
    }
}
```

如需詳細資訊，請參閱 AWS 支援使用者指南中的 AWS Trusted Advisor。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RefreshTrustedAdvisorCheck。

`resolve-case`

以下程式碼範例顯示如何使用 `resolve-case`。

AWS CLI

解決支援案例

下列`resolve-case`範例會解決您 AWS 帳戶中的支援案例。

```
aws support resolve-case \
    --case-id "case-12345678910-2013-c4c1d2bf33c5cf47"
```

輸出：

```
{
    "finalCaseStatus": "resolved",
    "initialCaseStatus": "work-in-progress"
}
```

如需詳細資訊，請參閱 AWS 支援使用者指南中的案例管理。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResolveCase。

使用 的 Amazon SWF 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon SWF 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`count-closed-workflow-executions`

以下程式碼範例顯示如何使用 `count-closed-workflow-executions`。

AWS CLI

計數已關閉的工作流程執行

您可以使用 `swf count-closed-workflow-executions` 來擷取指定網域的已關閉工作流程執行數目。您可以指定篩選條件來計算特定類別的執行。

需要 `--domain`和 `--close-time-filter`或 `--start-time-filter`引數。所有其他引數都是選用的。

```
aws swf count-closed-workflow-executions \
    --domain DataFrobtzz \
    --close-time-filter "{ \"latestDate\" : 1377129600, \"oldestDate\" :
 1370044800 }"
```

輸出：

```
{
    "count": 2,
    "truncated": false
}
```

如果「截斷」為 `true`，則「計數」代表 Amazon SWF 可傳回的最大數量。任何進一步的結果都會截斷。

若要減少傳回的結果數量，您可以：

修改 `--close-time-filter`或 `--start-time-filter`值以縮小搜尋的時間範圍。這些都是互斥的：您只能在 請求中指定其中一個。請使用 `--close-status-filter`、 `--execution-`

filter--tag-filter或 --type-filter引數來進一步篩選結果。不過，這些引數也是互斥的。

請參閱《Amazon Simple Workflow Service API 參考》中的 AlsoCountClosedWorkflowExecutions

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CountClosedWorkflowExecutions。

count-open-workflow-executions

以下程式碼範例顯示如何使用 count-open-workflow-executions。

AWS CLI

計數開放工作流程執行

您可以使用 swf count-open-workflow-executions 來擷取指定網域的開啟工作流程執行數目。您可以指定篩選條件來計算特定類別的執行。

需要 --domain和 --start-time-filter引數。所有其他引數都是選用的。

```
aws swf count-open-workflow-executions \
    --domain DataFrobtzz \
    --start-time-filter "{ \"latestDate\" : 1377129600, \"oldestDate\" :
 1370044800 }"
```

輸出：

```
{
    "count": 4,
    "truncated": false
}
```

如果「截斷」為 true，則「計數」代表 Amazon SWF 可傳回的最大數量。任何進一步的結果都會截斷。

若要減少傳回的結果數量，您可以：

修改--start-time-filter值以縮小搜尋的時間範圍。使用 --close-status-filter、 --execution-filter--tag-filter或 --type-filter引數進一步篩選結果。這些都是互斥：您只能在請求中指定其中一個。

如需詳細資訊，請參閱《Amazon Simple Workflow Service API 參考》中的 CountOpenWorkflowExecutions

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CountOpenWorkflowExecutions。

deprecate-domain

以下程式碼範例顯示如何使用 deprecate-domain。

AWS CLI

棄用網域

若要棄用網域 (您仍可以看見該網域，但無法在上面建立新的工作流程執行或註冊類型)，請使用 swf deprecate-domain。只有一個必要參數，--name，其採用您要棄用的網域名稱。

```
aws swf deprecate-domain \
    --name MyNeatNewDomain ""
```

如同 register-domain，不會有輸出傳回。不過，如果您使用 list-domains來檢視已註冊的網域，您會看到網域已棄用，且不再出現在傳回的資料中。

```
aws swf list-domains \
    --registration-status REGISTERED
        {
            "domainInfos": [
                {
                    "status": "REGISTERED",
                    "name": "DataFrobotz"
                },
                {
                    "status": "REGISTERED",
                    "name": "erontest"
                }
            ]
        }
```

如果您--registration-status DEPRECATED搭配 使用 list-domains，您會看到已棄用的網域。

```
aws swf list-domains \
```

```
    --registration-status DEPRECATED
        {
            "domainInfos": [
                {
                    "status": "DEPRECATED",
                    "name": "MyNeatNewDomain"
                }
            ]
        }
```

您仍然可以使用 describe-domain 來取得已棄用網域的相關資訊。

```
aws swf describe-domain \
    --name MyNeatNewDomain
        {
            "domainInfo": {
                "status": "DEPRECATED",
                "name": "MyNeatNewDomain"
            },
            "configuration": {
                "workflowExecutionRetentionPeriodInDays": "0"
            }
        }
```

請參閱《Amazon Simple Workflow Service API 參考》中的 AlsoDeprecateDomain

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeprecateDomain。

describe-domain

以下程式碼範例顯示如何使用 describe-domain。

AWS CLI

取得網域的相關資訊

若要取得特定網域的詳細資訊，請使用 swf describe-domain命令。只有一個必要參數：--name，其採用您想要取得資訊的網域名稱。

```
aws swf describe-domain \
    --name DataFrobotz
        {
```

```
            "domainInfo": {
                "status": "REGISTERED",
                "name": "DataFrobotz"
            },
            "configuration": {
                "workflowExecutionRetentionPeriodInDays": "1"
            }
        }
```

您也可以使用 describe-domain 取得已棄用網域的相關資訊。

```
aws swf describe-domain \
    --name MyNeatNewDomain
        {
            "domainInfo": {
                "status": "DEPRECATED",
                "name": "MyNeatNewDomain"
            },
            "configuration": {
                "workflowExecutionRetentionPeriodInDays": "0"
            }
        }
```

請參閱《Amazon Simple Workflow Service API 參考》中的 AlsoDescribeDomain

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDomain。

list-activity-types

以下程式碼範例顯示如何使用 list-activity-types。

AWS CLI

列出活動類型

若要取得網域的活動類型清單，請使用 swf list-activity-types。需要 --domain和 --registration-status引數。

```
aws swf list-activity-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED
```

輸出：

```
{
    "typeInfos": [
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.451,
            "activityType": {
                "version": "1",
                "name": "confirm-user-email"
            },
            "description": "subscribe confirm-user-email activity"
        },
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.709,
            "activityType": {
                "version": "1",
                "name": "confirm-user-phone"
            },
            "description": "subscribe confirm-user-phone activity"
        },
        {
            "status": "REGISTERED",
            "creationDate": 1371454149.871,
            "activityType": {
                "version": "1",
                "name": "get-subscription-info"
            },
            "description": "subscribe get-subscription-info activity"
        },
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.909,
            "activityType": {
                "version": "1",
                "name": "send-subscription-success"
            },
            "description": "subscribe send-subscription-success activity"
        },
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.085,
            "activityType": {
```

```
                "version": "1",
                "name": "subscribe-user-sns"
            },
            "description": "subscribe subscribe-user-sns activity"
        }
    ]
}
```

您可以使用 --name引數來僅選取具有特定名稱的活動類型：

```
aws swf list-activity-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED \
    --name "send-subscription-success"
```

輸出：

```
{
    "typeInfos": [
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.909,
            "activityType": {
                "version": "1",
                "name": "send-subscription-success"
            },
            "description": "subscribe send-subscription-success activity"
        }
    ]
}
```

若要擷取頁面中的結果，您可以設定 --maximum-page-size 引數。如果傳回的結果超過結果頁面，則會在結果集中傳回「nextPageToken」：

```
aws swf list-activity-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED \
    --maximum-page-size 2
```

輸出：

```
{
```

```
    "nextPageToken": "AAAAKgAAAAEAAAAAAAAAA1Gp1BelJq
+PmHvAnDxJYbup8+0R4LVtbXLDl7QNY7C3OpHo9Sz06D/GuFz1OyC73umBQ1tOPJ/gC/
aYpzDMqUIWIA1T9W0s2DryyZX4OC/6Lhk9/
o5kdsuWMSBkHhgaZjgwp3WJINIFJFdaSMxY2vYAX7AtRtpcqJuBDDRE9RaRqDGYqIYUMltarkiqpSY1ZVveBasBvlvyU
WGAaqehiDz7/JzLT/wWNNUMOd+Nhe",
    "typeInfos": [
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.451,
            "activityType": {
                "version": "1",
                "name": "confirm-user-email"
            },
            "description": "subscribe confirm-user-email activity"
        },
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.709,
            "activityType": {
                "version": "1",
                "name": "confirm-user-phone"
            },
            "description": "subscribe confirm-user-phone activity"
        }
    ]
}
```

您可以在 --next-page-token引數list-activity-types中將 nextPageToken 值傳遞給下一個呼叫，擷取下一頁的結果：

```
aws swf list-activity-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED \
    --maximum-page-size 2 \
    --next-page-token "AAAAKgAAAAEAAAAAAAAAA1Gp1BelJq
+PmHvAnDxJYbup8+0R4LVtbXLDl7QNY7C3OpHo9Sz06D/GuFz1OyC73umBQ1tOPJ/gC/
aYpzDMqUIWIA1T9W0s2DryyZX4OC/6Lhk9/
o5kdsuWMSBkHhgaZjgwp3WJINIFJFdaSMxY2vYAX7AtRtpcqJuBDDRE9RaRqDGYqIYUMltarkiqpSY1ZVveBasBvlvyU
WGAaqehiDz7/JzLT/wWNNUMOd+Nhe"
```

輸出：

```
{
```

```
    "nextPageToken": "AAAAKgAAAAEAAAAAAAAAAw+7LZ4GRZPzTqBHsp2wBxWB8m1sgLCclgCuq3J+h/
m3+vOfFqtkcjLwV5cc4OjNAzTCuq/
XcylPumGwkjbajtqpZpbqOcVNfjFxGoi0LB2Olbvv0krbUISBvlpFPmSWpDSZJsxg5UxCcweteSlFn1PNSZ/
MoinBZo8OTkjMuzcsTuKOzH9wCaR8ITcALJ3SaqHU3pyIRS5hPmFA3OLIc8zaAepjlaujo6hntNSCruB4"
    "typeInfos": [
        {
            "status": "REGISTERED",
            "creationDate": 1371454149.871,
            "activityType": {
                "version": "1",
                "name": "get-subscription-info"
            },
            "description": "subscribe get-subscription-info activity"
        },
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.909,
            "activityType": {
                "version": "1",
                "name": "send-subscription-success"
            },
            "description": "subscribe send-subscription-success activity"
        }
    ]
}
```

如果還有更多結果要傳回，則會傳回「nextPageToken」與結果。當沒有更多要傳回的結果頁面時，結果集中將不會傳回「nextPageToken」。

您可以使用 `--reverse-order`引數來反轉傳回結果的順序。這也會影響分頁結果。

```
aws swf list-activity-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED \
    --maximum-page-size 2 \
    --reverse-order
```

輸出：

```
{
    "nextPageToken": "AAAAKgAAAAEAAAAAAAAAwXcpu5ePSyQkrC
+8WMbmSrenuZC2ZkIXQYBPB/b9xIOVkj+bMEFhGj0KmmJ4rF7iddhjf7UMYCsfGkEn7mk
```

```
+yMCgVc1JxDWmB0EH46bhcmcLmYNQihMDmUWocpr7To6/R7CLu0St1gkFayxOidJXErQW0zdNfQaIWAnF/
cwioBbXlkz1fQzmDeU3M5oYGMPQIrUqkPq7pMEW0q0lK5eDN97NzFYdZZ/rlcLDWPZhUjY",
    "typeInfos": [
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.085,
            "activityType": {
                "version": "1",
                "name": "subscribe-user-sns"
            },
            "description": "subscribe subscribe-user-sns activity"
        },
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.909,
            "activityType": {
                "version": "1",
                "name": "send-subscription-success"
            },
            "description": "subscribe send-subscription-success activity"
        }
    ]
}
```

請參閱《Amazon Simple Workflow Service API 參考》中的 AlsoListActivityTypes

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListActivityTypes。

list-domains

以下程式碼範例顯示如何使用 `list-domains`。

AWS CLI

範例 1：列出已註冊的網域

下列`list-domains`命令範例列出您已為 帳戶註冊的 REGISTERED SWF 網域。

```
aws swf list-domains \
    --registration-status REGISTERED
```

輸出：

```
{
  "domainInfos": [
    {
      "status": "REGISTERED",
      "name": "DataFrobotz"
    },
    {
      "status": "REGISTERED",
      "name": "erontest"
    }
  ]
}
```

如需詳細資訊，請參閱《Amazon Simple Workflow Service API 參考》中的 ListDomains

範例 2：列出已棄用的網域

下列list-domains命令範例列出您已為 帳戶註冊的 DEPRECATED SWF 網域。已棄用網域是無法註冊新工作流程或活動的網域，但仍可查詢。

```
aws swf list-domains \
    --registration-status DEPRECATED
```

輸出：

```
{
  "domainInfos": [
    {
      "status": "DEPRECATED",
      "name": "MyNeatNewDomain"
    }
  ]
}
```

如需詳細資訊，請參閱《Amazon Simple Workflow Service API 參考》中的 ListDomains

範例 3：列出已註冊網域的第一頁

下列list-domains命令範例列出您使用 --maximum-page-size選項為帳戶註冊的 REGISTERED SWF 網域第一頁。

```
aws swf list-domains \
    --registration-status REGISTERED \
    --maximum-page-size 1
```

輸出：

```
{
    "domainInfos": [
        {
            "status": "REGISTERED",
            "name": "DataFrobotz"
        }
    ],
"nextPageToken": "AAAAKgAAAAEAAAAAAAAAA2QJKNtidVgd49TTeNwYcpD
+QKT2ynuEbibcQWe2QKrslMGe63gpS0MgZGpcpoKttL4OCXRFn98Xif557it
+wSZUsvUDtImjDLvguyuyyFdIZtvIxIKEOPm3k2r4OjAGaFsGOuVbrKljvla7wdU7FYH3OlkNCP8b7PBj9SBkUyGoiAg
}
```

如需詳細資訊，請參閱《Amazon Simple Workflow Service API 參考》中的 ListDomains

範例 4：列出已註冊網域的指定單一頁面

下列list-domains命令範例列出您使用 --maximum-page-size選項為帳戶註冊的 REGISTERED SWF 網域第一頁。

當您再次進行呼叫時，這次在 --next-page-token 引數nextPageToken中提供 的值，您會收到另一頁結果。

```
aws swf list-domains \
    --registration-status REGISTERED \
    --maximum-page-size 1 \
    --next-page-token "AAAAKgAAAAEAAAAAAAAAA2QJKNtidVgd49TTeNwYcpD
+QKT2ynuEbibcQWe2QKrslMGe63gpS0MgZGpcpoKttL4OCXRFn98Xif557it
+wSZUsvUDtImjDLvguyuyyFdIZtvIxIKEOPm3k2r4OjAGaFsGOuVbrKljvla7wdU7FYH3OlkNCP8b7PBj9SBkUyGoiAg
```

輸出：

```
{
    "domainInfos": [
        {
            "status": "REGISTERED",
```

```
            "name": "erontest"
        }
    ]
}
```

如果沒有後續的結果頁面可擷取，nextPageToken 就不會傳回到結果中。

如需詳細資訊，請參閱《Amazon Simple Workflow Service API 參考》中的 ListDomains

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDomains。

list-workflow-types

以下程式碼範例顯示如何使用 list-workflow-types。

AWS CLI

列出工作流程類型

若要取得網域的工作流程類型清單，請使用 swf list-workflow-types。需要 --domain和 --registration-status引數。以下是簡單的範例。

```
aws swf list-workflow-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED
```

輸出：

```
{
    "typeInfos": [
        {
            "status": "REGISTERED",
            "creationDate": 1371454149.598,
            "description": "DataFrobtzz subscribe workflow",
            "workflowType": {
                "version": "v3",
                "name": "subscribe"
            }
        }
    ]
}
```

如同 `list-activity-types`，您可以使用 `--name`引數來僅選取具有特定名稱的工作流程類型，並使用 `--maximum-page-size`引數與 協調`--next-page-token`以分頁結果。若要反轉傳回結果的順序，請使用 `--reverse-order`。

請參閱《Amazon Simple Workflow Service API 參考》中的 AlsoListWorkflowTypes

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListWorkflowTypes。

`register-domain`

以下程式碼範例顯示如何使用 `register-domain`。

AWS CLI

註冊網域

您可以使用 AWS CLI 來註冊新網域。使用 `swf register-domain` 命令。有兩個必要參數， `--name`採用網域名稱，而 `--workflow-execution-retention-period-in-days`採用整數來指定在此網域上保留工作流程執行資料的天數，最長為 90 天 （ 如需詳細資訊，請參閱 SWF 常見問答集 <https://aws.amazon.com/swf/faqs/#retain_limit：//>)。在指定的天數過後，工作流程執行資料將不會保留。

```
aws swf register-domain \
    --name MyNeatNewDomain \
    --workflow-execution-retention-period-in-days 0
    ""
```

註冊網域時，不會傳回任何內容 ("")，但可以使用 `swf list-domains` 或 `swf describe-domain` 來查看新的網域。

```
aws swf list-domains \
    --registration-status REGISTERED
        {
            "domainInfos": [
                {
                    "status": "REGISTERED",
                    "name": "DataFrobotz"
                },
                {
                    "status": "REGISTERED",
                    "name": "MyNeatNewDomain"
                },
```

```
            {
                "status": "REGISTERED",
                "name": "erontest"
            }
        ]
    }
```

使用 `swf describe-domain`：

```
aws swf describe-domain --
name MyNeatNewDomain
{
    "domainInfo": {
        "status": "REGISTERED",
        "name": "MyNeatNewDomain"
    },
    "configuration": {
        "workflowExecutionRetentionPeriodInDays": "0"

  }
}
```

請參閱《Amazon Simple Workflow Service API 參考》中的 AlsoRegisterDomain

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterDomain。

register-workflow-type

以下程式碼範例顯示如何使用 `register-workflow-type`。

AWS CLI

註冊工作流程類型

若要向 CLI AWS 註冊工作流程類型，請使用 `swf register-workflow-type`命令。

```
aws swf register-workflow-type \
    --domain DataFrobtzz \
    --name "MySimpleWorkflow" \
    --workflow-version "v1"
```

如果成功，命令不會產生輸出。

發生錯誤時（例如，如果您嘗試註冊相同的工作流程類型兩次，或指定不存在的網域），您會在 JSON 中收到回應。

```
{
    "message": "WorkflowType=[name=MySimpleWorkflow, version=v1]",
    "__type": "com.amazonaws.swf.base.model#TypeAlreadyExistsFault"
}
```

`--workflow-version` 需要 `--domain`、 `--name`和 。您也可以設定工作流程描述、逾時和子工作流程政策。

如需詳細資訊，請參閱《Amazon Simple Workflow Service API 參考》中的 RegisterWorkflowType

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterWorkflowType。

使用 的 Systems Manager 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Systems Manager 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`add-tags-to-resource`

以下程式碼範例顯示如何使用 `add-tags-to-resource`。

AWS CLI

範例 1：將標籤新增至維護時段

下列 `add-tags-to-resource` 範例示範將標籤新增至指定的維護時段。

```
aws ssm add-tags-to-resource \
```

```
    --resource-type "MaintenanceWindow" \
    --resource-id "mw-03eb9db428EXAMPLE" \
    --tags "Key=Stack,Value=Production"
```

此命令不會產生輸出。

範例 2：將標籤新增至參數

下列 add-tags-to-resource 範例示範將兩個標籤新增至指定的參數。

```
aws ssm add-tags-to-resource \
    --resource-type "Parameter" \
    --resource-id "My-Parameter" \
    --tags '[{"Key":"Region","Value":"East"},{"Key":"Environment",
 "Value":"Production"}]'
```

此命令不會產生輸出。

範例 3：將標籤新增至 SSM 文件

下列 add-tags-to-resource 範例示範將標籤新增至指定的文件。

```
aws ssm add-tags-to-resource \
    --resource-type "Document" \
    --resource-id "My-Document" \
    --tags "Key=Quarter,Value=Q322"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Tagging Systems Manager resources。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AddTagsToResource。

associate-ops-item-related-item

以下程式碼範例顯示如何使用 associate-ops-item-related-item。

AWS CLI

建立相關項目的關聯

下列associate-ops-item-related-item範例會將相關項目與 OpsItem 建立關聯。

```
aws ssm associate-ops-item-related-item \
    --ops-item-id "oi-649fExample" \
    --association-type "RelatesTo" \
    --resource-type "AWS::SSMIncidents::IncidentRecord" \
    --resource-uri "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/c2bde883-f7d5-343a-b13a-bf5fe9ea689f"
```

輸出：

```
{
    "AssociationId": "61d7178d-a30d-4bc5-9b4e-a9e74EXAMPLE"
}
```

如需詳細資訊，請參閱 Systems Manager 使用者指南中的在 OpsCenter 中使用 Incident Manager 事件。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateOpsItemRelatedItem。

cancel-command

以下程式碼範例顯示如何使用 cancel-command。

AWS CLI

範例 1：取消所有執行個體的命令

下列 cancel-command 範例嘗試取消已針對所有執行個體執行的指定命令。

```
aws ssm cancel-command \
    --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE"
```

此命令不會產生輸出。

範例 2：取消特定執行個體的命令

下列 cancel-command 範例嘗試僅取消指定執行個體的命令。

```
aws ssm cancel-command \
    --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE"
    --instance-ids "i-02573cafcfEXAMPLE"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Tagging Systems Manager Parameters。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CancelCommand。

cancel-maintenance-window-execution

以下程式碼範例顯示如何使用 cancel-maintenance-window-execution。

AWS CLI

取消維護時段執行

此cancel-maintenance-window-execution範例會停止已進行中的指定維護時段執行。

```
aws ssm cancel-maintenance-window-execution \
    --window-execution-id j2l8d5b5c-mw66-tk4d-r3g9-1d4d1EXAMPLE
```

輸出：

```
{
    "WindowExecutionId": "j2l8d5b5c-mw66-tk4d-r3g9-1d4d1EXAMPLE"
}
```

如需詳細資訊，請參閱 Systems Manager 使用者指南中的 Systems Manager 維護 Windows 教學課程 (AWS CLI)。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CancelMaintenanceWindowExecution。

create-activation

以下程式碼範例顯示如何使用 create-activation。

AWS CLI

建立受管執行個體啟用

下列 create-activation 範例示範建立受管執行個體啟用。

```
aws ssm create-activation \
    --default-instance-name "HybridWebServers" \
```

```
    --iam-role "HybridWebServersRole" \
    --registration-limit 5
```

輸出：

```
{
    "ActivationId": "5743558d-563b-4457-8682-d16c3EXAMPLE",
    "ActivationCode": "dRmgnYaFv567vEXAMPLE"
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Step 4: Create a Managed-Instance Activation for a Hybrid Environment。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateActivation。

create-association-batch

以下程式碼範例顯示如何使用 create-association-batch。

AWS CLI

建立多個關聯

此範例會將組態文件與多個執行個體建立關聯。輸出會傳回成功操作和失敗操作的清單 (如適用)。

命令：

```
aws ssm create-association-batch --entries "Name=AWS-
UpdateSSMAgent,InstanceId=i-1234567890abcdef0" "Name=AWS-
UpdateSSMAgent,InstanceId=i-9876543210abcdef0"
```

輸出：

```
{
  "Successful": [
      {
          "Name": "AWS-UpdateSSMAgent",
          "InstanceId": "i-1234567890abcdef0",
          "AssociationVersion": "1",
          "Date": 1550504725.007,
          "LastUpdateAssociationDate": 1550504725.007,
          "Status": {
              "Date": 1550504725.007,
```

```
            "Name": "Associated",
            "Message": "Associated with AWS-UpdateSSMAgent"
        },
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "DocumentVersion": "$DEFAULT",
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
        "Targets": [
            {
                "Key": "InstanceIds",
                "Values": [
                    "i-1234567890abcdef0"
                ]
            }
        ]
    },
    {
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-9876543210abcdef0",
        "AssociationVersion": "1",
        "Date": 1550504725.057,
        "LastUpdateAssociationDate": 1550504725.057,
        "Status": {
            "Date": 1550504725.057,
            "Name": "Associated",
            "Message": "Associated with AWS-UpdateSSMAgent"
        },
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "DocumentVersion": "$DEFAULT",
        "AssociationId": "9c9f7f20-5154-4fed-a83e-0123456789ab",
        "Targets": [
            {
                "Key": "InstanceIds",
                "Values": [
                    "i-9876543210abcdef0"
                ]
            }
        ]
    }
```

```
  ],
  "Failed": []
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateAssociationBatch。

create-association

以下程式碼範例顯示如何使用 create-association。

AWS CLI

範例 1：使用執行個體 ID 關聯文件

此範例使用執行個體 ID 來關聯組態文件與執行個體。

```
aws ssm create-association \
    --instance-id "i-0cb2b964d3e14fd9f" \
    --name "AWS-UpdateSSMAgent"
```

輸出：

```
{
    "AssociationDescription": {
        "Status": {
            "Date": 1487875500.33,
            "Message": "Associated with AWS-UpdateSSMAgent",
            "Name": "Associated"
        },
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-0cb2b964d3e14fd9f",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1487875500.33,
        "Date": 1487875500.33,
        "Targets": [
            {
                "Values": [
```

```
                    "i-0cb2b964d3e14fd9f"
                ],
                "Key": "InstanceIds"
            }
        ]
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager API 參考》中的 CreateAssociation。

範例 2：使用目標關聯文件

此範例使用目標來關聯組態文件與執行個體。

```
aws ssm create-association \
    --name "AWS-UpdateSSMAgent" \
    --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f"
```

輸出：

```
{
    "AssociationDescription": {
        "Status": {
            "Date": 1487875500.33,
            "Message": "Associated with AWS-UpdateSSMAgent",
            "Name": "Associated"
        },
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-0cb2b964d3e14fd9f",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1487875500.33,
        "Date": 1487875500.33,
        "Targets": [
            {
                "Values": [
                    "i-0cb2b964d3e14fd9f"
                ],
                "Key": "InstanceIds"
```

```
            }
        ]
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager API 參考》中的 CreateAssociation。

範例 3：建立僅執行一次的關聯

此範例建立在指定日期和時間只執行一次的新關聯。以過去或現在的日期建立的關聯 (在處理時，該日期已是過去的日期) 會立即執行。

```
aws ssm create-association \
    --name "AWS-UpdateSSMAgent" \
    --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f" \
    --schedule-expression "at(2020-05-14T15:55:00)"  \
    --apply-only-at-cron-interval
```

輸出：

```
{
    "AssociationDescription": {
        "Status": {
            "Date": 1487875500.33,
            "Message": "Associated with AWS-UpdateSSMAgent",
            "Name": "Associated"
        },
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-0cb2b964d3e14fd9f",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1487875500.33,
        "Date": 1487875500.33,
        "Targets": [
            {
                "Values": [
                    "i-0cb2b964d3e14fd9f"
                ],
                "Key": "InstanceIds"
```

```
            }
        ]
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager API 參考》中的 CreateAssociation，或《AWS Systems Manager 使用者指南》中的 Reference: Cron and rate expressions for Systems Manager。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateAssociation。

`create-document`

以下程式碼範例顯示如何使用 `create-document`。

AWS CLI

建立文件

以下 `create-document` 範例示範建立 Systems Manager 文件。

```
aws ssm create-document \
    --content file://exampleDocument.yml \
    --name "Example" \
    --document-type "Automation" \
    --document-format YAML
```

輸出：

```
{
    "DocumentDescription": {
        "Hash": "fc2410281f40779e694a8b95975d0f9f316da8a153daa94e3d9921102EXAMPLE",
        "HashType": "Sha256",
        "Name": "Example",
        "Owner": "29884EXAMPLE",
        "CreatedDate": 1583256349.452,
        "Status": "Creating",
        "DocumentVersion": "1",
        "Description": "Document Example",
        "Parameters": [
            {
                "Name": "AutomationAssumeRole",
```

```
                "Type": "String",
                "Description": "(Required) The ARN of the role that allows
 Automation to perform the actions on your behalf. If no role is specified, Systems
 Manager Automation uses your IAM permissions to execute this document.",
                "DefaultValue": ""
            },
            {
                "Name": "InstanceId",
                "Type": "String",
                "Description": "(Required) The ID of the Amazon EC2 instance.",
                "DefaultValue": ""
            }
        ],
        "PlatformTypes": [
            "Windows",
            "Linux"
        ],
        "DocumentType": "Automation",
        "SchemaVersion": "0.3",
        "LatestVersion": "1",
        "DefaultVersion": "1",
        "DocumentFormat": "YAML",
        "Tags": []
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Creating Systems Manager Documents。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateDocument。

`create-maintenance-window`

以下程式碼範例顯示如何使用 `create-maintenance-window`。

AWS CLI

範例 1：建立維護時段

下列 `create-maintenance-window` 範例會建立一個新的維護時段，該維護時段每五分鐘 (最長持續兩個小時，視需要) 防止新任務在維護時段執行結束後一小時內啟動、允許未關聯的目標 (尚未向維護時段註冊的執行個體)，以及透過使用其建立者打算在教學課程中使用的自訂標籤來進行表示。

```
aws ssm create-maintenance-window \
    --name "My-Tutorial-Maintenance-Window" \
    --schedule "rate(5 minutes)" \
    --duration 2 --cutoff 1 \
    --allow-unassociated-targets \
    --tags "Key=Purpose,Value=Tutorial"
```

輸出：

```
{
    "WindowId": "mw-0c50858d01EXAMPLE"
}
```

範例 2：建立只執行一次的維護時段

下列 create-maintenance-window 範例會建立只在指定的日期和時間執行一次的新維護時段。

```
aws ssm create-maintenance-window \
    --name My-One-Time-Maintenance-Window \
    --schedule "at(2020-05-14T15:55:00)" \
    --duration 5 \
    --cutoff 2 \
    --allow-unassociated-targets \
    --tags "Key=Environment,Value=Production"
```

輸出：

```
{
    "WindowId": "mw-01234567890abcdef"
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Maintenance Windows。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateMaintenanceWindow。

create-ops-item

以下程式碼範例顯示如何使用 create-ops-item。

AWS CLI

建立 OpsItems

下列 `create-ops-item` 範例在 OperationalData 中使用 /aws/resources 索引鍵來建立含 Amazon DynamoDB 相關資源的 OpsItem。

```
aws ssm create-ops-item \
    --title "EC2 instance disk full" \
    --description "Log clean up may have failed which caused the disk to be full" \
    --priority 2 \
    --source ec2 \
    --operational-data '{"/aws/resources":{"Value":"[{\"arn
\": \"arn:aws:dynamodb:us-west-2:12345678:table/OpsItems
\"}]","Type":"SearchableString"}}' \
    --notifications Arn="arn:aws:sns:us-west-2:12345678:TestUser"
```

輸出：

```
{
    "OpsItemId": "oi-1a2b3c4d5e6f"
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Creating OpsItems。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateOpsItem。

create-patch-baseline

以下程式碼範例顯示如何使用 `create-patch-baseline`。

AWS CLI

範例 1：建立具有自動核准的修補基準

下列 `create-patch-baseline` 範例示範建立 Windows Server 修補基準，該基準會在 Microsoft 發行適用於生產環境的修補程式七天之後對它們進行核准。

```
aws ssm create-patch-baseline \
    --name "Windows-Production-Baseline-AutoApproval" \
    --operating-system "WINDOWS" \
    --approval-
rules "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,Impo
```

```
{Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,UpdateRollups,CriticalUpdates]}]},Approv
 \
    --description "Baseline containing all updates approved for Windows Server
 production systems"
```

輸出：

```
{
    "BaselineId": "pb-045f10b4f3EXAMPLE"
}
```

範例 2：建立具有核准截止日期的修補基準

下列 create-patch-baseline 範例示範建立 Windows Server 修補基準，該基準會對 2020 年 7 月 7 日 (含) 之前發布的適用於生產環境的所有修補程式進行核准。

```
aws ssm create-patch-baseline \
    --name "Windows-Production-Baseline-AutoApproval" \
    --operating-system "WINDOWS" \
    --approval-
rules "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,Impo
{Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,UpdateRollups,CriticalUpdates]}]},Approv
 \
    --description "Baseline containing all updates approved for Windows Server
 production systems"
```

輸出：

```
{
    "BaselineId": "pb-045f10b4f3EXAMPLE"
}
```

範例 3：建立核准規則存放在 JSON 檔案中的修補基準

下列 create-patch-baseline 範例示範建立 Amazon Linux 2017.09 修補基準，該基準會在適用於生產環境的修補程式發布 7 天後對它們進行核准、指定修補基準的核准規則，以及指定修補程式的自訂儲存庫。

```
aws ssm create-patch-baseline \
    --cli-input-json file://my-amazon-linux-approval-rules-and-repo.json
```

my-amazon-linux-approval-rules-and-repo.json 的內容：

```
{
    "Name": "Amazon-Linux-2017.09-Production-Baseline",
    "Description": "My approval rules patch baseline for Amazon Linux 2017.09
 instances",
    "OperatingSystem": "AMAZON_LINUX",
    "Tags": [
        {
            "Key": "Environment",
            "Value": "Production"
        }
    ],
    "ApprovalRules": {
        "PatchRules": [
            {
                "ApproveAfterDays": 7,
                "EnableNonSecurity": true,
                "PatchFilterGroup": {
                    "PatchFilters": [
                        {
                            "Key": "SEVERITY",
                            "Values": [
                                "Important",
                                "Critical"
                            ]
                        },
                        {
                            "Key": "CLASSIFICATION",
                            "Values": [
                                "Security",
                                "Bugfix"
                            ]
                        },
                        {
                            "Key": "PRODUCT",
                            "Values": [
                                "AmazonLinux2017.09"
                            ]
                        }
                    ]
                }
            }
        ]
```

```
    },
    "Sources": [
        {
            "Name": "My-AL2017.09",
            "Products": [
                "AmazonLinux2017.09"
            ],
            "Configuration": "[amzn-main] \nname=amzn-main-Base
\nmirrorlist=http://repo./$awsregion./$awsdomain//$releasever/main/mirror.list //
nmirrorlist_expire=300//nmetadata_expire=300 \npriority=10 \nfailovermethod=priority
 \nfastestmirror_enabled=0 \ngpgcheck=1 \ngpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-
KEY-amazon-ga \nenabled=1 \nretries=3 \ntimeout=5\nreport_instanceid=yes"
        }
    ]
}
```

範例 4：建立將指定已核准和遭拒的修補程式的修補基準

下列 `create-patch-baseline` 範例明確指定作為預設核准規則例外狀況要核准和拒絕的修補程式。

```
aws ssm create-patch-baseline \
    --name "Amazon-Linux-2017.09-Alpha-Baseline" \
    --description "My custom approve/reject patch baseline for Amazon Linux 2017.09
 instances" \
    --operating-system "AMAZON_LINUX" \
    --approved-patches "CVE-2018-1234567,example-pkg-EE-2018*.amzn1.noarch" \
    --approved-patches-compliance-level "HIGH" \
    --approved-patches-enable-non-security \
    --tags "Key=Environment,Value=Alpha"
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Create a Custom Patch Baseline。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreatePatchBaseline。

create-resource-data-sync

以下程式碼範例顯示如何使用 `create-resource-data-sync`。

AWS CLI

建立資源資料同步

此範例會建立資源資料同步。如果命令成功，則無輸出訊息。

命令：

```
aws ssm create-resource-data-sync --sync-name "ssm-resource-data-sync" --s3-destination "BucketName=ssm-bucket,Prefix=inventory,SyncFormat=JsonSerDe,Region=us-east-1"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateResourceDataSync。

delete-activation

以下程式碼範例顯示如何使用 delete-activation。

AWS CLI

刪除受管執行個體啟用

下列 delete-activation 範例示範刪除受管執行個體啟用。

```
aws ssm delete-activation \
    --activation-id "aa673477-d926-42c1-8757-1358cEXAMPLE"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的設定混合環境的 AWS Systems Manager。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteActivation。

delete-association

以下程式碼範例顯示如何使用 delete-association。

AWS CLI

範例 1：使用關聯 ID 來刪除關聯

下列 delete-association 範例示範刪除指定關聯 ID 的關聯。如果命令成功，則無輸出訊息。

```
aws ssm delete-association \
```

```
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Editing and creating a new version of an association。

範例 2：刪除關聯

下列 delete-association 範例示範刪除執行個體與文件之間的關聯。如果命令成功，則無輸出訊息。

```
aws ssm delete-association \
    --instance-id "i-1234567890abcdef0" \
    --name "AWS-UpdateSSMAgent"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with associations in Systems Manager。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteAssociation。

delete-document

以下程式碼範例顯示如何使用 delete-document。

AWS CLI

刪除文件

下列 delete-document 範例示範刪除 Systems Manager 文件。

```
aws ssm delete-document \
    --name "Example"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Creating Systems Manager Documents。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteDocument。

delete-inventory

以下程式碼範例顯示如何使用 delete-inventory。

AWS CLI

刪除自訂庫存類型

此範例會刪除自訂庫存結構描述。

命令：

```
aws ssm delete-inventory --type-name "Custom:RackInfo" --schema-delete-
option "DeleteSchema"
```

輸出：

```
{
  "DeletionId": "d72ac9e8-1f60-4d40-b1c6-bf8c78c68c4d",
  "TypeName": "Custom:RackInfo",
  "DeletionSummary": {
      "TotalCount": 1,
      "RemainingCount": 1,
      "SummaryItems": [
          {
              "Version": "1.0",
              "Count": 1,
              "RemainingCount": 1
          }
      ]
  }
}
```

停用自訂庫存類型

此範例會停用自訂庫存結構描述。

命令：

```
aws ssm delete-inventory --type-name "Custom:RackInfo" --schema-delete-
option "DisableSchema"
```

輸出：

```
{
  "DeletionId": "6961492a-8163-44ec-aa1e-923364dd0850",
  "TypeName": "Custom:RackInformation",
  "DeletionSummary": {
      "TotalCount": 0,
      "RemainingCount": 0,
      "SummaryItems": []
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteInventory。

delete-maintenance-window

以下程式碼範例顯示如何使用 delete-maintenance-window。

AWS CLI

刪除維護時段

此 delete-maintenance-window 範例示範移除指定的維護時段。

```
aws ssm delete-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9"
```

輸出：

```
{
    "WindowId":"mw-1a2b3c4d5e6f7g8h9"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的刪除維護時段 (AWS CLI)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteMaintenanceWindow。

delete-parameter

以下程式碼範例顯示如何使用 delete-parameter。

AWS CLI

刪除參數

下列 delete-parameter 範例示範刪除指定的單個參數。

```
aws ssm delete-parameter \
    --name "MyParameter"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with Parameter Store。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteParameter。

delete-parameters

以下程式碼範例顯示如何使用 delete-parameters。

AWS CLI

刪除參數清單

下列delete-parameters範例會刪除指定的參數。

```
aws ssm delete-parameters \
    --names "MyFirstParameter" "MySecondParameter" "MyInvalidParameterName"
```

輸出：

```
{
    "DeletedParameters": [
        "MyFirstParameter",
        "MySecondParameter"
    ],
    "InvalidParameters": [
        "MyInvalidParameterName"
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with Parameter Store。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteParameters。

delete-patch-baseline

以下程式碼範例顯示如何使用 delete-patch-baseline。

AWS CLI

刪除修補基準

下列 delete-patch-baseline 範例示範刪除指定的修補基準。

```
aws ssm delete-patch-baseline \
    --baseline-id "pb-045f10b4f382baeda"
```

輸出：

```
{
    "BaselineId": "pb-045f10b4f382baeda"
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Update or Delete a Patch Baseline (Console)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeletePatchBaseline。

delete-resource-data-sync

以下程式碼範例顯示如何使用 delete-resource-data-sync。

AWS CLI

刪除資源資料同步

此範例會刪除資源資料同步。如果命令成功，則無輸出訊息。

命令：

```
aws ssm delete-resource-data-sync --sync-name "ssm-resource-data-sync"
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteResourceDataSync。

deregister-managed-instance

以下程式碼範例顯示如何使用 deregister-managed-instance。

AWS CLI

取消註冊受管執行個體

下列 deregister-managed-instance 範例示範取消註冊指定的受管執行個體。

```
aws ssm deregister-managed-instance \
    --instance-id 'mi-08ab247cdfEXAMPLE'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的在混合多雲端環境中取消註冊受管節點。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeregisterManagedInstance。

deregister-patch-baseline-for-patch-group

以下程式碼範例顯示如何使用 deregister-patch-baseline-for-patch-group。

AWS CLI

從修補基準中取消註冊修補程式群組

下列 deregister-patch-baseline-for-patch-group 範例示範從指定的修補基準中取消註冊指定的修補程式群組。

```
aws ssm deregister-patch-baseline-for-patch-group \
    --patch-group "Production" \
    --baseline-id "pb-0ca44a362fEXAMPLE"
```

輸出：

```
{
  "PatchGroup":"Production",
  "BaselineId":"pb-0ca44a362fEXAMPLE"
```

```
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Add a Patch Group to a Patch Baseline。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeregisterPatchBaselineForPatchGroup。

deregister-target-from-maintenance-window

以下程式碼範例顯示如何使用 deregister-target-from-maintenance-window。

AWS CLI

從維護時段中移除目標

下列 deregister-target-from-maintenance-window 範例示範從指定的維護時段中移除指定目標。

```
aws ssm deregister-target-from-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --window-target-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
```

輸出：

```
{
    "WindowId":"mw-ab12cd34ef56gh78",
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的更新維護時段 (AWS CLI)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeregisterTargetFromMaintenanceWindow。

deregister-task-from-maintenance-window

以下程式碼範例顯示如何使用 deregister-task-from-maintenance-window。

AWS CLI

從維護時段中移除任務

下列 deregister-task-from-maintenance-window 範例示範從指定的維護時段中移除指定任務。

```
aws ssm deregister-task-from-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --window-task-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c"
```

輸出：

```
{
    "WindowTaskId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c",
    "WindowId":"mw-ab12cd34ef56gh78"
}
```

如需詳細資訊，請參閱 Systems Manager 使用者指南中的 Systems Manager 維護 Windows 教學課程 (AWS CLI)。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeregisterTaskFromMaintenanceWindow。

describe-activations

以下程式碼範例顯示如何使用 describe-activations。

AWS CLI

描述啟用

下列describe-activations範例列出您 AWS 帳戶中啟用的詳細資訊。

```
aws ssm describe-activations
```

輸出：

```
{
    "ActivationList": [
        {
            "ActivationId": "5743558d-563b-4457-8682-d16c3EXAMPLE",
            "Description": "Example1",
            "IamRole": "HybridWebServersRole,
            "RegistrationLimit": 5,
            "RegistrationsCount": 5,
```

```
            "ExpirationDate": 1584316800.0,
            "Expired": false,
            "CreatedDate": 1581954699.792
        },
        {
            "ActivationId": "3ee0322b-f62d-40eb-b672-13ebfEXAMPLE",
            "Description": "Example2",
            "IamRole": "HybridDatabaseServersRole",
            "RegistrationLimit": 5,
            "RegistrationsCount": 5,
            "ExpirationDate": 1580515200.0,
            "Expired": true,
            "CreatedDate": 1578064132.002
        },
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Step 4: Create a Managed-Instance Activation for a Hybrid Environment。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeActivations。

describe-association-execution-targets

以下程式碼範例顯示如何使用 describe-association-execution-targets。

AWS CLI

取得關聯執行的詳細資訊

下列 describe-association-execution-targets 範例說明指定的關聯執行。

```
aws ssm describe-association-execution-targets \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --execution-id "7abb6378-a4a5-4f10-8312-0123456789ab"
```

輸出：

```
{
    "AssociationExecutionTargets": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
```

```
            "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
            "ResourceId": "i-1234567890abcdef0",
            "ResourceType": "ManagedInstance",
            "Status": "Success",
            "DetailedStatus": "Success",
            "LastExecutionDate": 1550505538.497,
            "OutputSource": {
                "OutputSourceId": "97fff367-fc5a-4299-aed8-0123456789ab",
                "OutputSourceType": "RunCommand"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Viewing association histories。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeAssociationExecutionTargets。

describe-association-executions

以下程式碼範例顯示如何使用 describe-association-executions。

AWS CLI

範例 1：取得關聯之所有執行的詳細資訊

下列 describe-association-executions 範例說明指定關聯的所有執行。

```
aws ssm describe-association-executions \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

輸出：

```
{
    "AssociationExecutions": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "474925ef-1249-45a2-b93d-0123456789ab",
            "Status": "Success",
            "DetailedStatus": "Success",
```

```
            "CreatedTime": 1550505827.119,
            "ResourceCountByStatus": "{Success=1}"
        },
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
            "Status": "Success",
            "DetailedStatus": "Success",
            "CreatedTime": 1550505536.843,
            "ResourceCountByStatus": "{Success=1}"
        },
        ...
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Viewing association histories。

範例 2：取得關聯在特定日期和時間之後的所有執行的詳細資訊

下列 `describe-association-executions` 範例說明關聯在指定日期和時間之後的所有執行。

```
aws ssm describe-association-executions \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --filters "Key=CreatedTime,Value=2019-02-18T16:00:00Z,Type=GREATER_THAN"
```

輸出：

```
{
    "AssociationExecutions": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "474925ef-1249-45a2-b93d-0123456789ab",
            "Status": "Success",
            "DetailedStatus": "Success",
            "CreatedTime": 1550505827.119,
            "ResourceCountByStatus": "{Success=1}"
        },
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
```

```
            "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
            "Status": "Success",
            "DetailedStatus": "Success",
            "CreatedTime": 1550505536.843,
            "ResourceCountByStatus": "{Success=1}"
        },
        ...
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Viewing association histories。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeAssociationExecutions。

describe-association

以下程式碼範例顯示如何使用 describe-association。

AWS CLI

範例 1：取得關聯的詳細資訊

下列 describe-association 範例說明指定關聯 ID 的關聯。

```
aws ssm describe-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

輸出：

```
{
    "AssociationDescription": {
        "Name": "AWS-GatherSoftwareInventory",
        "AssociationVersion": "1",
        "Date": 1534864780.995,
        "LastUpdateAssociationDate": 1543235759.81,
        "Overview": {
            "Status": "Success",
            "AssociationStatusAggregatedCount": {
                "Success": 2
            }
        },
        "DocumentVersion": "$DEFAULT",
```

```
        "Parameters": {
            "applications": [
                "Enabled"
            ],
            "awsComponents": [
                "Enabled"
            ],
            "customInventory": [
                "Enabled"
            ],
            "files": [
                ""
            ],
            "instanceDetailedInformation": [
                "Enabled"
            ],
            "networkConfig": [
                "Enabled"
            ],
            "services": [
                "Enabled"
            ],
            "windowsRegistry": [
                ""
            ],
            "windowsRoles": [
                "Enabled"
            ],
            "windowsUpdates": [
                "Enabled"
            ]
        },
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
        "Targets": [
            {
                "Key": "InstanceIds",
                "Values": [
                    "*"
                ]
            }
        ],
        "ScheduleExpression": "rate(24 hours)",
        "LastExecutionDate": 1550501886.0,
        "LastSuccessfulExecutionDate": 1550501886.0,
```

```
        "AssociationName": "Inventory-Association"
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Editing and creating a new version of an association。

範例 2：取得特定執行個體和文件之關聯的詳細資訊

下列 `describe-association` 範例說明執行個體與文件之間的關聯。

```
aws ssm describe-association \
    --instance-id "i-1234567890abcdef0" \
    --name "AWS-UpdateSSMAgent"
```

輸出：

```
{
    "AssociationDescription": {
        "Status": {
            "Date": 1487876122.564,
            "Message": "Associated with AWS-UpdateSSMAgent",
            "Name": "Associated"
        },
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-1234567890abcdef0",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Associated",
            "AssociationStatusAggregatedCount": {
                "Pending": 1
            }
        },
        "AssociationId": "d8617c07-2079-4c18-9847-1234567890ab",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1487876122.564,
        "Date": 1487876122.564,
        "Targets": [
            {
                "Values": [
                    "i-1234567890abcdef0"
                ],
                "Key": "InstanceIds"
```

```
            }
        ]
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Editing and creating a new version of an association。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeAssociation。

describe-automation-executions

以下程式碼範例顯示如何使用 describe-automation-executions。

AWS CLI

描述自動化執行

下列 describe-automation-executions 範例顯示有關自動化執行的詳細資訊。

```
aws ssm describe-automation-executions \
    --filters Key=ExecutionId,Values=73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

輸出：

```
{
    "AutomationExecutionMetadataList": [
        {
            "AutomationExecutionId": "73c8eef8-f4ee-4a05-820c-e354fEXAMPLE",
            "DocumentName": "AWS-StartEC2Instance",
            "DocumentVersion": "1",
            "AutomationExecutionStatus": "Success",
            "ExecutionStartTime": 1583737233.748,
            "ExecutionEndTime": 1583737234.719,
            "ExecutedBy": "arn:aws:sts::29884EXAMPLE:assumed-role/mw_service_role/
OrchestrationService",
            "LogFile": "",
            "Outputs": {},
            "Mode": "Auto",
            "Targets": [],
            "ResolvedTargets": {
                "ParameterValues": [],
```

```
                "Truncated": false
            },
            "AutomationType": "Local"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Running a Simple Automation Workflow。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeAutomationExecutions。

describe-automation-step-executions

以下程式碼範例顯示如何使用 describe-automation-step-executions。

AWS CLI

範例 1：描述自動化執行的所有步驟

下列 describe-automation-step-executions 範例顯示有關自動化執行步驟的詳細資訊。

```
aws ssm describe-automation-step-executions \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

輸出：

```
{
    "StepExecutions": [
        {
            "StepName": "startInstances",
            "Action": "aws:changeInstanceState",
            "ExecutionStartTime": 1583737234.134,
            "ExecutionEndTime": 1583737234.672,
            "StepStatus": "Success",
            "Inputs": {
                "DesiredState": "\"running\"",
                "InstanceIds": "[\"i-0cb99161f6EXAMPLE\"]"
            },
            "Outputs": {
                "InstanceStates": [
                    "running"
                ]
```

```
            },
            "StepExecutionId": "95e70479-cf20-4d80-8018-7e4e2EXAMPLE",
            "OverriddenParameters": {}
        }
    ]
}
```

範例 2：描述自動化執行的特定步驟

下列 describe-automation-step-executions 範例顯示有關自動化執行的特定步驟的詳細資訊。

```
aws ssm describe-automation-step-executions \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE \
    --filters Key=StepExecutionId,Values=95e70479-cf20-4d80-8018-7e4e2EXAMPLE
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Running an Automation Workflow Step by Step (Command Line)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeAutomationStepExecutions。

describe-available-patches

以下程式碼範例顯示如何使用 describe-available-patches。

AWS CLI

取得可用的修補程式

下列 describe-available-patches 範例擷取有關適用於 Windows Server 2019 的所有可用修補程式的詳細資訊，這些修補程式的 MSRC 嚴重性為「關鍵」。

```
aws ssm describe-available-patches \
    --
filters "Key=PRODUCT,Values=WindowsServer2019" "Key=MSRC_SEVERITY,Values=Critical"
```

輸出：

```
{
    "Patches": [
        {
            "Id": "fe6bd8c2-3752-4c8b-ab3e-1a7ed08767ba",
```

```
            "ReleaseDate": 1544047205.0,
            "Title": "2018-11 Update for Windows Server 2019 for x64-based Systems
 (KB4470788)",
            "Description": "Install this update to resolve issues in Windows. For a
 complete listing of the issues that are included in this update, see the associated
 Microsoft Knowledge Base article for more information. After you install this item,
 you may have to restart your computer.",
            "ContentUrl": "https://support.microsoft.com/en-us/kb/4470788",
            "Vendor": "Microsoft",
            "ProductFamily": "Windows",
            "Product": "WindowsServer2019",
            "Classification": "SecurityUpdates",
            "MsrcSeverity": "Critical",
            "KbNumber": "KB4470788",
            "MsrcNumber": "",
            "Language": "All"
        },
        {
            "Id": "c96115e1-5587-4115-b851-22baa46a3f11",
            "ReleaseDate": 1549994410.0,
            "Title": "2019-02 Security Update for Adobe Flash Player for Windows
 Server 2019 for x64-based Systems (KB4487038)",
            "Description": "A security issue has been identified in a Microsoft
 software product that could affect your system. You can help protect your system
 by installing this update from Microsoft. For a complete listing of the issues that
 are included in this update, see the associated Microsoft Knowledge Base article.
 After you install this update, you may have to restart your system.",
            "ContentUrl": "https://support.microsoft.com/en-us/kb/4487038",
            "Vendor": "Microsoft",
            "ProductFamily": "Windows",
            "Product": "WindowsServer2019",
            "Classification": "SecurityUpdates",
            "MsrcSeverity": "Critical",
            "KbNumber": "KB4487038",
            "MsrcNumber": "",
            "Language": "All"
        },
        ...
    ]
}
```

取得特定修補程式的詳細資訊

下列 `describe-available-patches` 範例擷取有關指定修補程式的詳細資訊。

```
aws ssm describe-available-patches \
    --filters "Key=PATCH_ID,Values=KB4480979"
```

輸出：

```
{
    "Patches": [
        {
            "Id": "680861e3-fb75-432e-818e-d72e5f2be719",
            "ReleaseDate": 1546970408.0,
            "Title": "2019-01 Security Update for Adobe Flash Player for Windows
 Server 2016 for x64-based Systems (KB4480979)",
            "Description": "A security issue has been identified in a Microsoft
 software product that could affect your system. You can help protect your system
 by installing this update from Microsoft. For a complete listing of the issues that
 are included in this update, see the associated Microsoft Knowledge Base article.
 After you install this update, you may have to restart your system.",
            "ContentUrl": "https://support.microsoft.com/en-us/kb/4480979",
            "Vendor": "Microsoft",
            "ProductFamily": "Windows",
            "Product": "WindowsServer2016",
            "Classification": "SecurityUpdates",
            "MsrcSeverity": "Critical",
            "KbNumber": "KB4480979",
            "MsrcNumber": "",
            "Language": "All"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 How Patch Manager Operations Work。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeAvailablePatches。

describe-document-permission

以下程式碼範例顯示如何使用 describe-document-permission。

AWS CLI

描述文件許可

下列 describe-document-permission 範例顯示有關公開共用之 Systems Manager 文件的許可詳細資訊。

```
aws ssm describe-document-permission \
    --name "Example" \
    --permission-type "Share"
```

輸出：

```
{
    "AccountIds": [
        "all"
    ],
    "AccountSharingInfoList": [
        {
            "AccountId": "all",
            "SharedDocumentVersion": "$DEFAULT"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Share a Systems Manager Document。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeDocumentPermission。

describe-document

以下程式碼範例顯示如何使用 describe-document。

AWS CLI

顯示文件的詳細資訊

下列describe-document範例顯示您 AWS 帳戶中 Systems Manager 文件的詳細資訊。

```
aws ssm describe-document \
    --name "Example"
```

輸出：

```
{
    "Document": {
        "Hash": "fc2410281f40779e694a8b95975d0f9f316da8a153daa94e3d9921102EXAMPLE",
        "HashType": "Sha256",
        "Name": "Example",
        "Owner": "29884EXAMPLE",
        "CreatedDate": 1583257938.266,
        "Status": "Active",
        "DocumentVersion": "1",
        "Description": "Document Example",
        "Parameters": [
            {
                "Name": "AutomationAssumeRole",
                "Type": "String",
                "Description": "(Required) The ARN of the role that allows
 Automation to perform the actions on your behalf. If no role is specified, Systems
 Manager Automation uses your IAM permissions to execute this document.",
                "DefaultValue": ""
            },
            {
                "Name": "InstanceId",
                "Type": "String",
                "Description": "(Required) The ID of the Amazon EC2 instance.",
                "DefaultValue": ""
            }
        ],
        "PlatformTypes": [
            "Windows",
            "Linux"
        ],
        "DocumentType": "Automation",
        "SchemaVersion": "0.3",
        "LatestVersion": "1",
        "DefaultVersion": "1",
        "DocumentFormat": "YAML",
        "Tags": []
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Creating Systems Manager Documents。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeDocument。

describe-effective-instance-associations

以下程式碼範例顯示如何使用 describe-effective-instance-associations。

AWS CLI

取得執行個體有效關聯的詳細資訊

下列 describe-effective-instance-associations 範例示範擷取有關執行個體有效關聯的詳細資訊。

命令：

```
aws ssm describe-effective-instance-associations --instance-id "i-1234567890abcdef0"
```

輸出：

```
{
    "Associations": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "InstanceId": "i-1234567890abcdef0",
            "Content": "{\n    \"schemaVersion\": \"1.2\",\n    \"description\":
 \"Update the Amazon SSM Agent to the latest version or specified version.\",\n
   \"parameters\": {\n        \"version\": {\n            \"default\": \"\",\n
         \"description\": \"(Optional) A specific version of the Amazon SSM Agent
 to install. If not specified, the agent will be updated to the latest version.\",
\n            \"type\": \"String\"\n        },\n        \"allowDowngrade\": {\n
            \"default\": \"false\",\n            \"description\": \"(Optional)
 Allow the Amazon SSM Agent service to be downgraded to an earlier version. If
 set to false, the service can be upgraded to newer versions only (default). If
 set to true, specify the earlier version.\",\n            \"type\": \"String\",
\n            \"allowedValues\": [\n                \"true\",\n
 \"false\"\n            ]\n        }\n    },\n    \"runtimeConfig\": {\n
 \"aws:updateSsmAgent\": {\n            \"properties\": [\n                {\n
              \"agentName\": \"amazon-ssm-agent\",\n                \"source\":
 \"https://s3.{Region}.amazonaws.com/amazon-ssm-{Region}/ssm-agent-manifest.json\",
\n                \"allowDowngrade\": \"{{ allowDowngrade }}\",\n
 \"targetVersion\": \"{{ version }}\"\n                }\n            ]\n        }\n
    }\n}\n",
            "AssociationVersion": "1"
        }
    ]
```

```
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeEffectiveInstanceAssociations。

describe-effective-patches-for-patch-baseline

以下程式碼範例顯示如何使用 describe-effective-patches-for-patch-baseline。

AWS CLI

範例 1：取得由自訂修補基準定義的所有修補程式

下列describe-effective-patches-for-patch-baseline範例會傳回目前 AWS 帳戶中自訂修補程式基準定義的修補程式。請注意，對於自訂基準，--baseline-id 只需要 ID。

```
aws ssm describe-effective-patches-for-patch-baseline \
    --baseline-id "pb-08b654cf9b9681f04"
```

輸出：

```
{
    "EffectivePatches": [
        {
            "Patch": {
                "Id": "fe6bd8c2-3752-4c8b-ab3e-1a7ed08767ba",
                "ReleaseDate": 1544047205.0,
                "Title": "2018-11 Update for Windows Server 2019 for x64-based
 Systems (KB4470788)",
                "Description": "Install this update to resolve issues in Windows.
 For a complete listing of the issues that are included in this update, see the
 associated Microsoft Knowledge Base article for more information. After you install
 this item, you may have to restart your computer.",
                "ContentUrl": "https://support.microsoft.com/en-us/kb/4470788",
                "Vendor": "Microsoft",
                "ProductFamily": "Windows",
                "Product": "WindowsServer2019",
                "Classification": "SecurityUpdates",
                "MsrcSeverity": "Critical",
                "KbNumber": "KB4470788",
                "MsrcNumber": "",
                "Language": "All"
```

```
            },
            "PatchStatus": {
                "DeploymentStatus": "APPROVED",
                "ComplianceLevel": "CRITICAL",
                "ApprovalDate": 1544047205.0
            }
        },
        {
            "Patch": {
                "Id": "915a6b1a-f556-4d83-8f50-b2e75a9a7e58",
                "ReleaseDate": 1549994400.0,
                "Title": "2019-02 Cumulative Update for .NET Framework 3.5 and 4.7.2
 for Windows Server 2019 for x64 (KB4483452)",
                "Description": "A security issue has been identified in a Microsoft
 software product that could affect your system. You can help protect your system by
 installing this update from Microsoft. For a complete listing of the issues that
 are included in this update, see the associated Microsoft Knowledge Base article.
 After you install this update, you may have to restart your system.",
                "ContentUrl": "https://support.microsoft.com/en-us/kb/4483452",
                "Vendor": "Microsoft",
                "ProductFamily": "Windows",
                "Product": "WindowsServer2019",
                "Classification": "SecurityUpdates",
                "MsrcSeverity": "Important",
                "KbNumber": "KB4483452",
                "MsrcNumber": "",
                "Language": "All"
            },
            "PatchStatus": {
                "DeploymentStatus": "APPROVED",
                "ComplianceLevel": "CRITICAL",
                "ApprovalDate": 1549994400.0
            }
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

範例 2：取得受 AWS 管修補程式基準定義的所有修補程式

下列describe-effective-patches-for-patch-baseline範例會傳回 AWS 受管修補程式基準定義的修補程式。請注意，對於 AWS 受管基準， 需要完整的基準 ARN --baseline-id

```
aws ssm describe-effective-patches-for-patch-baseline \
    --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-020d361a05defe4ed"
```

如需範例輸出，請參閱範例 1。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 How Security Patches Are Selected。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeEffectivePatchesForPatchBaseline。

describe-instance-associations-status

以下程式碼範例顯示如何使用 describe-instance-associations-status。

AWS CLI

描述執行個體關聯的狀態

此範例顯示執行個體關聯的詳細資訊。

命令：

```
aws ssm describe-instance-associations-status --instance-id "i-1234567890abcdef0"
```

輸出：

```
{
  "InstanceAssociationStatusInfos": [
      {
          "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
          "Name": "AWS-GatherSoftwareInventory",
          "DocumentVersion": "1",
          "AssociationVersion": "1",
          "InstanceId": "i-1234567890abcdef0",
          "ExecutionDate": 1550501886.0,
          "Status": "Success",
          "ExecutionSummary": "1 out of 1 plugin processed, 1 success, 0 failed, 0
 timedout, 0 skipped. ",
          "AssociationName": "Inventory-Association"
      },
```

```
        {
            "AssociationId": "5c5a31f6-6dae-46f9-944c-0123456789ab",
            "Name": "AWS-UpdateSSMAgent",
            "DocumentVersion": "1",
            "AssociationVersion": "1",
            "InstanceId": "i-1234567890abcdef0",
            "ExecutionDate": 1550505828.548,
            "Status": "Success",
            "DetailedStatus": "Success",
            "AssociationName": "UpdateSSMAgent"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeInstanceAssociationsStatus。

describe-instance-information

以下程式碼範例顯示如何使用 describe-instance-information。

AWS CLI

範例 1：描述受管執行個體資訊

下列 describe-instance-information 範例擷取每個受管執行個體的詳細資訊。

```
aws ssm describe-instance-information
```

範例 2：描述有關特定受管執行個體的資訊

下列 describe-instance-information 範例顯示受管執行個體的詳細資訊i-028ea792daEXAMPLE。

```
aws ssm describe-instance-information \
    --filters "Key=InstanceIds,Values=i-028ea792daEXAMPLE"
```

範例 3：描述有關具有特定標籤索引鍵之受管執行個體的資訊

下列 describe-instance-information 範例顯示具有標籤索引鍵 DEV 之受管執行個體的詳細資訊。

```
aws ssm describe-instance-information \
```

```
    --filters "Key=tag-key,Values=DEV"
```

輸出：

```
{
    "InstanceInformationList": [
        {
            "InstanceId": "i-028ea792daEXAMPLE",
            "PingStatus": "Online",
            "LastPingDateTime": 1582221233.421,
            "AgentVersion": "2.3.842.0",
            "IsLatestVersion": true,
            "PlatformType": "Linux",
            "PlatformName": "SLES",
            "PlatformVersion": "15.1",
            "ResourceType": "EC2Instance",
            "IPAddress": "192.0.2.0",
            "ComputerName": "ip-198.51.100.0.us-east-2.compute.internal",
            "AssociationStatus": "Success",
            "LastAssociationExecutionDate": 1582220806.0,
            "LastSuccessfulAssociationExecutionDate": 1582220806.0,
            "AssociationOverview": {
                "DetailedStatus": "Success",
                "InstanceAssociationStatusAggregatedCount": {
                    "Success": 2
                }
            }
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Managed Instances。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeInstanceInformation。

describe-instance-patch-states-for-patch-group

以下程式碼範例顯示如何使用 describe-instance-patch-states-for-patch-group。

AWS CLI

範例 1：取得修補程式群組的執行個體狀態

下列 describe-instance-patch-states-for-patch-group 範例示範擷取有關指定修補程式群組的每個執行個體之修補程式摘要狀態的詳細資訊。

```
aws ssm describe-instance-patch-states-for-patch-group \
    --patch-group "Production"
```

輸出：

```
{
    "InstancePatchStates": [
        {
            "InstanceId": "i-02573cafcfEXAMPLE",
            "PatchGroup": "Production",
            "BaselineId": "pb-0c10e65780EXAMPLE",
            "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
            "OwnerInformation": "",
            "InstalledCount": 32,
            "InstalledOtherCount": 1,
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
            "MissingCount": 2,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": 2671,
            "NotApplicableCount": 400,
            "OperationStartTime": "2021-08-04T11:03:50.590000-07:00",
            "OperationEndTime": "2021-08-04T11:04:21.555000-07:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot",
            "CriticalNonCompliantCount": 0,
            "SecurityNonCompliantCount": 1,
            "OtherNonCompliantCount": 0
        },
        {
            "InstanceId": "i-0471e04240EXAMPLE",
            "PatchGroup": "Production",
            "BaselineId": "pb-09ca3fb51fEXAMPLE",
            "SnapshotId": "05d8ffb0-1bbe-4812-ba2d-d9b7bEXAMPLE",
            "OwnerInformation": "",
            "InstalledCount": 32,
            "InstalledOtherCount": 1,
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
            "MissingCount": 2,
```

```
            "FailedCount": 0,
            "UnreportedNotApplicableCount": 2671,
            "NotApplicableCount": 400,
            "OperationStartTime": "2021-08-04T22:06:20.340000-07:00",
            "OperationEndTime": "2021-08-04T22:07:11.220000-07:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot",
            "CriticalNonCompliantCount": 0,
            "SecurityNonCompliantCount": 1,
            "OtherNonCompliantCount": 0
        }
    ]
}
```

範例 2：取得缺少超過五個修補程式之修補程式群組的執行個體狀態

下列 describe-instance-patch-states-for-patch-group 範例針對缺少超過五個修補程式的執行個體，擷取有關指定修補程式群組之修補程式摘要狀態的詳細資訊。

```
aws ssm describe-instance-patch-states-for-patch-group \
    --filters Key=MissingCount,Type=GreaterThan,Values=5 \
    --patch-group "Production"
```

輸出：

```
{
    "InstancePatchStates": [
        {
            "InstanceId": "i-02573cafcfEXAMPLE",
            "PatchGroup": "Production",
            "BaselineId": "pb-0c10e65780EXAMPLE",
            "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
            "OwnerInformation": "",
            "InstalledCount": 46,
            "InstalledOtherCount": 4,
            "InstalledPendingRebootCount": 1,
            "InstalledRejectedCount": 1,
            "MissingCount": 7,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": 232,
            "NotApplicableCount": 654,
            "OperationStartTime": "2021-08-04T11:03:50.590000-07:00",
            "OperationEndTime": "2021-08-04T11:04:21.555000-07:00",
```

```
            "Operation": "Scan",
            "RebootOption": "NoReboot",
            "CriticalNonCompliantCount": 0,
            "SecurityNonCompliantCount": 1,
            "OtherNonCompliantCount": 1
        }
    ]
}
```

範例 3：取得修補程式群組的執行個體狀態，其中需要重新啟動的執行個體少於十個

下列 describe-instance-patch-states-for-patch-group 範例示範擷取有關執行個體的指定修補程式群組之修補程式摘要狀態的詳細資訊，其中需要重新啟動的執行個體少於十個。

```
aws ssm describe-instance-patch-states-for-patch-group \
    --filters Key=InstalledPendingRebootCount,Type=LessThan,Values=10 \
    --patch-group "Production"
```

輸出：

```
{
    "InstancePatchStates": [
        {
            "InstanceId": "i-02573cafcfEXAMPLE",
            "BaselineId": "pb-0c10e65780EXAMPLE",
            "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
            "PatchGroup": "Production",
            "OwnerInformation": "",
            "InstalledCount": 32,
            "InstalledOtherCount": 1,
            "InstalledPendingRebootCount": 4,
            "InstalledRejectedCount": 0,
            "MissingCount": 2,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": 846,
            "NotApplicableCount": 212,
            "OperationStartTime": "2021-08-046T11:03:50.590000-07:00",
            "OperationEndTime": "2021-08-06T11:04:21.555000-07:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot",
            "CriticalNonCompliantCount": 0,
            "SecurityNonCompliantCount": 1,
            "OtherNonCompliantCount": 0
```

```
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Understanding patch compliance state values。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeInstancePatchStatesForPatchGroup。

describe-instance-patch-states

以下程式碼範例顯示如何使用 describe-instance-patch-states。

AWS CLI

取得執行個體的修補程式摘要狀態

此 describe-instance-patch-states 範例示範取得執行個體的修補程式摘要狀態。

```
aws ssm describe-instance-patch-states \
    --instance-ids "i-1234567890abcdef0"
```

輸出：

```
{
    "InstancePatchStates": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "PatchGroup": "my-patch-group",
            "BaselineId": "pb-0713accee01234567",
            "SnapshotId": "521c3536-930c-4aa9-950e-01234567abcd",
            "CriticalNonCompliantCount": 2,
            "SecurityNonCompliantCount": 2,
            "OtherNonCompliantCount": 1,
            "InstalledCount": 123,
            "InstalledOtherCount": 334,
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
            "MissingCount": 1,
            "FailedCount": 2,
            "UnreportedNotApplicableCount": 11,
```

```
            "NotApplicableCount": 2063,
            "OperationStartTime": "2021-05-03T11:00:56-07:00",
            "OperationEndTime": "2021-05-03T11:01:09-07:00",
            "Operation": "Scan",
            "LastNoRebootInstallOperationTime": "2020-06-14T12:17:41-07:00",
            "RebootOption": "RebootIfNeeded"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS 使用者指南》中的 About Patch Compliance。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeInstancePatchStates。

describe-instance-patches

以下程式碼範例顯示如何使用 describe-instance-patches。

AWS CLI

範例 1：取得執行個體的修補程式狀態詳細資訊

下列 describe-instance-patches 範例示範擷取有關指定執行個體修補程式的詳細資訊。

```
aws ssm describe-instance-patches \
    --instance-id "i-1234567890abcdef0"
```

輸出：

```
{
    "Patches": [
        {
            "Title": "2019-01 Security Update for Adobe Flash Player for Windows
 Server 2016 for x64-based Systems (KB4480979)",
            "KBId": "KB4480979",
            "Classification": "SecurityUpdates",
            "Severity": "Critical",
            "State": "Installed",
            "InstalledTime": "2019-01-09T00:00:00+00:00"
        },
        {
            "Title": "",
```

```
            "KBId": "KB4481031",
            "Classification": "",
            "Severity": "",
            "State": "InstalledOther",
            "InstalledTime": "2019-02-08T00:00:00+00:00"
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

範例 2：取得狀態為「缺少」的執行個體修補程式清單

下列 describe-instance-patches 範例示範擷取有關指定執行個體處於「缺少」狀態的修補程式資訊。

```
aws ssm describe-instance-patches \
    --instance-id "i-1234567890abcdef0" \
    --filters Key=State,Values=Missing
```

輸出：

```
{
    "Patches": [
        {
            "Title": "Windows Malicious Software Removal Tool x64 - February 2019
 (KB890830)",
            "KBId": "KB890830",
            "Classification": "UpdateRollups",
            "Severity": "Unspecified",
            "State": "Missing",
            "InstalledTime": "1970-01-01T00:00:00+00:00"
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 About Patch Compliance States。

範例 3：取得執行個體自指定 InstalledTime 以來安裝的修補程式清單

下列 describe-instance-patches 範例結合使用 --filters 和 --query，擷取有關指定執行個體自指定時間以來所安裝修補程式的資訊。

```
aws ssm describe-instance-patches \
    --instance-id "i-1234567890abcdef0" \
    --filters Key=State,Values=Installed \
    --query "Patches[?InstalledTime >= `2023-01-01T16:00:00`]"
```

輸出：

```
{
    "Patches": [
        {
            "Title": "2023-03 Cumulative Update for Windows Server 2019 (1809) for
 x64-based Systems (KB5023702)",
            "KBId": "KB5023702",
            "Classification": "SecurityUpdates",
            "Severity": "Critical",
            "State": "Installed",
            "InstalledTime": "2023-03-16T11:00:00+00:00"
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeInstancePatches。

describe-inventory-deletions

以下程式碼範例顯示如何使用 describe-inventory-deletions。

AWS CLI

取得庫存刪除

此範例會擷取庫存刪除操作的詳細資訊。

命令：

```
aws ssm describe-inventory-deletions
```

輸出：

```
{
  "InventoryDeletions": [
      {
          "DeletionId": "6961492a-8163-44ec-aa1e-01234567850",
          "TypeName": "Custom:RackInformation",
          "DeletionStartTime": 1550254911.0,
          "LastStatus": "InProgress",
          "LastStatusMessage": "The Delete is in progress",
          "DeletionSummary": {
              "TotalCount": 0,
              "RemainingCount": 0,
              "SummaryItems": []
          },
          "LastStatusUpdateTime": 1550254911.0
      },
      {
          "DeletionId": "d72ac9e8-1f60-4d40-b1c6-987654321c4d",
          "TypeName": "Custom:RackInfo",
          "DeletionStartTime": 1550254859.0,
          "LastStatus": "InProgress",
          "LastStatusMessage": "The Delete is in progress",
          "DeletionSummary": {
              "TotalCount": 1,
              "RemainingCount": 1,
              "SummaryItems": [
                  {
                      "Version": "1.0",
                      "Count": 1,
                      "RemainingCount": 1
                  }
              ]
          },
          "LastStatusUpdateTime": 1550254859.0
      }
  ]
}
```

取得特定庫存刪除的詳細資訊

此範例會擷取特定庫存刪除操作的詳細資訊。

命令：

```
aws ssm describe-inventory-deletions --deletion-id "d72ac9e8-1f60-4d40-
b1c6-987654321c4d"
```

輸出：

```
{
  "InventoryDeletions": [
      {
          "DeletionId": "d72ac9e8-1f60-4d40-b1c6-987654321c4d",
          "TypeName": "Custom:RackInfo",
          "DeletionStartTime": 1550254859.0,
          "LastStatus": "InProgress",
          "LastStatusMessage": "The Delete is in progress",
          "DeletionSummary": {
              "TotalCount": 1,
              "RemainingCount": 1,
              "SummaryItems": [
                  {
                      "Version": "1.0",
                      "Count": 1,
                      "RemainingCount": 1
                  }
              ]
          },
          "LastStatusUpdateTime": 1550254859.0
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeInventoryDeletions。

describe-maintenance-window-execution-task-invocations

以下程式碼範例顯示如何使用 describe-maintenance-window-execution-task-invocations。

AWS CLI

取得針對維護時段任務執行所進行的特定任務調用

下列 describe-maintenance-window-execution-task-invocations 範例列出了作為指定維護時段執行的一部分執行之指定任務的調用。

```
aws ssm describe-maintenance-window-execution-task-invocations \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2a638355" \
    --task-id "ac0c6ae1-daa3-4a89-832e-d384503b6586"
```

輸出：

```
{
    "WindowExecutionTaskInvocationIdentities": [
        {
            "Status": "SUCCESS",
            "Parameters": "{\"documentName\":\"AWS-RunShellScript\",\"instanceIds\":
[\"i-0000293ffd8c57862\"],\"parameters\":{\"commands\":[\"df\"]},\"maxConcurrency\":
\"1\",\"maxErrors\":\"1\"}",
            "InvocationId": "e274b6e1-fe56-4e32-bd2a-8073c6381d8b",
            "StartTime": 1487692834.723,
            "EndTime": 1487692834.871,
            "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2a638355",
            "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d384503b6586"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的檢視任務和任務執行 (AWS CLI) 的相關資訊。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeMaintenanceWindowExecutionTaskInvocations。

describe-maintenance-window-execution-tasks

以下程式碼範例顯示如何使用 describe-maintenance-window-execution-tasks。

AWS CLI

列出與維護時段執行關聯的所有任務

下列 ssm describe-maintenance-window-execution-tasks 範例列出了與指定維護時段執行關聯的任務。

```
aws ssm describe-maintenance-window-execution-tasks \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

輸出：

```
{
    "WindowExecutionTaskIdentities": [
        {
            "Status": "SUCCESS",
            "TaskArn": "AWS-RunShellScript",
            "StartTime": 1487692834.684,
            "TaskType": "RUN_COMMAND",
            "EndTime": 1487692835.005,
            "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
            "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的檢視任務和任務執行 (AWS CLI) 的相關資訊。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeMaintenanceWindowExecutionTasks。

describe-maintenance-window-executions

以下程式碼範例顯示如何使用 describe-maintenance-window-executions。

AWS CLI

範例 1：列出維護時段的所有執行

下列 describe-maintenance-window-executions 範例列出了指定維護時段的所有執行。

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE"
```

輸出：

```
{
    "WindowExecutions": [
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
```

```
            "WindowExecutionId": "6027b513-64fe-4cf0-be7d-1191aEXAMPLE",
            "Status": "IN_PROGRESS",
            "StartTime": "2021-08-04T11:00:00.000000-07:00"

        },
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowExecutionId": "ff75b750-4834-4377-8f61-b3cadEXAMPLE",
            "Status": "SUCCESS",
            "StartTime": "2021-08-03T11:00:00.000000-07:00",
            "EndTime": "2021-08-03T11:37:21.450000-07:00"
        },
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowExecutionId": "9fac7dd9-ff21-42a5-96ad-bbc4bEXAMPLE",
            "Status": "FAILED",
            "StatusDetails": "One or more tasks in the orchestration failed.",
            "StartTime": "2021-08-02T11:00:00.000000-07:00",
            "EndTime": "2021-08-02T11:22:36.190000-07:00"
        }
    ]
}
```

範例 2：列出指定日期之前維護時段的所有執行

下列 describe-maintenance-window-executions 範例列出了指定日期之前指定維護時段的所有執行。

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=ExecutedBefore,Values=2021-08-03T00:00:00Z"
```

輸出：

```
{
    "WindowExecutions": [
        {
        "WindowId": "mw-ab12cd34eEXAMPLE",
        "WindowExecutionId": "9fac7dd9-ff21-42a5-96ad-bbc4bEXAMPLE",
        "Status": "FAILED",
        "StatusDetails": "One or more tasks in the orchestration failed.",
        "StartTime": "2021-08-02T11:00:00.000000-07:00",
        "EndTime": "2021-08-02T11:22:36.190000-07:00"
```

```
    }
    ]
}
```

範例 3：列出指定日期後維護時段的所有執行

下列 describe-maintenance-window-executions 範例列出了指定日期之後指定維護時段的所有執行。

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=ExecutedAfter,Values=2021-08-04T00:00:00Z"
```

輸出：

```
{
    "WindowExecutions": [
        {
        "WindowId": "mw-ab12cd34eEXAMPLE",
        "WindowExecutionId": "6027b513-64fe-4cf0-be7d-1191aEXAMPLE",
        "Status": "IN_PROGRESS",
        "StartTime": "2021-08-04T11:00:00.000000-07:00"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的檢視任務和任務執行 (AWS CLI) 的相關資訊。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeMaintenanceWindowExecutions。

describe-maintenance-window-schedule

以下程式碼範例顯示如何使用 describe-maintenance-window-schedule。

AWS CLI

範例 1：列出維護時段即將進行的執行

下列describe-maintenance-window-schedule範例列出指定維護時段的所有即將執行。

```
aws ssm describe-maintenance-window-schedule \
    --window-id mw-ab12cd34eEXAMPLE
```

輸出：

```
{
    "ScheduledWindowExecutions": [
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2020-02-19T16:00Z"
        },
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2020-02-26T16:00Z"
        },
        ...
    ]
}
```

範例 2：列出指定日期之前維護時段的所有即將執行

下列`describe-maintenance-window-schedule`範例列出指定維護時段在指定日期之前發生的所有即將執行。

```
aws ssm describe-maintenance-window-schedule \
    --window-id mw-0ecb1226dd7b2e9a6 \
    --filters "Key=ScheduledBefore,Values=2020-02-15T06:00:00Z"
```

範例 3：列出指定日期之後維護時段的所有即將執行

下列`describe-maintenance-window-schedule`範例列出指定維護時段在指定日期之後發生的所有即將執行。

```
aws ssm describe-maintenance-window-schedule \
    --window-id mw-0ecb1226dd7b2e9a6 \
    --filters "Key=ScheduledAfter,Values=2020-02-15T06:00:00Z"
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的檢視維護 Windows (AWS CLI) 的相關資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeMaintenanceWindowSchedule。

describe-maintenance-window-targets

以下程式碼範例顯示如何使用 describe-maintenance-window-targets。

AWS CLI

範例 1：列出維護時段的所有目標

下列 describe-maintenance-window-targets 範例列出了維護時段的所有目標。

```
aws ssm describe-maintenance-window-targets \
    --window-id "mw-06cf17cbefEXAMPLE"
```

輸出：

```
{
    "Targets": [
        {
            "ResourceType": "INSTANCE",
            "OwnerInformation": "Single instance",
            "WindowId": "mw-06cf17cbefEXAMPLE",
            "Targets": [
                {
                    "Values": [
                        "i-0000293ffdEXAMPLE"
                    ],
                    "Key": "InstanceIds"
                }
            ],
            "WindowTargetId": "350d44e6-28cc-44e2-951f-4b2c9EXAMPLE"
        },
        {
            "ResourceType": "INSTANCE",
            "OwnerInformation": "Two instances in a list",
            "WindowId": "mw-06cf17cbefEXAMPLE",
            "Targets": [
                {
                    "Values": [
                        "i-0000293ffdEXAMPLE",
                        "i-0cb2b964d3EXAMPLE"
```

```
                    ],
                    "Key": "InstanceIds"
                }
            ],
            "WindowTargetId": "e078a987-2866-47be-bedd-d9cf4EXAMPLE"
        }
    ]
}
```

範例 2：列出符合特定擁有者資訊值的維護時段目標

此 describe-maintenance-window-targets 範例列出了具有特定值的維護時段的所有目標。

```
aws ssm describe-maintenance-window-targets \
    --window-id "mw-0ecb1226ddEXAMPLE" \
    --filters "Key=OwnerInformation,Values=CostCenter1"
```

輸出：

```
{
    "Targets": [
        {
            "WindowId": "mw-0ecb1226ddEXAMPLE",
            "WindowTargetId": "da89dcc3-7f9c-481d-ba2b-edcb7d0057f9",
            "ResourceType": "INSTANCE",
            "Targets": [
                {
                    "Key": "tag:Environment",
                    "Values": [
                        "Prod"
                    ]
                }
            ],
            "OwnerInformation": "CostCenter1",
            "Name": "ProdTarget1"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的檢視維護 Windows (AWS CLI) 的相關資訊。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeMaintenanceWindowTargets。

describe-maintenance-window-tasks

以下程式碼範例顯示如何使用 describe-maintenance-window-tasks。

AWS CLI

範例 1：列出維護時段的所有任務

下列 describe-maintenance-window-tasks 範例列出了指定維護時段的所有任務。

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-06cf17cbefEXAMPLE"
```

輸出：

```
{
    "Tasks": [
        {
            "WindowId": "mw-06cf17cbefEXAMPLE",
            "WindowTaskId": "018b31c3-2d77-4b9e-bd48-c91edEXAMPLE",
            "TaskArn": "AWS-RestartEC2Instance",
            "TaskParameters": {},
            "Type": "AUTOMATION",
            "Description": "Restarting EC2 Instance for maintenance",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "My-Automation-Example-Task",
            "Priority": 0,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ]
        },
        {
```

```
            "WindowId": "mw-06cf17cbefEXAMPLE",
            "WindowTaskId": "1943dee0-0a17-4978-9bf4-3cc2fEXAMPLE",
            "TaskArn": "AWS-DisableS3BucketPublicReadWrite",
            "TaskParameters": {},
            "Type": "AUTOMATION",
            "Description": "Automation task to disable read/write access on public
 S3 buckets",
            "MaxConcurrency": "10",
            "MaxErrors": "5",
            "Name": "My-Disable-S3-Public-Read-Write-Access-Automation-Task",
            "Priority": 0,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ]
        }
    ]
}
```

範例 2：列出叫用 AWS-RunPowerShellScript 命令文件的維護時段的所有任務

下列 `describe-maintenance-window-tasks` 範例列出了調用 `AWS-RunPowerShellScript` 命令文件之指定維護時段的所有任務。

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=TaskArn,Values=AWS-RunPowerShellScript"
```

輸出：

```
{
    "Tasks": [
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",
            "TaskArn": "AWS-RunPowerShellScript",
            "Type": "RUN_COMMAND",
```

```
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 1,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "MyTask"
        }
    ]
}
```

範例 3：列出優先順序為 3 的維護時段的所有任務

下列 describe-maintenance-window-tasks 範例列出了 Priority 為 3 的指定維護時段的所有任務。

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=Priority,Values=3"
```

輸出：

```
{
    "Tasks": [
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",
            "TaskArn": "AWS-RunPowerShellScript",
            "Type": "RUN_COMMAND",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
```

```
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 3,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "MyRunCommandTask"
        },
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowTaskId": "ee45feff-ad65-4a6c-b478-5cab8EXAMPLE",
            "TaskArn": "AWS-RestartEC2Instance",
            "Type": "AUTOMATION",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 3,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "MaxConcurrency": "10",
            "MaxErrors": "5",
            "Name": "My-Automation-Task",
            "Description": "A description for my Automation task"
        }
    ]
}
```

範例 4：列出優先順序為 1 且使用 Run Command 的維護時段的所有任務

此 `describe-maintenance-window-tasks` 範例列出了 `Priority` 為 1 且使用 Run Command 的指定維護時段的所有任務。

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-ab12cd34eEXAMPLE" \
```

```
    --filters "Key=Priority,Values=1" "Key=TaskType,Values=RUN_COMMAND"
```

輸出：

```
{
    "Tasks": [
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",
            "TaskArn": "AWS-RunPowerShellScript",
            "Type": "RUN_COMMAND",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 1,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "MyRunCommandTask"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的檢視維護時段 (AWS CLI) 的相關資訊。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeMaintenanceWindowTasks。

describe-maintenance-windows-for-target

以下程式碼範例顯示如何使用 describe-maintenance-windows-for-target。

AWS CLI

列出與特定執行個體相關聯的所有維護時段

下列describe-maintenance-windows-for-target範例列出具有與指定執行個體相關聯之目標或任務的維護時段。

```
aws ssm describe-maintenance-windows-for-target \
    --targets Key=InstanceIds,Values=i-1234567890EXAMPLE \
    --resource-type INSTANCE
```

輸出：

```
{
    "WindowIdentities": [
        {
            "WindowId": "mw-0c5ed765acEXAMPLE",
            "Name": "My-First-Maintenance-Window"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的檢視維護 Windows (AWS CLI) 的相關資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeMaintenanceWindowsForTarget。

describe-maintenance-windows

以下程式碼範例顯示如何使用 describe-maintenance-windows。

AWS CLI

範例 1：列出所有維護時段

下列describe-maintenance-windows範例列出目前區域中 AWS 您帳戶中的所有維護時段。

```
aws ssm describe-maintenance-windows
```

輸出：

```
{
    "WindowIdentities": [
        {
```

```
            "WindowId": "mw-0ecb1226ddEXAMPLE",
            "Name": "MyMaintenanceWindow-1",
            "Enabled": true,
            "Duration": 2,
            "Cutoff": 1,
            "Schedule": "rate(180 minutes)",
            "NextExecutionTime": "2020-02-12T23:19:20.596Z"
        },
        {
            "WindowId": "mw-03eb9db428EXAMPLE",
            "Name": "MyMaintenanceWindow-2",
            "Enabled": true,
            "Duration": 3,
            "Cutoff": 1,
            "Schedule": "rate(7 days)",
            "NextExecutionTime": "2020-02-17T23:22:00.956Z"
        },
    ]
}
```

列出所有已啟用的維護時段

下列 describe-maintenance-windows 範例列出了所有已啟用的維護時段。

```
aws ssm describe-maintenance-windows \
    --filters "Key=Enabled,Values=true"
```

範例 3：列出符合特定名稱的維護時段

此 describe-maintenance-windows 範例列出了具有指定名稱的所有維護時段。

```
aws ssm describe-maintenance-windows \
    --filters "Key=Name,Values=MyMaintenanceWindow"
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的檢視維護 Windows (AWS CLI) 的相關資訊。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeMaintenanceWindows。

describe-ops-items

以下程式碼範例顯示如何使用 describe-ops-items。

AWS CLI

列出一組 OpsItems

下列describe-ops-items範例顯示您 AWS 帳戶中所有開啟 OpsItems 的清單。

```
aws ssm describe-ops-items \
    --ops-item-filters "Key=Status,Values=Open,Operator=Equal"
```

輸出：

```
{
    "OpsItemSummaries": [
        {
            "CreatedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-Role/
fbf77cbe264a33509569f23e4EXAMPLE",
            "CreatedTime": "2020-03-14T17:02:46.375000-07:00",
            "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-
Role/fbf77cbe264a33509569f23e4EXAMPLE",
            "LastModifiedTime": "2020-03-14T17:02:46.375000-07:00",
            "Source": "SSM",
            "Status": "Open",
            "OpsItemId": "oi-7cfc5EXAMPLE",
            "Title": "SSM Maintenance Window execution failed",
            "OperationalData": {
                "/aws/dedup": {
                    "Value": "{\"dedupString\":\"SSMOpsItems-SSM-maintenance-window-
execution-failed\"}",
                    "Type": "SearchableString"
                },
                "/aws/resources": {
                    "Value": "[{\"arn\":\"arn:aws:ssm:us-
east-2:111222333444:maintenancewindow/mw-034093d322EXAMPLE\"}]",
                    "Type": "SearchableString"
                }
            },
            "Category": "Availability",
            "Severity": "3"
        },
        {
            "CreatedBy": "arn:aws:sts::1112223233444:assumed-role/OpsItem-CWE-Role/
fbf77cbe264a33509569f23e4EXAMPLE",
            "CreatedTime": "2020-02-26T11:43:15.426000-08:00",
```

```
            "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-
Role/fbf77cbe264a33509569f23e4EXAMPLE",
            "LastModifiedTime": "2020-02-26T11:43:15.426000-08:00",
            "Source": "EC2",
            "Status": "Open",
            "OpsItemId": "oi-6f966EXAMPLE",
            "Title": "EC2 instance stopped",
            "OperationalData": {
                "/aws/automations": {
                    "Value": "[ { \"automationType\": \"AWS:SSM:Automation\",
 \"automationId\": \"AWS-RestartEC2Instance\" } ]",
                    "Type": "SearchableString"
                },
                "/aws/dedup": {
                    "Value": "{\"dedupString\":\"SSMOpsItems-EC2-instance-stopped
\"}",
                    "Type": "SearchableString"
                },
                "/aws/resources": {
                    "Value": "[{\"arn\":\"arn:aws:ec2:us-
east-2:111222333444:instance/i-0beccfbc02EXAMPLE\"}]",
                    "Type": "SearchableString"
                }
            },
            "Category": "Availability",
            "Severity": "3"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with OpsItems。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeOpsItems。

describe-parameters

以下程式碼範例顯示如何使用 describe-parameters。

AWS CLI

範例 1：列出所有參數

下列describe-parameters範例列出目前 AWS 帳戶和區域中的所有參數。

```
aws ssm describe-parameters
```

輸出：

```
{
    "Parameters": [
        {
            "Name": "MySecureStringParameter",
            "Type": "SecureString",
            "KeyId": "alias/aws/ssm",
            "LastModifiedDate": 1582155479.205,
            "LastModifiedUser": "arn:aws:sts::111222333444:assumed-role/Admin/
Richard-Roe-Managed",
            "Description": "This is a SecureString parameter",
            "Version": 2,
            "Tier": "Advanced",
            "Policies": [
                {
                    "PolicyText": "{\"Type\":\"Expiration\",\"Version\":\"1.0\",
\"Attributes\":{\"Timestamp\":\"2020-07-07T22:30:00Z\"}}",
                    "PolicyType": "Expiration",
                    "PolicyStatus": "Pending"
                },
                {
                    "PolicyText": "{\"Type\":\"ExpirationNotification\",\"Version\":
\"1.0\",\"Attributes\":{\"Before\":\"12\",\"Unit\":\"Hours\"}}",
                    "PolicyType": "ExpirationNotification",
                    "PolicyStatus": "Pending"
                }
            ]
        },
        {
            "Name": "MyStringListParameter",
            "Type": "StringList",
            "LastModifiedDate": 1582154764.222,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is a StringList parameter",
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "MyStringParameter",
```

```
            "Type": "String",
            "LastModifiedDate": 1582154711.976,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Alejandro-Rosalez",
            "Description": "This is a String parameter",
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "latestAmi",
            "Type": "String",
            "LastModifiedDate": 1580862415.521,
            "LastModifiedUser": "arn:aws:sts::111222333444:assumed-role/lambda-ssm-
role/Automation-UpdateSSM-Param",
            "Version": 3,
            "Tier": "Standard",
            "Policies": []
        }
    ]
}
```

範例 2：列出符合特定中繼資料的所有參數

此 `describe-parameters` 範例列出了符合篩選條件的所有參數。

aws ssm describe-parameters --filters "Key=Type,Values=StringList"

輸出：

```
{
    "Parameters": [
        {
            "Name": "MyStringListParameter",
            "Type": "StringList",
            "LastModifiedDate": 1582154764.222,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is a StringList parameter",
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Searching for Systems Manager Parameters。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeParameter。

describe-patch-baselines

以下程式碼範例顯示如何使用 describe-patch-baselines。

AWS CLI

範例 1：列出所有修補基準

下列 describe-patch-baselines 範例示範擷取目前區域的帳戶中所有修補基準的詳細資訊。

```
aws ssm describe-patch-baselines
```

輸出：

```
{
    "BaselineIdentities": [
        {
            "BaselineName": "AWS-SuseDefaultPatchBaseline",
            "DefaultBaseline": true,
            "BaselineDescription": "Default Patch Baseline for Suse Provided by
 AWS.",
            "BaselineId": "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-0123fdb36e334a3b2",
            "OperatingSystem": "SUSE"
        },
        {
            "BaselineName": "AWS-DefaultPatchBaseline",
            "DefaultBaseline": false,
            "BaselineDescription": "Default Patch Baseline Provided by AWS.",
            "BaselineId": "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-020d361a05defe4ed",
            "OperatingSystem": "WINDOWS"
        },
        ...
        {
            "BaselineName": "MyWindowsPatchBaseline",
            "DefaultBaseline": true,
```

```
            "BaselineDescription": "My patch baseline for EC2 instances for Windows
 Server",
            "BaselineId": "pb-0ad00e0dd7EXAMPLE",
            "OperatingSystem": "WINDOWS"
        }
    ]
}
```

範例 2：列出 提供的所有修補程式基準 AWS

下列describe-patch-baselines範例列出 提供的所有修補程式基準 AWS。

```
aws ssm describe-patch-baselines \
    --filters "Key=OWNER,Values=[AWS]"
```

範例 3：列出您擁有的所有修補基準

下列 describe-patch-baselines 範例列出了在目前區域的帳戶中建立的所有自訂修補基準。

```
aws ssm describe-patch-baselines \
    --filters "Key=OWNER,Values=[Self]"
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 About Predefined and Custom Patch Baselines。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribePatchBaselines。

describe-patch-group-state

以下程式碼範例顯示如何使用 describe-patch-group-state。

AWS CLI

取得修補程式群組的狀態

下列 describe-patch-group-state 範例示範擷取修補程式群組的高階修補程式合規摘要。

```
aws ssm describe-patch-group-state \
    --patch-group "Production"
```

輸出：

```
{
    "Instances": 21,
    "InstancesWithCriticalNonCompliantPatches": 1,
    "InstancesWithFailedPatches": 2,
    "InstancesWithInstalledOtherPatches": 3,
    "InstancesWithInstalledPatches": 21,
    "InstancesWithInstalledPendingRebootPatches": 2,
    "InstancesWithInstalledRejectedPatches": 1,
    "InstancesWithMissingPatches": 3,
    "InstancesWithNotApplicablePatches": 4,
    "InstancesWithOtherNonCompliantPatches": 1,
    "InstancesWithSecurityNonCompliantPatches": 1,
    "InstancesWithUnreportedNotApplicablePatches": 2
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 About patch groups <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-patchgroups.html>__ 和 Understanding patch compliance state values。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribePatchGroupState。

describe-patch-groups

以下程式碼範例顯示如何使用 describe-patch-groups。

AWS CLI

顯示修補程式群組註冊

下列 describe-patch-groups 範例列出了修補程式群組註冊。

```
aws ssm describe-patch-groups
```

輸出：

```
{
    "Mappings": [
        {
            "PatchGroup": "Production",
            "BaselineIdentity": {
                "BaselineId": "pb-0123456789abcdef0",
                "BaselineName": "ProdPatching",
```

```
                "OperatingSystem": "WINDOWS",
                "BaselineDescription": "Patches for Production",
                "DefaultBaseline": false
            }
        },
        {
            "PatchGroup": "Development",
            "BaselineIdentity": {
                "BaselineId": "pb-0713accee01234567",
                "BaselineName": "DevPatching",
                "OperatingSystem": "WINDOWS",
                "BaselineDescription": "Patches for Development",
                "DefaultBaseline": true
            }
        },
        ...
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Create a Patch Group <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html>__ 和 Add a Patch Group to a Patch Baseline。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribePatchGroups。

describe-patch-properties

以下程式碼範例顯示如何使用 `describe-patch-properties`。

AWS CLI

列出 Amazon Linux 修補程式可用性

下列`describe-patch-properties`範例顯示 Amazon Linux 產品清單，其中包含您 AWS 帳戶中可用的修補程式。

```
aws ssm describe-patch-properties \
    --operating-system AMAZON_LINUX \
    --property PRODUCT
```

輸出：

```
{
```

```
    "Properties": [
        {
            "Name": "AmazonLinux2012.03"
        },
        {
            "Name": "AmazonLinux2012.09"
        },
        {
            "Name": "AmazonLinux2013.03"
        },
        {
            "Name": "AmazonLinux2013.09"
        },
        {
            "Name": "AmazonLinux2014.03"
        },
        {
            "Name": "AmazonLinux2014.09"
        },
        {
            "Name": "AmazonLinux2015.03"
        },
        {
            "Name": "AmazonLinux2015.09"
        },
        {
            "Name": "AmazonLinux2016.03"
        },
        {
            "Name": "AmazonLinux2016.09"
        },
        {
            "Name": "AmazonLinux2017.03"
        },
        {
            "Name": "AmazonLinux2017.09"
        },
        {
            "Name": "AmazonLinux2018.03"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 About Patch Baselines。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribePatchProperties。

describe-sessions

以下程式碼範例顯示如何使用 describe-sessions。

AWS CLI

範例 1：列出所有作用中的 Session Manager 工作階段

此describe-sessions範例會擷取過去 30 天內由指定使用者啟動，最近（連線和中斷連線的工作階段）建立的作用中工作階段清單。此命令只會傳回使用 Session Manager 啟動之目標連線的結果。它不會列出透過其他方式進行的連線，例如遠端桌面連線或 SSH。

```
aws ssm describe-sessions \
    --state "Active" \
    --filters "key=Owner,value=arn:aws:sts::123456789012:assumed-role/Administrator/
Shirley-Rodriguez"
```

輸出：

```
{
    "Sessions": [
        {
            "SessionId": "John-07a16060613c408b5",
            "Target": "i-1234567890abcdef0",
            "Status": "Connected",
            "StartDate": 1550676938.352,
            "Owner": "arn:aws:sts::123456789012:assumed-role/Administrator/Shirley-
Rodriguez",
            "OutputUrl": {}
        },
        {
            "SessionId": "John-01edf534b8b56e8eb",
            "Target": "i-9876543210abcdef0",
            "Status": "Connected",
            "StartDate": 1550676842.194,
            "Owner": "arn:aws:sts::123456789012:assumed-role/Administrator/Shirley-
Rodriguez",
            "OutputUrl": {}
        }
```

```
    ]
}
```

範例 2：列出所有已終止的 Session Manager 工作階段

此`describe-sessions`範例會擷取所有使用者過去 30 天內最近終止的工作階段清單。

```
aws ssm describe-sessions \
    --state "History"
```

輸出：

```
{
    "Sessions": [
        {
            "SessionId": "Mary-Major-0022b1eb2b0d9e3bd",
            "Target": "i-1234567890abcdef0",
            "Status": "Terminated",
            "StartDate": 1550520701.256,
            "EndDate": 1550521931.563,
            "Owner": "arn:aws:sts::123456789012:assumed-role/Administrator/Mary-
Major"
        },
        {
            "SessionId": "Jane-Roe-0db53f487931ed9d4",
            "Target": "i-9876543210abcdef0",
            "Status": "Terminated",
            "StartDate": 1550161369.149,
            "EndDate": 1550162580.329,
            "Owner": "arn:aws:sts::123456789012:assumed-role/Administrator/Jane-Roe"
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的檢視工作階段歷史記錄。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeSessions。

`disassociate-ops-item-related-item`

以下程式碼範例顯示如何使用 `disassociate-ops-item-related-item`。

AWS CLI

刪除相關項目關聯

下列disassociate-ops-item-related-item範例會刪除 OpsItem 與相關項目之間的關聯。

```
aws ssm disassociate-ops-item-related-item \
    --ops-item-id "oi-f99f2EXAMPLE" \
    --association-id "e2036148-cccb-490e-ac2a-390e5EXAMPLE"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Systems Manager 使用者指南中的在 OpsCenter 中使用 Incident Manager 事件。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateOpsItemRelatedItem。

get-automation-execution

以下程式碼範例顯示如何使用 get-automation-execution。

AWS CLI

顯示有關自動化執行的詳細資訊

下列 get-automation-execution 範例顯示有關自動化執行的詳細資訊。

```
aws ssm get-automation-execution \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

輸出：

```
{
    "AutomationExecution": {
        "AutomationExecutionId": "73c8eef8-f4ee-4a05-820c-e354fEXAMPLE",
        "DocumentName": "AWS-StartEC2Instance",
        "DocumentVersion": "1",
        "ExecutionStartTime": 1583737233.748,
        "ExecutionEndTime": 1583737234.719,
        "AutomationExecutionStatus": "Success",
        "StepExecutions": [
```

```
            {
                "StepName": "startInstances",
                "Action": "aws:changeInstanceState",
                "ExecutionStartTime": 1583737234.134,
                "ExecutionEndTime": 1583737234.672,
                "StepStatus": "Success",
                "Inputs": {
                    "DesiredState": "\"running\"",
                    "InstanceIds": "[\"i-0cb99161f6EXAMPLE\"]"
                },
                "Outputs": {
                    "InstanceStates": [
                        "running"
                    ]
                },
                "StepExecutionId": "95e70479-cf20-4d80-8018-7e4e2EXAMPLE",
                "OverriddenParameters": {}
            }
        ],
        "StepExecutionsTruncated": false,
        "Parameters": {
            "AutomationAssumeRole": [
                ""
            ],
            "InstanceId": [
                "i-0cb99161f6EXAMPLE"
            ]
        },
        "Outputs": {},
        "Mode": "Auto",
        "ExecutedBy": "arn:aws:sts::29884EXAMPLE:assumed-role/mw_service_role/
OrchestrationService",
        "Targets": [],
        "ResolvedTargets": {
            "ParameterValues": [],
            "Truncated": false
        }
    }
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的逐步解說：修補 Linux AMI (AWS CLI)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetAutomationExecution。

get-calendar-state

以下程式碼範例顯示如何使用 get-calendar-state。

AWS CLI

範例 1：取得變更行事曆的目前狀態

此get-calendar-state範例會傳回目前時間的行事曆狀態。由於範例未指定時間，因此會報告行事曆的目前狀態。

```
aws ssm get-calendar-state \
    --calendar-names "MyCalendar"
```

輸出：

```
{
    "State": "OPEN",
    "AtTime": "2020-02-19T22:28:51Z",
    "NextTransitionTime": "2020-02-24T21:15:19Z"
}
```

範例 2：取得指定時間的變更行事曆狀態

此get-calendar-state範例會傳回指定時間的行事曆狀態。

```
aws ssm get-calendar-state \
    --calendar-names "MyCalendar" \
    --at-time "2020-07-19T21:15:19Z"
```

輸出：

```
{
    "State": "CLOSED",
    "AtTime": "2020-07-19T21:15:19Z"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的取得變更行事曆的狀態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetCalendarState。

get-command-invocation

以下程式碼範例顯示如何使用 get-command-invocation。

AWS CLI

顯示命令調用的詳細資訊

下列 get-command-invocation 範例列出了指定執行個體上指定命令的所有調用。

```
aws ssm get-command-invocation \
    --command-id "ef7fdfd8-9b57-4151-a15c-db9a12345678" \
    --instance-id "i-1234567890abcdef0"
```

輸出：

```
{
    "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",
    "InstanceId": "i-1234567890abcdef0",
    "Comment": "b48291dd-ba76-43e0-b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-
d6ce8EXAMPLE",
    "DocumentName": "AWS-UpdateSSMAgent",
    "DocumentVersion": "",
    "PluginName": "aws:updateSsmAgent",
    "ResponseCode": 0,
    "ExecutionStartDateTime": "2020-02-19T18:18:03.419Z",
    "ExecutionElapsedTime": "PT0.091S",
    "ExecutionEndDateTime": "2020-02-19T18:18:03.419Z",
    "Status": "Success",
    "StatusDetails": "Success",
    "StandardOutputContent": "Updating amazon-ssm-agent from 2.3.842.0 to latest
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/
ssm-agent-manifest.json\namazon-ssm-agent 2.3.842.0 has already been installed,
 update skipped\n",
    "StandardOutputUrl": "",
    "StandardErrorContent": "",
    "StandardErrorUrl": "",
    "CloudWatchOutputConfig": {
        "CloudWatchLogGroupName": "",
        "CloudWatchOutputEnabled": false
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Understanding Command Statuses。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetCommandInvocation。

get-connection-status

以下程式碼範例顯示如何使用 get-connection-status。

AWS CLI

顯示受管執行個體的連線狀態

此 get-connection-status 範例傳回指定受管執行個體的連線狀態。

```
aws ssm get-connection-status \
    --target i-1234567890abcdef0
```

輸出：

```
{
    "Target": "i-1234567890abcdef0",
    "Status": "connected"
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetConnectionStatus。

get-default-patch-baseline

以下程式碼範例顯示如何使用 get-default-patch-baseline。

AWS CLI

範例 1：顯示 Windows 預設修補基準

下列 get-default-patch-baseline 範例示範擷取 Windows Server 預設修補基準的詳細資訊。

```
aws ssm get-default-patch-baseline
```

輸出：

```
{
  "BaselineId": "pb-0713accee01612345",
  "OperatingSystem": "WINDOWS"
}
```

範例 2：顯示 Amazon Linux 預設修補基準

下列 get-default-patch-baseline 範例示範擷取 Amazon Linux 預設修補基準的詳細資訊。

```
aws ssm get-default-patch-baseline \
    --operating-system AMAZON_LINUX
```

輸出：

```
{
    "BaselineId": "pb-047c6eb9c8fc12345",
    "OperatingSystem": "AMAZON_LINUX"
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 About Predefined and Custom Patch Baselines <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-baselines.html>__ 和 Set an Existing Patch Baseline as the Default。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetDefaultPatchBaseline。

get-deployable-patch-snapshot-for-instance

以下程式碼範例顯示如何使用 get-deployable-patch-snapshot-for-instance。

AWS CLI

擷取執行個體使用的修補基準的目前快照

下列 get-deployable-patch-snapshot-for-instance 範例示範擷取執行個體所用之指定修補基準的目前快照詳細資訊。此命令必須使用執行個體憑證從執行個體執行。若要確保它使用執行個體憑證，請執行 aws configure，並僅指定執行個體的區域。將 Access Key 和 Secret Key 欄位留空。

秘訣：使用 uuidgen 來產生 snapshot-id。

```
aws ssm get-deployable-patch-snapshot-for-instance \
    --instance-id "i-1234567890abcdef0" \
    --snapshot-id "521c3536-930c-4aa9-950e-01234567abcd"
```

輸出：

```
{
    "InstanceId": "i-1234567890abcdef0",
    "SnapshotId": "521c3536-930c-4aa9-950e-01234567abcd",
    "Product": "AmazonLinux2018.03",
    "SnapshotDownloadUrl": "https://patch-baseline-snapshot-us-
east-1.s3.amazonaws.com/
ed85194ef27214f5984f28b4d664d14f7313568fea7d4b6ac6c10ad1f729d7e7-773304212436/
AMAZON_LINUX-521c3536-930c-4aa9-950e-01234567abcd?X-Amz-Algorithm=AWS4-HMAC-
SHA256&X-Amz-Date=20190215T164031Z&X-Amz-SignedHeaders=host&X-Amz-Expires=86400&X-
Amz-Credential=AKIAJ5C56P35AEBRX2QQ%2F20190215%2Fus-east-1%2Fs3%2Faws4_request&X-
Amz-Signature=efaaaf6e3878e77f48a6697e015efdbda9c426b09c5822055075c062f6ad2149"
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Parameter name: Snapshot ID。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetDeployablePatchSnapshotForInstance。

get-document

以下程式碼範例顯示如何使用 get-document。

AWS CLI

取得文件內容

下列 get-document 範例顯示 Systems Manager 文件的內容。

```
aws ssm get-document \
    --name "AWS-RunShellScript"
```

輸出：

```
{
    "Name": "AWS-RunShellScript",
    "DocumentVersion": "1",
    "Status": "Active",
    "Content": "{\n    \"schemaVersion\":\"1.2\",\n    \"description\":\"Run a
 shell script or specify the commands to run.\",\n    \"parameters\":{\n
 \"commands\":{\n            \"type\":\"StringList\",\n            \"description
\":\"(Required) Specify a shell script or a command to run.\",\n
 \"minItems\":1,\n            \"displayType\":\"textarea\"\n        },\n
 \"workingDirectory\":{\n            \"type\":\"String\",\n            \"default
\":\"\",\n            \"description\":\"(Optional) The path to the working
 directory on your instance.\",\n            \"maxChars\":4096\n        },\n
   \"executionTimeout\":{\n            \"type\":\"String\",\n            \"default
\":\"3600\",\n            \"description\":\"(Optional) The time in seconds for a
 command to complete before it is considered to have failed. Default is 3600 (1
 hour). Maximum is 172800 (48 hours).\",\n            \"allowedPattern\":\"([1-9]
[0-9]{0,4})|(1[0-6][0-9]{4})|(17[0-1][0-9]{3})|(172[0-7][0-9]{2})|(172800)\"\n
        }\n    },\n    \"runtimeConfig\":{\n        \"aws:runShellScript\":{\n
            \"properties\":[\n                {\n                    \"id\":
\"0.aws:runShellScript\",\n                    \"runCommand\":\"{{ commands }}\",\n
                    \"workingDirectory\":\"{{ workingDirectory }}\",\n
       \"timeoutSeconds\":\"{{ executionTimeout }}\"\n                }\n
  ]\n        }\n    }\n}\n",
    "DocumentType": "Command",
    "DocumentFormat": "JSON"
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 AWS Systems Manager Documents。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetDocument。

get-inventory-schema

以下程式碼範例顯示如何使用 get-inventory-schema。

AWS CLI

檢視庫存結構描述

此範例傳回帳戶的庫存類型名稱清單。

命令：

```
aws ssm get-inventory-schema
```

輸出：

```
{
  "Schemas": [
      {
          "TypeName": "AWS:AWSComponent",
          "Version": "1.0",
          "Attributes": [
              {
                  "Name": "Name",
                  "DataType": "STRING"
              },
              {
                  "Name": "ApplicationType",
                  "DataType": "STRING"
              },
              {
                  "Name": "Publisher",
                  "DataType": "STRING"
              },
              {
                  "Name": "Version",
                  "DataType": "STRING"
              },
              {
                  "Name": "InstalledTime",
                  "DataType": "STRING"
              },
              {
                  "Name": "Architecture",
                  "DataType": "STRING"
              },
              {
                  "Name": "URL",
                  "DataType": "STRING"
              }
          ]
      },
      ...
  ],
  "NextToken": "--token string truncated--"
```

```
}
```

檢視特定庫存類型的庫存結構描述

此範例會傳回 AWS：AWS元件庫存類型的庫存結構描述。

命令：

```
aws ssm get-inventory-schema --type-name "AWS:AWSComponent"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetInventorySchema。

get-inventory

以下程式碼範例顯示如何使用 get-inventory。

AWS CLI

檢視庫存

此範例示範取得庫存的自訂中繼資料。

命令：

```
aws ssm get-inventory
```

輸出：

```
{
  "Entities": [
      {
          "Data": {
              "AWS:InstanceInformation": {
                  "Content": [
                      {
                          "ComputerName": "ip-172-31-44-222.us-
west-2.compute.internal",
                          "InstanceId": "i-0cb2b964d3e14fd9f",
                          "IpAddress": "172.31.44.222",
                          "AgentType": "amazon-ssm-agent",
                          "ResourceType": "EC2Instance",
                          "AgentVersion": "2.0.672.0",
                          "PlatformVersion": "2016.09",
```

```
                        "PlatformName": "Amazon Linux AMI",
                        "PlatformType": "Linux"
                    }
                ],
                "TypeName": "AWS:InstanceInformation",
                "SchemaVersion": "1.0",
                "CaptureTime": "2017-02-20T18:03:58Z"
            }
        },
        "Id": "i-0cb2b964d3e14fd9f"
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetInventory。

get-maintenance-window-execution-task-invocation

以下程式碼範例顯示如何使用 get-maintenance-window-execution-task-invocation。

AWS CLI

取得維護時段任務調用的相關資訊

下列get-maintenance-window-execution-task-invocation範例列出屬於指定維護時段執行一部分的指定任務調用相關資訊。

```
aws ssm get-maintenance-window-execution-task-invocation \
    --window-execution-id "bc494bfa-e63b-49f6-8ad1-aa9f2EXAMPLE" \
    --task-id "96f2ad59-97e3-461d-a63d-40c8aEXAMPLE" \
    --invocation-id "a5273e2c-d2c6-4880-b3e1-5e550EXAMPLE"
```

輸出：

```
{
    "Status": "SUCCESS",
    "Parameters": "{\"comment\":\"\",\"documentName\":\"AWS-RunPowerShellScript\",
\"instanceIds\":[\"i-1234567890EXAMPLE\"],\"maxConcurrency\":\"1\",\"maxErrors\":
\"1\",\"parameters\":{\"executionTimeout\":[\"3600\"],\"workingDirectory\":[\"\"],
\"commands\":[\"echo Hello\"]},\"timeoutSeconds\":600}",
    "ExecutionId": "03b6baa0-5460-4e15-83f2-ea685EXAMPLE",
    "InvocationId": "a5273e2c-d2c6-4880-b3e1-5e550EXAMPLE",
```

```
    "StartTime": 1549998326.421,
    "TaskType": "RUN_COMMAND",
    "EndTime": 1550001931.784,
    "WindowExecutionId": "bc494bfa-e63b-49f6-8ad1-aa9f2EXAMPLE",
    "StatusDetails": "Failed",
    "TaskExecutionId": "96f2ad59-97e3-461d-a63d-40c8aEXAMPLE"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的檢視任務和任務執行 (AWS CLI) 的相關資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMaintenanceWindowExecutionTaskInvocation。

get-maintenance-window-execution-task

以下程式碼範例顯示如何使用 get-maintenance-window-execution-task。

AWS CLI

取得維護時段任務執行的相關資訊

下列 get-maintenance-window-execution-task 範例列出了有關屬於指定維護時段執行之任務的資訊。

```
aws ssm get-maintenance-window-execution-task \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE" \
    --task-id "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
```

輸出：

```
{
    "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
    "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE",
    "TaskArn": "AWS-RunPatchBaseline",
    "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
    "Type": "RUN_COMMAND",
    "TaskParameters": [
        {
            "BaselineOverride": {
                "Values": [
                    ""
```

```
                ]
            },
            "InstallOverrideList": {
                "Values": [
                    ""
                ]
            },
            "Operation": {
                "Values": [
                    "Scan"
                ]
            },
            "RebootOption": {
                "Values": [
                    "RebootIfNeeded"
                ]
            },
            "SnapshotId": {
                "Values": [
                    "{{ aws:ORCHESTRATION_ID }}"
                ]
            },
            "aws:InstanceId": {
                "Values": [
                    "i-02573cafcfEXAMPLE",
                    "i-0471e04240EXAMPLE",
                    "i-07782c72faEXAMPLE"
                ]
            }
        }
    ],
    "Priority": 1,
    "MaxConcurrency": "1",
    "MaxErrors": "3",
    "Status": "SUCCESS",
    "StartTime": "2021-08-04T11:45:35.088000-07:00",
    "EndTime": "2021-08-04T11:53:09.079000-07:00"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的檢視任務和任務執行 (AWS CLI) 的相關資訊。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetMaintenanceWindowExecutionTask。

get-maintenance-window-execution

以下程式碼範例顯示如何使用 get-maintenance-window-execution。

AWS CLI

取得維護時段任務執行的相關資訊

下列 get-maintenance-window-execution 範例列出了有關作為指定維護時段執行的一部分執行之任務的資訊。

```
aws ssm get-maintenance-window-execution \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

輸出：

```
{
    "Status": "SUCCESS",
    "TaskIds": [
        "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
    ],
    "StartTime": 1487692834.595,
    "EndTime": 1487692835.051,
    "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的檢視任務和任務執行 (AWS CLI) 的相關資訊。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetMaintenanceWindowExecution。

get-maintenance-window-task

以下程式碼範例顯示如何使用 get-maintenance-window-task。

AWS CLI

取得維護時段任務的相關資訊

下列get-maintenance-window-task範例會擷取指定維護時段任務的詳細資訊。

```
aws ssm get-maintenance-window-task \
```

```
    --window-id mw-0c5ed765acEXAMPLE \
    --window-task-id 0e842a8d-2d44-4886-bb62-af8dcEXAMPLE
```

輸出：

```
{
    "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
    "MaxErrors": "1",
    "TaskArn": "AWS-RunPowerShellScript",
    "MaxConcurrency": "1",
    "WindowTaskId": "0e842a8d-2d44-4886-bb62-af8dcEXAMPLE",
    "TaskParameters": {},
    "Priority": 1,
    "TaskInvocationParameters": {
        "RunCommand": {
            "Comment": "",
            "TimeoutSeconds": 600,
            "Parameters": {
                "commands": [
                    "echo Hello"
                ],
                "executionTimeout": [
                    "3600"
                ],
                "workingDirectory": [
                    ""
                ]
            }
        }
    },
    "WindowId": "mw-0c5ed765acEXAMPLE",
    "TaskType": "RUN_COMMAND",
    "Targets": [
        {
            "Values": [
                "84c818da-b619-4d3d-9651-946f3EXAMPLE"
            ],
            "Key": "WindowTargetIds"
        }
    ],
    "Name": "ExampleTask"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的檢視維護 Windows (AWS CLI) 的相關資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMaintenanceWindowTask。

get-maintenance-window

以下程式碼範例顯示如何使用 get-maintenance-window。

AWS CLI

取得有關維護時段的資訊

下列 get-maintenance-window 範例示範擷取有關指定維護時段的詳細資訊。

```
aws ssm get-maintenance-window \
    --window-id "mw-03eb9db428EXAMPLE"
```

輸出：

```
{
    "AllowUnassociatedTargets": true,
    "CreatedDate": 1515006912.957,
    "Cutoff": 1,
    "Duration": 6,
    "Enabled": true,
    "ModifiedDate": 2020-01-01T10:04:04.099Z,
    "Name": "My-Maintenance-Window",
    "Schedule": "rate(3 days)",
    "WindowId": "mw-03eb9db428EXAMPLE",
    "NextExecutionTime": "2020-02-25T00:08:15.099Z"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的檢視維護時段 (AWS CLI) 的相關資訊。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetMaintenanceWindow。

get-ops-item

以下程式碼範例顯示如何使用 get-ops-item。

AWS CLI

檢視 OpsItem 的相關資訊

下列get-ops-item範例顯示指定 OpsItem 的詳細資訊。

```
aws ssm get-ops-item \
    --ops-item-id oi-0b725EXAMPLE
```

輸出：

```
{
    "OpsItem": {
        "CreatedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-Role/
fbf77cbe264a33509569f23e4EXAMPLE",
        "CreatedTime": "2019-12-04T15:52:16.793000-08:00",
        "Description": "CloudWatch Event Rule SSMOpsItems-EC2-instance-terminated
 was triggered. Your EC2 instance has terminated. See below for more details.",
        "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-Role/
fbf77cbe264a33509569f23e4EXAMPLE",
        "LastModifiedTime": "2019-12-04T15:52:16.793000-08:00",
        "Notifications": [],
        "RelatedOpsItems": [],
        "Status": "Open",
        "OpsItemId": "oi-0b725EXAMPLE",
        "Title": "EC2 instance terminated",
        "Source": "EC2",
        "OperationalData": {
            "/aws/automations": {
                "Value": "[ { \"automationType\": \"AWS:SSM:Automation\",
 \"automationId\": \"AWS-CreateManagedWindowsInstance\" }, { \"automationType\":
 \"AWS:SSM:Automation\", \"automationId\": \"AWS-CreateManagedLinuxInstance\" } ]",
                "Type": "SearchableString"
            },
            "/aws/dedup": {
                "Value": "{\"dedupString\":\"SSMOpsItems-EC2-instance-terminated
\"}",
                "Type": "SearchableString"
            },
            "/aws/resources": {
                "Value": "[{\"arn\":\"arn:aws:ec2:us-east-2:111222333444:instance/
i-05adec7e97EXAMPLE\"}]",
                "Type": "SearchableString"
```

```
            },
            "event-time": {
                "Value": "2019-12-04T23:52:16Z",
                "Type": "String"
            },
            "instance-state": {
                "Value": "terminated",
                "Type": "String"
            }
        },
        "Category": "Availability",
        "Severity": "4"
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with OpsItems。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetOpsItem。

get-ops-summary

以下程式碼範例顯示如何使用 get-ops-summary。

AWS CLI

檢視所有 OpsItems 的摘要

下列get-ops-summary範例顯示您 AWS 帳戶中所有 OpsItems 的摘要。

```
aws ssm get-ops-summary
```

輸出：

```
{
    "Entities": [
        {
            "Id": "oi-4309fEXAMPLE",
            "Data": {
                "AWS:OpsItem": {
                    "CaptureTime": "2020-02-26T18:58:32.918Z",
                    "Content": [
                        {
                            "AccountId": "111222333444",
```

```
                            "Category": "Availability",
                            "CreatedBy": "arn:aws:sts::111222333444:assumed-role/
OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
                            "CreatedTime": "2020-02-26T19:10:44.149Z",
                            "Description": "CloudWatch Event Rule SSMOpsItems-EC2-
instance-terminated was triggered. Your EC2 instance has terminated. See below for
 more details.",
                            "LastModifiedBy": "arn:aws:sts::111222333444:assumed-
role/OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
                            "LastModifiedTime": "2020-02-26T19:10:44.149Z",
                            "Notifications": "",
                            "OperationalData": "{\"/aws/automations\":
{\"type\":\"SearchableString\",\"value\":\"[ { \\\"automationType\\\": \\
\"AWS:SSM:Automation\\\", \\\"automationId\\\": \\\"AWS-CreateManagedWindowsInstance
\\\" }, { \\\"automationType\\\": \\\"AWS:SSM:Automation\\\", \\\"automationId
\\\": \\\"AWS-CreateManagedLinuxInstance\\\" } ]\"},\"/aws/resources\":
{\"type\":\"SearchableString\",\"value\":\"[{\\\"arn\\\":\\\"arn:aws:ec2:us-
east-2:111222333444:instance/i-0acbd0800fEXAMPLE\\\"}]\"},\"/aws/dedup\":{\"type\":
\"SearchableString\",\"value\":\"{\\\"dedupString\\\":\\\"SSMOpsItems-EC2-instance-
terminated\\\"}\"}}",
                            "OpsItemId": "oi-4309fEXAMPLE",
                            "RelatedItems": "",
                            "Severity": "3",
                            "Source": "EC2",
                            "Status": "Open",
                            "Title": "EC2 instance terminated"
                        }
                    ]
                }
            }
        },
        {
            "Id": "oi-bb2a0e6a4541",
            "Data": {
                "AWS:OpsItem": {
                    "CaptureTime": "2019-11-26T19:20:06.161Z",
                    "Content": [
                        {
                            "AccountId": "111222333444",
                            "Category": "Availability",
                            "CreatedBy": "arn:aws:sts::111222333444:assumed-role/
OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
                            "CreatedTime": "2019-11-26T20:00:07.237Z",
```

```
                            "Description": "CloudWatch Event Rule SSMOpsItems-SSM-
maintenance-window-execution-failed was triggered. Your SSM Maintenance Window
 execution has failed. See below for more details.",
                            "LastModifiedBy": "arn:aws:sts::111222333444:assumed-
role/OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
                            "LastModifiedTime": "2019-11-26T20:00:07.237Z",
                            "Notifications": "",
                            "OperationalData": "{\"/aws/resources\":{\"type
\":\"SearchableString\",\"value\":\"[{\\\"arn\\\":\\\"arn:aws:ssm:us-
east-2:111222333444:maintenancewindow/mw-0e83ba440dEXAMPLE\\\"}]\"},\"/aws/dedup\":
{\"type\":\"SearchableString\",\"value\":\"{\\\"dedupString\\\":\\\"SSMOpsItems-SSM-
maintenance-window-execution-failed\\\"}\"}}",
                            "OpsItemId": "oi-bb2a0EXAMPLE",
                            "RelatedItems": "",
                            "Severity": "3",
                            "Source": "SSM",
                            "Status": "Open",
                            "Title": "SSM Maintenance Window execution failed"
                        }
                    ]
                }
            }
        ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with OpsItems。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetOpsSummary。

get-parameter-history

以下程式碼範例顯示如何使用 get-parameter-history。

AWS CLI

取得參數的值歷史記錄

下列 get-parameter-history 範例列出了指定參數的變更歷史記錄，包括其值。

```
aws ssm get-parameter-history \
    --name "MyStringParameter"
```

輸出：


```
{
    "Parameters": [
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "LastModifiedDate": 1582154711.976,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is the first version of my String parameter",
            "Value": "Veni",
            "Version": 1,
            "Labels": [],
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "LastModifiedDate": 1582156093.471,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is the second version of my String parameter",
            "Value": "Vidi",
            "Version": 2,
            "Labels": [],
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "LastModifiedDate": 1582156117.545,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is the third version of my String parameter",
            "Value": "Vici",
            "Version": 3,
            "Labels": [],
            "Tier": "Standard",
            "Policies": []
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with parameter versions。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetParameterHistory。

get-parameter

以下程式碼範例顯示如何使用 get-parameter。

AWS CLI

範例 1：顯示參數的值

下列 get-parameter 範例列出了指定單一參數的值。

```
aws ssm get-parameter \
    --name "MyStringParameter"
```

輸出：

```
{
    "Parameter": {
        "Name": "MyStringParameter",
        "Type": "String",
        "Value": "Veni",
        "Version": 1,
        "LastModifiedDate": 1530018761.888,
        "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/MyStringParameter"
        "DataType": "text"
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with Parameter Store。

範例 2：解密 SecureString 參數的值

下列 get-parameter 範例示範解密指定 SecureString 參數的值。

```
aws ssm get-parameter \
    --name "MySecureStringParameter" \
    --with-decryption
```

輸出：

```
{
    "Parameter": {
        "Name": "MySecureStringParameter",
        "Type": "SecureString",
        "Value": "16679b88-310b-4895-a943-e0764EXAMPLE",
        "Version": 2,
        "LastModifiedDate": 1582155479.205,
        "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/
MySecureStringParameter"
        "DataType": "text"
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with Parameter Store。

範例 3：使用標籤顯示參數的值

下列 `get-parameter` 範例列出具有指定標籤之指定單一參數的值。

```
aws ssm get-parameter \
    --name "MyParameter:label"
```

輸出：

```
{
    "Parameter": {
        "Name": "MyParameter",
        "Type": "String",
        "Value": "parameter version 2",
        "Version": 2,
        "Selector": ":label",
        "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",
        "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",
        "DataType": "text"
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with parameter labels。

範例 4：以版本顯示參數的值

下列 get-parameter 範例列出了指定單一參數版本的值。

```
aws ssm get-parameter \
    --name "MyParameter:2"
```

輸出：

```
{
    "Parameter": {
        "Name": "MyParameter",
        "Type": "String",
        "Value": "parameter version 2",
        "Version": 2,
        "Selector": ":2",
        "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",
        "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",
        "DataType": "text"
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with parameter labels。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetParameter。

get-parameters-by-path

以下程式碼範例顯示如何使用 get-parameters-by-path。

AWS CLI

列出特定路徑中的參數

下列get-parameters-by-path範例列出指定階層內的參數。

```
aws ssm get-parameters-by-path \
    --path "/site/newyork/department/"
```

輸出：

```
{
    "Parameters": [
        {
            "Name": "/site/newyork/department/marketing",
            "Type": "String",
            "Value": "Floor 2",
            "Version": 1,
            "LastModifiedDate": 1530018761.888,
            "ARN": "arn:aws:ssm:us-east-1:111222333444:parameter/site/newyork/
department/marketing"
        },
        {
            "Name": "/site/newyork/department/infotech",
            "Type": "String",
            "Value": "Floor 3",
            "Version": 1,
            "LastModifiedDate": 1530018823.429,
            "ARN": "arn:aws:ssm:us-east-1:111222333444:parameter/site/newyork/
department/infotech"
        },
        ...
    ]
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的使用參數階層。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetParametersByPath。

get-parameters

以下程式碼範例顯示如何使用 get-parameters。

AWS CLI

範例 1：列出參數的值

下列 get-parameters 範例列出了三個指定參數的值。

```
aws ssm get-parameters \
    --names "MyStringParameter" "MyStringListParameter" "MyInvalidParameterName"
```

輸出：

```
{
    "Parameters": [
        {
            "Name": "MyStringListParameter",
            "Type": "StringList",
            "Value": "alpha,beta,gamma",
            "Version": 1,
            "LastModifiedDate": 1582154764.222,
            "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/
MyStringListParameter"
            "DataType": "text"
        },
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "Value": "Vici",
            "Version": 3,
            "LastModifiedDate": 1582156117.545,
            "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/MyStringParameter"
            "DataType": "text"
        }
    ],
    "InvalidParameters": [
        "MyInvalidParameterName"
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with Parameter Store。

範例 2：使用 ``--query`` 選項列出多個參數的名稱和值

下列 `get-parameters` 範例列出了指定參數的名稱和值。

```
aws ssm get-parameters \
    --names MyStringParameter MyStringListParameter \
    --query "Parameters[*].{Name:Name,Value:Value}"
```

輸出：

```
[
    {
```

```
        "Name": "MyStringListParameter",
        "Value": "alpha,beta,gamma"
    },
    {
        "Name": "MyStringParameter",
        "Value": "Vidi"
    }
]
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with Parameter Store。

範例 3：使用標籤顯示參數的值

下列 `get-parameter` 範例列出具有指定標籤之指定單一參數的值。

```
aws ssm get-parameter \
    --name "MyParameter:label"
```

輸出：

```
{
    "Parameters": [
        {
            "Name": "MyLabelParameter",
            "Type": "String",
            "Value": "parameter by label",
            "Version": 1,
            "Selector": ":label",
            "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",
            "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",
            "DataType": "text"
        },
        {
            "Name": "MyVersionParameter",
            "Type": "String",
            "Value": "parameter by version",
            "Version": 2,
            "Selector": ":2",
            "LastModifiedDate": "2021-03-24T16:20:28.236000-07:00",
            "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/unlabel-param",
            "DataType": "text"
        }
```

```
    ],
    "InvalidParameters": []
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with parameter labels。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetParameters。

get-patch-baseline-for-patch-group

以下程式碼範例顯示如何使用 get-patch-baseline-for-patch-group。

AWS CLI

顯示修補程式群組的修補基準

下列 get-patch-baseline-for-patch-group 範例示範擷取有關指定修補程式群組之修補基準的詳細資訊。

```
aws ssm get-patch-baseline-for-patch-group \
    --patch-group "DEV"
```

輸出：

```
{
    "PatchGroup": "DEV",
    "BaselineId": "pb-0123456789abcdef0",
    "OperatingSystem": "WINDOWS"
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Create a Patch Group <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html>__ 和 Add a Patch Group to a Patch Baseline。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetPatchBaselineForPatchGroup。

get-patch-baseline

以下程式碼範例顯示如何使用 get-patch-baseline。

AWS CLI

顯示修補基準

下列 get-patch-baseline 範例示範擷取指定修補基準的詳細資訊。

```
aws ssm get-patch-baseline \
    --baseline-id "pb-0123456789abcdef0"
```

輸出：

```
{
    "BaselineId": "pb-0123456789abcdef0",
    "Name": "WindowsPatching",
    "OperatingSystem": "WINDOWS",
    "GlobalFilters": {
        "PatchFilters": []
    },
    "ApprovalRules": {
        "PatchRules": [
            {
                "PatchFilterGroup": {
                    "PatchFilters": [
                        {
                            "Key": "PRODUCT",
                            "Values": [
                                "WindowsServer2016"
                            ]
                        }
                    ]
                },
                "ComplianceLevel": "CRITICAL",
                "ApproveAfterDays": 0,
                "EnableNonSecurity": false
            }
        ]
    },
    "ApprovedPatches": [],
    "ApprovedPatchesComplianceLevel": "UNSPECIFIED",
    "ApprovedPatchesEnableNonSecurity": false,
    "RejectedPatches": [],
    "RejectedPatchesAction": "ALLOW_AS_DEPENDENCY",
    "PatchGroups": [
```

```
        "QA",
        "DEV"
    ],
    "CreatedDate": 1550244180.465,
    "ModifiedDate": 1550244180.465,
    "Description": "Patches for Windows Servers",
    "Sources": []
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 About Patch Baselines。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetPatchBaseline。

get-service-setting

以下程式碼範例顯示如何使用 get-service-setting。

AWS CLI

擷取參數存放區輸送量的服務設定

下列 get-service-setting 此範例會擷取指定區域中參數存放區輸送量的目前服務設定。

```
aws ssm get-service-setting \
    --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-store/high-throughput-enabled
```

輸出：

```
{
    "ServiceSetting": {
        "SettingId": "/ssm/parameter-store/high-throughput-enabled",
        "SettingValue": "false",
        "LastModifiedDate": 1555532818.578,
        "LastModifiedUser": "System",
        "ARN": "arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-store/high-throughput-enabled",
        "Status": "Default"
    }
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的增加參數存放區輸送量。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetServiceSetting。

label-parameter-version

以下程式碼範例顯示如何使用 label-parameter-version。

AWS CLI

範例 1：將標籤新增至參數的最新版本

下列label-parameter-version範例會將標籤新增至指定參數的最新版本。

```
aws ssm label-parameter-version \
    --name "MyStringParameter" \
    --labels "ProductionReady"
```

輸出：

```
{
    "InvalidLabels": [],
    "ParameterVersion": 3
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with parameter labels。

範例 2：將標籤新增至特定版本的參數

下列label-parameter-version範例會將標籤新增至指定版本的參數。

```
aws ssm label-parameter-version \
    --name "MyStringParameter" \
    --labels "ProductionReady" \
    --parameter-version "2" --labels "DevelopmentReady"
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with parameter labels。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 LabelParameterVersion。

list-association-versions

以下程式碼範例顯示如何使用 list-association-versions。

AWS CLI

列出特定關聯 ID 之關聯的所有版本

下列 list-association-versions 範例列出了指定關聯的所有版本。

```
aws ssm list-association-versions \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

輸出：

```
{
"AssociationVersions": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "CreatedDate": 1550505536.726,
            "Name": "AWS-UpdateSSMAgent",
            "Parameters": {
                "allowDowngrade": [
                    "false"
                ],
                "version": [
                    ""
                ]
            },
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-1234567890abcdef0"
                    ]
                }
            ],
            "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
            "AssociationName": "UpdateSSMAgent"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with associations in Systems Manager。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListAssociationVersions。

`list-associations`

以下程式碼範例顯示如何使用 `list-associations`。

AWS CLI

範例 1：列出特定執行個體的關聯

下列 list-associations 範例列出了具有 AssociationName、UpdateSSMAgent 的所有關聯。

```
aws ssm list-associations /
    --association-filter-list "key=AssociationName,value=UpdateSSMAgent"
```

輸出：

```
{
    "Associations": [
        {
            "Name": "AWS-UpdateSSMAgent",
            "InstanceId": "i-1234567890abcdef0",
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-016648b75dd622dab"
                    ]
                }
            ],
            "Overview": {
                "Status": "Pending",
                "DetailedStatus": "Associated",
                "AssociationStatusAggregatedCount": {
                    "Pending": 1
                }
            },
            "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
```

```
            "AssociationName": "UpdateSSMAgent"
        }
    ]
}
```

如需詳細資訊，請參閱《Systems Manager 使用者指南》中的 Working with associations in Systems Manager。

範例 2：列出特定文件的關聯

下列 list-associations 範例列出了指定文件的所有關聯。

```
aws ssm list-associations /
    --association-filter-list "key=Name,value=AWS-UpdateSSMAgent"
```

輸出：

```
{
    "Associations": [
        {
            "Name": "AWS-UpdateSSMAgent",
            "InstanceId": "i-1234567890abcdef0",
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-1234567890abcdef0"
                    ]
                }
            ],
            "LastExecutionDate": 1550505828.548,
            "Overview": {
                "Status": "Success",
                "DetailedStatus": "Success",
                "AssociationStatusAggregatedCount": {
                    "Success": 1
                }
            },
            "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
            "AssociationName": "UpdateSSMAgent"
        },
```

```
    {
            "Name": "AWS-UpdateSSMAgent",
            "InstanceId": "i-9876543210abcdef0",
            "AssociationId": "fbc07ef7-b985-4684-b82b-0123456789ab",
            "AssociationVersion": "1",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-9876543210abcdef0"
                    ]
                }
            ],
            "LastExecutionDate": 1550507531.0,
            "Overview": {
                "Status": "Success",
                "AssociationStatusAggregatedCount": {
                    "Success": 1
                }
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Systems Manager 使用者指南》中的 Working with associations in Systems Manager。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListAssociations。

`list-command-invocations`

以下程式碼範例顯示如何使用 `list-command-invocations`。

AWS CLI

列出特定命令的調用

下列 `list-command-invocations` 範例列出了命令的所有調用。

```
aws ssm list-command-invocations \
    --command-id "ef7fdfd8-9b57-4151-a15c-db9a12345678" \
    --details
```

輸出：


```
{
    "CommandInvocations": [
        {
            "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",
            "InstanceId": "i-02573cafcfEXAMPLE",
            "InstanceName": "",
            "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
            "DocumentName": "AWS-UpdateSSMAgent",
            "DocumentVersion": "",
            "RequestedDateTime": 1582136283.089,
            "Status": "Success",
            "StatusDetails": "Success",
            "StandardOutputUrl": "",
            "StandardErrorUrl": "",
            "CommandPlugins": [
                {
                    "Name": "aws:updateSsmAgent",
                    "Status": "Success",
                    "StatusDetails": "Success",
                    "ResponseCode": 0,
                    "ResponseStartDateTime": 1582136283.419,
                    "ResponseFinishDateTime": 1582136283.51,
                    "Output": "Updating amazon-ssm-agent from 2.3.842.0 to latest
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/
ssm-agent-manifest.json\namazon-ssm-agent 2.3.842.0 has already been installed,
 update skipped\n",
                    "StandardOutputUrl": "",
                    "StandardErrorUrl": "",
                    "OutputS3Region": "us-east-2",
                    "OutputS3BucketName": "",
                    "OutputS3KeyPrefix": ""
                }
            ],
            "ServiceRole": "",
            "NotificationConfig": {
                "NotificationArn": "",
                "NotificationEvents": [],
                "NotificationType": ""
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
```

```
                "CloudWatchOutputEnabled": false
            }
        },
        {
            "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",
            "InstanceId": "i-0471e04240EXAMPLE",
            "InstanceName": "",
            "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
            "DocumentName": "AWS-UpdateSSMAgent",
            "DocumentVersion": "",
            "RequestedDateTime": 1582136283.02,
            "Status": "Success",
            "StatusDetails": "Success",
            "StandardOutputUrl": "",
            "StandardErrorUrl": "",
            "CommandPlugins": [
                {
                    "Name": "aws:updateSsmAgent",
                    "Status": "Success",
                    "StatusDetails": "Success",
                    "ResponseCode": 0,
                    "ResponseStartDateTime": 1582136283.812,
                    "ResponseFinishDateTime": 1582136295.031,
                    "Output": "Updating amazon-ssm-agent from 2.3.672.0 to latest
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/
ssm-agent-manifest.json\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/
amazon-ssm-us-east-2/amazon-ssm-agent-updater/2.3.842.0/amazon-ssm-agent-updater-
snap-amd64.tar.gz\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/
amazon-ssm-us-east-2/amazon-ssm-agent/2.3.672.0/amazon-ssm-agent-snap-amd64.tar.gz
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/
amazon-ssm-agent/2.3.842.0/amazon-ssm-agent-snap-amd64.tar.gz\nInitiating amazon-
ssm-agent update to 2.3.842.0\namazon-ssm-agent updated successfully to 2.3.842.0",
                    "StandardOutputUrl": "",
                    "StandardErrorUrl": "",
                    "OutputS3Region": "us-east-2",
                    "OutputS3BucketName": "",
                    "OutputS3KeyPrefix": "8bee3135-398c-4d31-99b6-e42d2EXAMPLE/
i-0471e04240EXAMPLE/awsupdateSsmAgent"
                }
            ],
            "ServiceRole": "",
            "NotificationConfig": {
                "NotificationArn": "",
```

```
                "NotificationEvents": [],
                "NotificationType": ""
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Understanding Command Statuses。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListCommandInvocations。

`list-commands`

以下程式碼範例顯示如何使用 `list-commands`。

AWS CLI

範例 1：取得特定命令的狀態

下列 `list-commands` 範例示範擷取並顯示指定命令的狀態。

```
aws ssm list-commands \
    --command-id "0831e1a8-a1ac-4257-a1fd-c831bEXAMPLE"
```

範例 2：取得在特定日期後請求之命令的狀態

下列 `list-commands` 範例示範擷取在指定日期後請求的命令詳細資訊。

```
aws ssm list-commands \
    --filter "key=InvokedAfter,value=2020-02-01T00:00:00Z"
```

範例 3：列出 AWS 帳戶中請求的所有命令

下列`list-commands`範例列出目前 AWS 帳戶和區域中使用者請求的所有命令。

```
aws ssm list-commands
```

輸出：

```
{
    "Commands": [
        {
            "CommandId": "8bee3135-398c-4d31-99b6-e42d2EXAMPLE",
            "DocumentName": "AWS-UpdateSSMAgent",
            "DocumentVersion": "",
            "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
            "ExpiresAfter": "2020-02-19T11:28:02.500000-08:00",
            "Parameters": {},
            "InstanceIds": [
                "i-028ea792daEXAMPLE",
                "i-02feef8c46EXAMPLE",
                "i-038613f3f0EXAMPLE",
                "i-03a530a2d4EXAMPLE",
                "i-083b678d37EXAMPLE",
                "i-0dee81debaEXAMPLE"
            ],
            "Targets": [],
            "RequestedDateTime": "2020-02-19T10:18:02.500000-08:00",
            "Status": "Success",
            "StatusDetails": "Success",
            "OutputS3BucketName": "",
            "OutputS3KeyPrefix": "",
            "MaxConcurrency": "50",
            "MaxErrors": "100%",
            "TargetCount": 6,
            "CompletedCount": 6,
            "ErrorCount": 0,
            "DeliveryTimedOutCount": 0,
            "ServiceRole": "",
            "NotificationConfig": {
                "NotificationArn": "",
                "NotificationEvents": [],
                "NotificationType": ""
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        }
        {
```

```
            "CommandId": "e9ade581-c03d-476b-9b07-26667EXAMPLE",
            "DocumentName": "AWS-FindWindowsUpdates",
            "DocumentVersion": "1",
            "Comment": "",
            "ExpiresAfter": "2020-01-24T12:37:31.874000-08:00",
            "Parameters": {
                "KbArticleIds": [
                    ""
                ],
                "UpdateLevel": [
                    "All"
                ]
            },
            "InstanceIds": [],
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-00ec29b21eEXAMPLE",
                        "i-09911ddd90EXAMPLE"
                    ]
                }
            ],
            "RequestedDateTime": "2020-01-24T11:27:31.874000-08:00",
            "Status": "Success",
            "StatusDetails": "Success",
            "OutputS3BucketName": "my-us-east-2-bucket",
            "OutputS3KeyPrefix": "my-rc-output",
            "MaxConcurrency": "50",
            "MaxErrors": "0",
            "TargetCount": 2,
            "CompletedCount": 2,
            "ErrorCount": 0,
            "DeliveryTimedOutCount": 0,
            "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "NotificationConfig": {
                "NotificationArn": "arn:aws:sns:us-east-2:111222333444:my-us-east-2-
notification-arn",
                "NotificationEvents": [
                    "All"
                ],
                "NotificationType": "Invocation"
            },
```

```
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        }
        {
            "CommandId": "d539b6c3-70e8-4853-80e5-0ce4fEXAMPLE",
            "DocumentName": "AWS-RunPatchBaseline",
            "DocumentVersion": "1",
            "Comment": "",
            "ExpiresAfter": "2020-01-24T12:21:04.350000-08:00",
            "Parameters": {
                "InstallOverrideList": [
                    ""
                ],
                "Operation": [
                    "Install"
                ],
                "RebootOption": [
                    "RebootIfNeeded"
                ],
                "SnapshotId": [
                    ""
                ]
            },
            "InstanceIds": [],
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-00ec29b21eEXAMPLE",
                        "i-09911ddd90EXAMPLE"
                    ]
                }
            ],
            "RequestedDateTime": "2020-01-24T11:11:04.350000-08:00",
            "Status": "Success",
            "StatusDetails": "Success",
            "OutputS3BucketName": "my-us-east-2-bucket",
            "OutputS3KeyPrefix": "my-rc-output",
            "MaxConcurrency": "50",
            "MaxErrors": "0",
            "TargetCount": 2,
            "CompletedCount": 2,
```

```
            "ErrorCount": 0,
            "DeliveryTimedOutCount": 0,
            "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "NotificationConfig": {
                "NotificationArn": "arn:aws:sns:us-east-2:111222333444:my-us-east-2-
notification-arn",
                "NotificationEvents": [
                    "All"
                ],
                "NotificationType": "Invocation"
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Running Commands Using Systems Manager Run Command。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListCommands。

`list-compliance-items`

以下程式碼範例顯示如何使用 `list-compliance-items`。

AWS CLI

列出特定執行個體的合規項目

此範例列出了指定執行個體的所有合規項目。

命令：

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance"
```

輸出：

```
{
```

```
  "ComplianceItems": [
      {
          "ComplianceType": "Association",
          "ResourceType": "ManagedInstance",
          "ResourceId": "i-1234567890abcdef0",
          "Id": "8dfe3659-4309-493a-8755-0123456789ab",
          "Title": "",
          "Status": "COMPLIANT",
          "Severity": "UNSPECIFIED",
          "ExecutionSummary": {
              "ExecutionTime": 1550408470.0
          },
          "Details": {
              "DocumentName": "AWS-GatherSoftwareInventory",
              "DocumentVersion": "1"
          }
      },
      {
          "ComplianceType": "Association",
          "ResourceType": "ManagedInstance",
          "ResourceId": "i-1234567890abcdef0",
          "Id": "e4c2ed6d-516f-41aa-aa2a-0123456789ab",
          "Title": "",
          "Status": "COMPLIANT",
          "Severity": "UNSPECIFIED",
          "ExecutionSummary": {
              "ExecutionTime": 1550508475.0
          },
          "Details": {
              "DocumentName": "AWS-UpdateSSMAgent",
              "DocumentVersion": "1"
          }
      },
              ...
  ],
  "NextToken": "--token string truncated--"
}
```

列出特定執行個體和關聯 ID 的合規項目

此範例列出了指定執行個體和關聯 ID 的所有合規項目。

命令：

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance" --
filters "Key=ComplianceType,Values=Association,Type=EQUAL" "Key=Id,Values=e4c2ed6d-516f-41aa
aa2a-0123456789ab,Type=EQUAL"
```

列出執行個體在特定日期和時間之後的合規項目

此範例列出了執行個體在指定日期和時間之後的所有合規項目。

命令：

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance" --
filters "Key=ExecutionTime,Values=2019-02-18T16:00:00Z,Type=GREATER_THAN"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListComplianceItems。

list-compliance-summaries

以下程式碼範例顯示如何使用 list-compliance-summaries。

AWS CLI

列出所有合規類型的合規摘要

此範例列出了帳戶中所有合規類型的合規摘要。

命令：

```
aws ssm list-compliance-summaries
```

輸出：

```
{
  "ComplianceSummaryItems": [
      {
          "ComplianceType": "Association",
          "CompliantSummary": {
              "CompliantCount": 2,
              "SeveritySummary": {
                  "CriticalCount": 0,
                  "HighCount": 0,
```

```
                "MediumCount": 0,
                "LowCount": 0,
                "InformationalCount": 0,
                "UnspecifiedCount": 2
            }
        },
        "NonCompliantSummary": {
            "NonCompliantCount": 0,
            "SeveritySummary": {
                "CriticalCount": 0,
                "HighCount": 0,
                "MediumCount": 0,
                "LowCount": 0,
                "InformationalCount": 0,
                "UnspecifiedCount": 0
            }
        }
    },
    {
        "ComplianceType": "Patch",
        "CompliantSummary": {
            "CompliantCount": 1,
            "SeveritySummary": {
                "CriticalCount": 0,
                "HighCount": 0,
                "MediumCount": 0,
                "LowCount": 0,
                "InformationalCount": 0,
                "UnspecifiedCount": 1
            }
        },
        "NonCompliantSummary": {
            "NonCompliantCount": 1,
            "SeveritySummary": {
                "CriticalCount": 1,
                "HighCount": 0,
                "MediumCount": 0,
                "LowCount": 0,
                "InformationalCount": 0,
                "UnspecifiedCount": 0
            }
        }
    },
            ...
```

```
  ],
  "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

列出特定合規類型的合規摘要

此範例列出了「修補程式」合規類型的合規摘要。

命令：

```
aws ssm list-compliance-summaries --
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListComplianceSummaries。

list-document-metadata-history

以下程式碼範例顯示如何使用 list-document-metadata-history。

AWS CLI

範例：檢視變更範本的核准歷史記錄和狀態

下列list-document-metadata-history範例會傳回指定 Change Manager 變更範本的核准歷史記錄。

```
aws ssm list-document-metadata-history \
    --name MyChangeManageTemplate \
    --metadata DocumentReviews
```

輸出：

```
{
    "Name": "MyChangeManagerTemplate",
    "DocumentVersion": "1",
    "Author": "arn:aws:iam::111222333444;:user/JohnDoe",
    "Metadata": {
        "ReviewerResponse": [
            {
                "CreateTime": "2021-07-30T11:58:28.025000-07:00",
                "UpdatedTime": "2021-07-30T12:01:19.274000-07:00",
```

```
                "ReviewStatus": "APPROVED",
                "Comment": [
                    {
                        "Type": "COMMENT",
                        "Content": "I approve this template version"
                    }
                ],
                "Reviewer": "arn:aws:iam::111222333444;:user/ShirleyRodriguez"
            },
            {
                "CreateTime": "2021-07-30T11:58:28.025000-07:00",
                "UpdatedTime": "2021-07-30T11:58:28.025000-07:00",
                "ReviewStatus": "PENDING"
            }
        ]
    }
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的檢閱和核准或拒絕變更範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListDocumentMetadataHistory。

list-document-versions

以下程式碼範例顯示如何使用 `list-document-versions`。

AWS CLI

列出文件版本

下列 `list-document-versions` 範例列出了 Systems Manager 文件的所有版本。

```
aws ssm list-document-versions \
    --name "Example"
```

輸出：

```
{
    "DocumentVersions": [
        {
            "Name": "Example",
            "DocumentVersion": "1",
            "CreatedDate": 1583257938.266,
```

```
            "IsDefaultVersion": true,
            "DocumentFormat": "YAML",
            "Status": "Active"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Sending Commands that Use the Document Version Parameter。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListDocumentVersion。

`list-documents`

以下程式碼範例顯示如何使用 `list-documents`。

AWS CLI

範例 1：列出文件

下列 `list-documents` 範例列出了由請求帳戶擁有、以自訂標籤標記的文件。

```
aws ssm list-documents \
    --filters Key=Owner,Values=Self Key=tag:DocUse,Values=Testing
```

輸出：

```
{
    "DocumentIdentifiers": [
        {
            "Name": "Example",
            "Owner": "29884EXAMPLE",
            "PlatformTypes": [
                "Windows",
                "Linux"
            ],
            "DocumentVersion": "1",
            "DocumentType": "Automation",
            "SchemaVersion": "0.3",
            "DocumentFormat": "YAML",
            "Tags": [
                {
```

```
                    "Key": "DocUse",
                    "Value": "Testing"
                }
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 AWS Systems Manager Documents。

範例 2：列出共用文件

下列`list-documents`範例列出共用文件，包括非 擁有的私有共用文件 AWS。

```
aws ssm list-documents \
    --filters Key=Name,Values=sharedDocNamePrefix  Key=Owner,Values=Private
```

輸出：

```
{
    "DocumentIdentifiers": [
        {
            "Name": "Example",
            "Owner": "12345EXAMPLE",
            "PlatformTypes": [
                "Windows",
                "Linux"
            ],
            "DocumentVersion": "1",
            "DocumentType": "Command",
            "SchemaVersion": "0.3",
            "DocumentFormat": "YAML",
            "Tags": []
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 AWS Systems Manager Documents。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListDocuments。

list-inventory-entries

以下程式碼範例顯示如何使用 list-inventory-entries。

AWS CLI

範例 1：檢視執行個體的特定庫存類型項目

下列list-inventory-entries範例列出特定執行個體上 AWS：應用程式庫存類型的庫存項目。

```
aws ssm list-inventory-entries \
    --instance-id "i-1234567890abcdef0" \
    --type-name "AWS:Application"
```

輸出：

```
{
  "TypeName": "AWS:Application",
  "InstanceId": "i-1234567890abcdef0",
  "SchemaVersion": "1.1",
  "CaptureTime": "2019-02-15T12:17:55Z",
  "Entries": [
    {
      "Architecture": "i386",
      "Name": "Amazon SSM Agent",
      "PackageId": "{88a60be2-89a1-4df8-812a-80863c2a2b68}",
      "Publisher": "Amazon Web Services",
      "Version": "2.3.274.0"
    },
    {
      "Architecture": "x86_64",
      "InstalledTime": "2018-05-03T13:42:34Z",
      "Name": "AmazonCloudWatchAgent",
      "Publisher": "",
      "Version": "1.200442.0"
    }
  ]
}
```

範例 2：檢視指派給執行個體的自訂庫存項目

下列 list-inventory-entries 範例列出了指派給執行個體的自訂庫存項目。

```
aws ssm list-inventory-entries \
    --instance-id "i-1234567890abcdef0" \
    --type-name "Custom:RackInfo"
```

輸出：

```
{
  "TypeName": "Custom:RackInfo",
  "InstanceId": "i-1234567890abcdef0",
  "SchemaVersion": "1.0",
  "CaptureTime": "2021-05-22T10:01:01Z",
  "Entries": [
    {
      "RackLocation": "Bay B/Row C/Rack D/Shelf E"
    }
  ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListInventoryEntries。

list-ops-item-related-items

以下程式碼範例顯示如何使用 list-ops-item-related-items。

AWS CLI

列出 OpsItem 的相關項目資源

下列list-ops-item-related-items範例列出 OpsItem 的相關項目資源。

```
aws ssm list-ops-item-related-items \
    --ops-item-id "oi-f99f2EXAMPLE"
```

輸出：

```
{
    "Summaries": [
        {
            "OpsItemId": "oi-f99f2EXAMPLE",
            "AssociationId": "e2036148-cccb-490e-ac2a-390e5EXAMPLE",
            "ResourceType": "AWS::SSMIncidents::IncidentRecord",
            "AssociationType": "IsParentOf",
```

```
            "ResourceUri": "arn:aws:ssm-incidents::111122223333:incident-record/
example-response/64bd9b45-1d0e-2622-840d-03a87a1451fa",
            "CreatedBy": {
                "Arn": "arn:aws:sts::111122223333:assumed-role/
AWSServiceRoleForIncidentManager/IncidentResponse"
            },
            "CreatedTime": "2021-08-11T18:47:14.994000+00:00",
            "LastModifiedBy": {
                "Arn": "arn:aws:sts::111122223333:assumed-role/
AWSServiceRoleForIncidentManager/IncidentResponse"
            },
            "LastModifiedTime": "2021-08-11T18:47:14.994000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱 Systems Manager 使用者指南中的在 OpsCenter 中使用 Incident Manager 事件。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListOpsItemRelatedItems。

`list-resource-compliance-summaries`

以下程式碼範例顯示如何使用 `list-resource-compliance-summaries`。

AWS CLI

列出資源層級合規摘要計數

此範例列出了資源層級合規摘要計數。

命令：

```
aws ssm list-resource-compliance-summaries
```

輸出：

```
{
  "ResourceComplianceSummaryItems": [
      {
          "ComplianceType": "Association",
          "ResourceType": "ManagedInstance",
          "ResourceId": "i-1234567890abcdef0",
```

```
            "Status": "COMPLIANT",
            "OverallSeverity": "UNSPECIFIED",
            "ExecutionSummary": {
                "ExecutionTime": 1550509273.0
            },
            "CompliantSummary": {
                "CompliantCount": 2,
                "SeveritySummary": {
                    "CriticalCount": 0,
                    "HighCount": 0,
                    "MediumCount": 0,
                    "LowCount": 0,
                    "InformationalCount": 0,
                    "UnspecifiedCount": 2
                }
            },
            "NonCompliantSummary": {
                "NonCompliantCount": 0,
                "SeveritySummary": {
                    "CriticalCount": 0,
                    "HighCount": 0,
                    "MediumCount": 0,
                    "LowCount": 0,
                    "InformationalCount": 0,
                    "UnspecifiedCount": 0
                }
            }
        },
        {
            "ComplianceType": "Patch",
            "ResourceType": "ManagedInstance",
            "ResourceId": "i-9876543210abcdef0",
            "Status": "COMPLIANT",
            "OverallSeverity": "UNSPECIFIED",
            "ExecutionSummary": {
                "ExecutionTime": 1550248550.0,
                "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
                "ExecutionType": "Command"
            },
            "CompliantSummary": {
                "CompliantCount": 397,
                "SeveritySummary": {
                    "CriticalCount": 0,
                    "HighCount": 0,
```

```
                    "MediumCount": 0,
                    "LowCount": 0,
                    "InformationalCount": 0,
                    "UnspecifiedCount": 397
                }
            },
            "NonCompliantSummary": {
                "NonCompliantCount": 0,
                "SeveritySummary": {
                    "CriticalCount": 0,
                    "HighCount": 0,
                    "MediumCount": 0,
                    "LowCount": 0,
                    "InformationalCount": 0,
                    "UnspecifiedCount": 0
                }
            }
        }
    ],
    "NextToken": "--token string truncated--"
}
```

列出特定合規類型的資源層級合規摘要

此範例列出了「修補程式」合規類型的資源層級合規摘要。

命令：

```
aws ssm list-resource-compliance-summaries --
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListResourceComplianceSummaries。

list-resource-data-sync

以下程式碼範例顯示如何使用 list-resource-data-sync。

AWS CLI

列出您的資源資料同步組態

此範例會擷取資源資料同步組態的相關資訊。

```
aws ssm list-resource-data-sync
```

輸出：

```
{
    "ResourceDataSyncItems": [
        {
            "SyncName": "MyResourceDataSync",
            "S3Destination": {
                "BucketName": "ssm-resource-data-sync",
                "SyncFormat": "JsonSerDe",
                "Region": "us-east-1"
            },
            "LastSyncTime": 1550261472.003,
            "LastSuccessfulSyncTime": 1550261472.003,
            "LastStatus": "Successful",
            "SyncCreatedTime": 1543235736.72,
            "LastSyncStatusMessage": "The sync was successfully completed"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourceDataSync。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出套用至修補基準的標籤

下列 list-tags-for-resource 範例列出了修補基準的標籤。

```
aws ssm list-tags-for-resource \
    --resource-type "PatchBaseline" \
    --resource-id "pb-0123456789abcdef0"
```

輸出：

```
{
```

```
    "TagList": [
        {
            "Key": "Environment",
            "Value": "Production"
        },
        {
            "Key": "Region",
            "Value": "EMEA"
        }
    ]
}
```

如需詳細資訊，請參閱《 AWS 一般參考》中的標記 AWS 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`modify-document-permission`

以下程式碼範例顯示如何使用 `modify-document-permission`。

AWS CLI

修改文件許可

下列 `modify-document-permission` 範例公開共用 Systems Manager 文件。

```
aws ssm modify-document-permission \
    --name "Example" \
    --permission-type "Share" \
    --account-ids-to-add "All"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Share a Systems Manager Document。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ModifyDocumentPermission。

`put-compliance-items`

以下程式碼範例顯示如何使用 `put-compliance-items`。

AWS CLI

將合規類型和合規詳細資訊註冊至指定的執行個體

此範例示範將合規類型 Custom:AVCheck 註冊到指定的受管執行個體。如果命令成功，則無輸出訊息。

命令：

```
aws ssm put-compliance-items --resource-id "i-1234567890abcdef0" --
resource-type "ManagedInstance" --compliance-type "Custom:AVCheck"
 --execution-summary "ExecutionTime=2019-02-18T16:00:00Z" --
items "Id=Version2.0,Title=ScanHost,Severity=CRITICAL,Status=COMPLIANT"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutComplianceItems。

put-inventory

以下程式碼範例顯示如何使用 put-inventory。

AWS CLI

向執行個體指派客戶中繼資料

此範例會將機架位置資訊指派給執行個體。如果命令成功，則無輸出訊息。

命令 (Linux)：

```
aws ssm put-inventory --instance-id "i-016648b75dd622dab" --items
 '[{"TypeName": "Custom:RackInfo","SchemaVersion": "1.0","CaptureTime":
 "2019-01-22T10:01:01Z","Content":[{"RackLocation": "Bay B/Row C/Rack D/Shelf
 E"}]}]'
```

命令 (Windows)：

```
aws ssm put-inventory --instance-id "i-016648b75dd622dab" --
items "TypeName=Custom:RackInfo,SchemaVersion=1.0,CaptureTime=2019-01-22T10:01:01Z,Content=[
 B/Row C/Rack D/Shelf F'}]"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutInventory。

put-parameter

以下程式碼範例顯示如何使用 put-parameter。

AWS CLI

範例 1：變更參數值

下列 put-parameter 範例示範變更指定參數的值。

```
aws ssm put-parameter \
    --name "MyStringParameter" \
    --type "String" \
    --value "Vici" \
    --overwrite
```

輸出：

```
{
    "Version": 2,
    "Tier": "Standard"
}
```

如需詳細資訊，請參閱 Systems Manager 使用者指南中的建立 Systems Manager 參數 (AWS CLI)、管理參數層和使用參數政策。 AWS

範例 2：建立進階參數

下列 put-parameter 範例示範建立進階參數。

```
aws ssm put-parameter \
    --name "MyAdvancedParameter" \
    --description "This is an advanced parameter" \
    --value "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
 tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
 quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat
 [truncated]" \
    --type "String" \
    --tier Advanced
```

輸出：

```
{
    "Version": 1,
    "Tier": "Advanced"
}
```

如需詳細資訊，請參閱 Systems Manager 使用者指南中的建立 Systems Manager 參數 (AWS CLI)、管理參數層和使用參數政策。 AWS

範例 3：將標準參數轉換為進階參數

下列put-parameter範例會將現有的標準參數轉換為進階參數。

```
aws ssm put-parameter \
    --name "MyConvertedParameter" \
    --value "abc123" \
    --type "String" \
    --tier Advanced \
    --overwrite
```

輸出：

```
{
    "Version": 2,
    "Tier": "Advanced"
}
```

如需詳細資訊，請參閱 Systems Manager 使用者指南中的建立 Systems Manager 參數 (AWS CLI)、管理參數層和使用參數政策。 AWS

範例 4：建立已附加政策的參數

下列 put-parameter 範例示範建立已附加參數政策的進階參數。

```
aws ssm put-parameter \
    --name "/Finance/Payroll/q2accesskey" \
    --value "P@sSwW)rd" \
    --type "SecureString" \
    --tier Advanced \
    --policies "[{\"Type\":\"Expiration\",\"Version\":\"1.0\",\"Attributes\":
{\"Timestamp\":\"2020-06-30T00:00:00.000Z\"}},{\"Type\":\"ExpirationNotification\",
```

```
\"Version\":\"1.0\",\"Attributes\":{\"Before\":\"5\",\"Unit\":\"Days\"}},{\"Type\":
\"NoChangeNotification\",\"Version\":\"1.0\",\"Attributes\":{\"After\":\"60\",\"Unit
\":\"Days\"}}]"
```

輸出：

```
{
    "Version": 1,
    "Tier": "Advanced"
}
```

如需詳細資訊，請參閱《Systems Manager 使用者指南》中的建立 Systems Manager 參數 (AWS CLI)、管理參數層和使用參數政策。 AWS

範例 5：將政策新增至現有參數

下列 put-parameter 範例示範將政策連接至現有的進階參數。

```
aws ssm put-parameter \
    --name "/Finance/Payroll/q2accesskey" \
    --value "N3wP@sSwW)rd" \
    --type "SecureString" \
    --tier Advanced \
    --policies "[{\"Type\":\"Expiration\",\"Version\":\"1.0\",\"Attributes\":
{\"Timestamp\":\"2020-06-30T00:00:00.000Z\"}},{\"Type\":\"ExpirationNotification\",
\"Version\":\"1.0\",\"Attributes\":{\"Before\":\"5\",\"Unit\":\"Days\"}},{\"Type\":
\"NoChangeNotification\",\"Version\":\"1.0\",\"Attributes\":{\"After\":\"60\",\"Unit
\":\"Days\"}}]"
    --overwrite
```

輸出：

```
{
    "Version": 2,
    "Tier": "Advanced"
}
```

如需詳細資訊，請參閱《Systems Manager 使用者指南》中的建立 Systems Manager 參數 (AWS CLI)、管理參數層和使用參數政策。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutParameter。

register-default-patch-baseline

以下程式碼範例顯示如何使用 register-default-patch-baseline。

AWS CLI

設定預設的修補基準

下列 register-default-patch-baseline 範例示範將指定的自訂修補基準註冊為其支援的作業系統類型的預設修補基準。

```
aws ssm register-default-patch-baseline \
    --baseline-id "pb-abc123cf9bEXAMPLE"
```

輸出：

```
{
    "BaselineId":"pb-abc123cf9bEXAMPLE"
}
```

下列register-default-patch-baseline範例會將 AWS for CentOS 提供的預設修補程式基準註冊為預設修補程式基準。

```
aws ssm register-default-patch-baseline \
    --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-0574b43a65ea646ed"
```

輸出：

```
{
    "BaselineId":"pb-abc123cf9bEXAMPLE"
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 About Predefined and Custom Patch Baselines。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 RegisterDefaultPatchBaseline。

register-patch-baseline-for-patch-group

以下程式碼範例顯示如何使用 register-patch-baseline-for-patch-group。

AWS CLI

註冊修補程式群組的修補基準

下列 `register-patch-baseline-for-patch-group` 範例示範註冊修補程式群組的修補基準。

```
aws ssm register-patch-baseline-for-patch-group \
    --baseline-id "pb-045f10b4f382baeda" \
    --patch-group "Production"
```

輸出：

```
{
    "BaselineId": "pb-045f10b4f382baeda",
    "PatchGroup": "Production"
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Create a Patch Group <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html>__ 和 Add a Patch Group to a Patch Baseline。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 RegisterPatchBaselineForPatchGroup。

register-target-with-maintenance-window

以下程式碼範例顯示如何使用 `register-target-with-maintenance-window`。

AWS CLI

範例 1：向維護時段註冊單一目標

下列 `register-target-with-maintenance-window` 範例示範向維護時段註冊執行個體。

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --target "Key=InstanceIds,Values=i-0000293ffd8c57862" \
    --owner-information "Single instance" \
    --resource-type "INSTANCE"
```

輸出：

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

範例 2：使用執行個體 ID 向維護時段註冊多個目標

下列 register-target-with-maintenance-window 範例示範透過指定執行個體 ID，向維護時段註冊兩個執行個體。

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --target "Key=InstanceIds,Values=i-0000293ffd8c57862,i-0cb2b964d3e14fd9f" \
    --owner-information "Two instances in a list" \
    --resource-type "INSTANCE"
```

輸出：

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

範例 3：使用資源標籤向維護時段註冊目標

下列 register-target-with-maintenance-window 範例示範透過指定已套用至執行個體的資源標籤，向維護時段註冊執行個體。

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-06cf17cbefcb4bf4f" \
    --targets "Key=tag:Environment,Values=Prod" "Key=Role,Values=Web" \
    --owner-information "Production Web Servers" \
    --resource-type "INSTANCE"
```

輸出：

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

範例 4：使用一組標籤索引鍵註冊目標

下列 register-target-with-maintenance-window 範例示範註冊執行個體，這些執行個體擁有一個或多個標指派的標籤索引鍵，無論其索引鍵值為何。

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "INSTANCE" \
    --target "Key=tag-key,Values=Name,Instance-Type,CostCenter"
```

輸出：

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

範例 5：使用資源群組名稱註冊目標

下列 register-target-with-maintenance-window 範例示範註冊指定的資源群組，無論其中包含的資源類型為何。

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "RESOURCE_GROUP" \
    --target "Key=resource-groups:Name,Values=MyResourceGroup"
```

輸出：

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的向維護時段 (AWS CLI) 註冊目標執行個體。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 RegisterTargetWithMaintenanceWindow。

register-task-with-maintenance-window

以下程式碼範例顯示如何使用 register-task-with-maintenance-window。

AWS CLI

範例 1：向維護時段註冊 Automation 任務

下列 `register-task-with-maintenance-window` 範例示範向以執行個體為目標的維護時段註冊 Automation 任務。

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-082dcd7649EXAMPLE" \
    --targets Key=InstanceIds,Values=i-1234520122EXAMPLE \
    --task-arn AWS-RestartEC2Instance \
    --service-role-arn arn:aws:iam::111222333444:role/SSM --task-type AUTOMATION \
    --task-invocation-parameters "{\"Automation\":{\"DocumentVersion\":\"\$LATEST\",
\"Parameters\":{\"InstanceId\":[\"{{RESOURCE_ID}}\"]}}}" \
    --priority 0 \
    --max-concurrency 1 \
    --max-errors 1 \
    --name "AutomationExample" \
    --description "Restarting EC2 Instance for maintenance"
```

輸出：

```
{
    "WindowTaskId":"11144444-5555-6666-7777-88888888"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的向維護時段 (AWS CLI) 註冊任務。

範例 2：向維護時段註冊 Lambda 任務

下列 `register-task-with-maintenance-window` 範例示範向以執行個體為目標的維護時段註冊 Lambda 任務。

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-082dcd7649dee04e4" \
    --targets Key=InstanceIds,Values=i-12344d305eEXAMPLE \
    --task-arn arn:aws:lambda:us-east-1:111222333444:function:SSMTestLAMBDA \
    --service-role-arn arn:aws:iam::111222333444:role/SSM \
    --task-type LAMBDA \
    --task-invocation-parameters '{"Lambda":{"Payload":"{\"InstanceId\":
\"{{RESOURCE_ID}}\",\"targetType\":\"{{TARGET_TYPE}}\"}","Qualifier":"$LATEST"}}' \
    --priority 0 \
```

```
    --max-concurrency 10 \
    --max-errors 5 \
    --name "Lambda_Example" \
    --description "My Lambda Example"
```

輸出：

```
{
    "WindowTaskId":"22244444-5555-6666-7777-88888888"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的向維護時段 (AWS CLI) 註冊任務。

範例 3：向維護時段註冊 Run Command 任務

下列 `register-task-with-maintenance-window` 範例示範向以執行個體為目標的維護時段註冊 Run Command 任務。

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-082dcd7649dee04e4" \
    --targets "Key=InstanceIds,Values=i-12344d305eEXAMPLE" \
    --service-role-arn "arn:aws:iam::111222333444:role/SSM" \
    --task-type "RUN_COMMAND" \
    --name "SSMInstallPowerShellModule" \
    --task-arn "AWS-InstallPowerShellModule" \
    --task-invocation-parameters "{\"RunCommand\":{\"Comment\":\"\",
\"OutputS3BucketName\":\"runcommandlogs\",\"Parameters\":{\"commands\":[\"Get-
Module -ListAvailable\"],\"executionTimeout\":[\"3600\"],\"source\":[\"https:\/
\/gallery.technet.microsoft.com\/EZOut-33ae0fb7\/file\/110351\/1\/EZOut.zip\"],
\"workingDirectory\":[\"\\\\\"]},\"TimeoutSeconds\":600}}" \
    --max-concurrency 1 \
    --max-errors 1 \
    --priority 10
```

輸出：

```
{
    "WindowTaskId":"33344444-5555-6666-7777-88888888"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的向維護時段 (AWS CLI) 註冊任務。

範例 4：向維護時段註冊 Step Functions 任務

下列 register-task-with-maintenance-window 範例示範向以執行個體為目標的維護時段註冊 Step Functions 任務。

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-1234d787d6EXAMPLE" \
    --targets Key=WindowTargetIds,Values=12347414-69c3-49f8-95b8-ed2dcEXAMPLE \
    --task-arn arn:aws:states:us-
east-1:111222333444:stateMachine:SSMTestStateMachine \
    --service-role-arn arn:aws:iam::111222333444:role/MaintenanceWindows \
    --task-type STEP_FUNCTIONS \
    --task-invocation-parameters '{"StepFunctions":{"Input":"{\"InstanceId\":
\"{{RESOURCE_ID}}\"}"}}' \
    --priority 0 \
    --max-concurrency 10 \
    --max-errors 5 \
    --name "Step_Functions_Example" \
    --description "My Step Functions Example"
```

輸出：

```
{
    "WindowTaskId":"44444444-5555-6666-7777-88888888"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的向維護時段 (AWS CLI) 註冊任務。

範例 5：使用維護時段目標 ID 註冊任務

下列 register-task-with-maintenance-window 範例示範使用維護時段目標 ID 註冊任務。維護時段目標 ID 位於 aws ssm register-target-with-maintenance-window 命令的輸出中。您也可以從 aws ssm describe-maintenance-window-targets 命令的輸出中擷取它。

```
aws ssm register-task-with-maintenance-window \
    --targets "Key=WindowTargetIds,Values=350d44e6-28cc-44e2-951f-4b2c9EXAMPLE" \
    --task-arn "AWS-RunShellScript" \
    --service-role-arn "arn:aws:iam::111222333444:role/MaintenanceWindowsRole" \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --task-type "RUN_COMMAND" \
    --task-parameters  "{\"commands\":{\"Values\":[\"df\"]}}" \
    --max-concurrency 1 \
```

```
    --max-errors 1 \
    --priority 10
```

輸出：

```
{
    "WindowTaskId":"33344444-5555-6666-7777-88888888"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的向維護時段 (AWS CLI) 註冊任務。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 RegisterTaskWithMaintenanceWindow。

remove-tags-from-resource

以下程式碼範例顯示如何使用 remove-tags-from-resource。

AWS CLI

從修補基準中移除標籤

下列 remove-tags-from-resource 範例示範從修補基準中移除標籤。

```
aws ssm remove-tags-from-resource \
    --resource-type "PatchBaseline" \
    --resource-id "pb-0123456789abcdef0" \
    --tag-keys "Region"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《 AWS 一般參考》中的標記 AWS 資源。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 RemoveTagsFromResource。

reset-service-setting

以下程式碼範例顯示如何使用 reset-service-setting。

AWS CLI

重設參數存放區輸送量的服務設定

下列reset-service-setting範例會將指定區域中參數存放區輸送量的服務設定重設為不再使用增加的輸送量。

```
aws ssm reset-service-setting \
    --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-store/high-throughput-enabled
```

輸出：

```
{
    "ServiceSetting": {
        "SettingId": "/ssm/parameter-store/high-throughput-enabled",
        "SettingValue": "false",
        "LastModifiedDate": 1555532818.578,
        "LastModifiedUser": "System",
        "ARN": "arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-store/high-throughput-enabled",
        "Status": "Default"
    }
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的增加參數存放區輸送量。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetServiceSetting。

resume-session

以下程式碼範例顯示如何使用 resume-session。

AWS CLI

若要繼續 Session Manager 工作階段

此resume-session範例會在中斷連線後，繼續使用具有執行個體的 Session Manager 工作階段。請注意，若要使用此互動式命令，則需要在進行呼叫的用戶端機器上安裝 Session Manager 外掛程式。

```
aws ssm resume-session \
    --session-id Mary-Major-07a16060613c408b5
```

輸出：

```
{
    "SessionId": "Mary-Major-07a16060613c408b5",
    "TokenValue":
 "AAEAAVbTGsaOnyvcUoNGqifbv5r/8lgxuQljCuY8qVcvOnoBAAAAAFxtd3jIXAFUUXGTJ7zF/
AWJPwDviOlF5p3dlAgrqVIVO6IEXhkHLz0/1gXKRKEME71E6TLOplLDJAMZ
+kREejkZu4c5AxMkrQjMF+gtHP1bYJKTwtHQd1wjulPLexO8SHl7g5R/
wekrj6WsDUpnEegFBfGftpAIz2GXQVfTJXKfkc5qepQ11C11DOIT2dozOqXgHwfQHfAKLErM5dWDZqKwyT1Z3iw7unQd
+ihfGa6MEJJ97Jmat/a2TspEnOjNn9Mvu5iwXIW2yCvWZrGUj+/
QI5Xr7s1XJBEnSKR54o4fN0GV9RWl0RZsZm1m1ki0JJtiwwgZ",
    "StreamUrl": "wss://ssmmessages.us-east-2.amazonaws.com/v1/data-channel/Mary-
Major-07a16060613c408b5?role=publish_subscribe"
}
```

如需詳細資訊，請參閱 Systems Manager 使用者指南中的安裝 CLI AWS 的 Session Manager 外掛程式。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResumeSession。

send-automation-signal

以下程式碼範例顯示如何使用 send-automation-signal。

AWS CLI

將訊號傳送至自動化執行

下列send-automation-signal範例會將核准訊號傳送至自動化執行。

```
aws ssm send-automation-signal \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE \
    --signal-type "Approve"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的使用核准者執行自動化工作流程。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 SendAutomationSignal。

send-command

以下程式碼範例顯示如何使用 send-command。

AWS CLI

範例 1：在一個或多個遠端執行個體上執行命令

下列 send-command 範例示範在目標執行個體上執行 echo 命令。

```
aws ssm send-command \
    --document-name "AWS-RunShellScript" \
    --parameters 'commands=["echo HelloWorld"]' \
    --targets "Key=instanceids,Values=i-1234567890abcdef0" \
    --comment "echo HelloWorld"
```

輸出：

```
{
    "Command": {
        "CommandId": "92853adf-ba41-4cd6-9a88-142d1EXAMPLE",
        "DocumentName": "AWS-RunShellScript",
        "DocumentVersion": "",
        "Comment": "echo HelloWorld",
        "ExpiresAfter": 1550181014.717,
        "Parameters": {
            "commands": [
                "echo HelloWorld"
            ]
        },
        "InstanceIds": [
            "i-0f00f008a2dcbefe2"
        ],
        "Targets": [],
        "RequestedDateTime": 1550173814.717,
        "Status": "Pending",
        "StatusDetails": "Pending",
        "OutputS3BucketName": "",
        "OutputS3KeyPrefix": "",
        "MaxConcurrency": "50",
        "MaxErrors": "0",
        "TargetCount": 1,
        "CompletedCount": 0,
        "ErrorCount": 0,
        "DeliveryTimedOutCount": 0,
        "ServiceRole": "",
        "NotificationConfig": {
```

```
            "NotificationArn": "",
            "NotificationEvents": [],
            "NotificationType": ""
        },
        "CloudWatchOutputConfig": {
            "CloudWatchLogGroupName": "",
            "CloudWatchOutputEnabled": false
        }
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Running Commands Using Systems Manager Run Command。

範例 2：取得有關執行個體的 IP 資訊

以下 send-command 範例示範擷取有關執行個體的 IP 資訊。

```
aws ssm send-command \
    --instance-ids "i-1234567890abcdef0" \
    --document-name "AWS-RunShellScript" \
    --comment "IP config" \
    --parameters "commands=ifconfig"
```

如需範例輸出，請參閱範例 1。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Running Commands Using Systems Manager Run Command。

範例 3：在具有特定標籤的執行個體上執行命令

下列 send-command 範例示範在具有標籤索引鍵 "ENV" 和值 "Dev" 的執行個體上執行命令。

```
aws ssm send-command \
    --targets "Key=tag:ENV,Values=Dev" \
    --document-name "AWS-RunShellScript" \
    --parameters "commands=ifconfig"
```

如需範例輸出，請參閱範例 1。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Running Commands Using Systems Manager Run Command。

範例 4：執行傳送 SNS 通知的命令

下列 send-command 範例示範執行傳送所有通知事件和 Command 通知類型之 SNS 通知的命令。

```
aws ssm send-command \
    --instance-ids "i-1234567890abcdef0" \
    --document-name "AWS-RunShellScript" \
    --comment "IP config" \
    --parameters "commands=ifconfig" \
    --service-role-arn "arn:aws:iam::123456789012:role/SNS_Role" \
    --notification-config "NotificationArn=arn:aws:sns:us-
east-1:123456789012:SNSTopicName,NotificationEvents=All,NotificationType=Command"
```

如需範例輸出，請參閱範例 1。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Running Commands Using Systems Manager Run Command。

範例 5：執行輸出至 S3 和 CloudWatch 的命令

下列 send-command 範例示範執行命令，該命令將命令詳細資訊輸出至 S3 儲存貯體和 CloudWatch Logs 日誌群組。

```
aws ssm send-command \
    --instance-ids "i-1234567890abcdef0" \
    --document-name "AWS-RunShellScript" \
    --comment "IP config" \
    --parameters "commands=ifconfig" \
    --output-s3-bucket-name "s3-bucket-name" \
    --output-s3-key-prefix "runcommand" \
    --cloud-watch-output-
config "CloudWatchOutputEnabled=true,CloudWatchLogGroupName=CWLGroupName"
```

如需範例輸出，請參閱範例 1。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Running Commands Using Systems Manager Run Command。

範例 6：在具有不同標籤的多個執行個體上執行命令

下列 send-command 範例示範在具有兩個不同標籤索引鍵和值的執行個體上執行命令。

```
aws ssm send-command \
    --document-name "AWS-RunPowerShellScript" \
    --parameters commands=["echo helloWorld"] \
    --targets Key=tag:Env,Values=Dev Key=tag:Role,Values=WebServers
```

如需範例輸出，請參閱範例 1。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Running Commands Using Systems Manager Run Command。

範例 7：將具有相同標籤索引鍵的多個執行個體設定為目標

下列 send-command 範例示範在具有相同標籤索引鍵但值不同的執行個體上執行命令。

```
aws ssm send-command \
    --document-name "AWS-RunPowerShellScript" \
    --parameters commands=["echo helloWorld"] \
    --targets Key=tag:Env,Values=Dev,Test
```

如需範例輸出，請參閱範例 1。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Running Commands Using Systems Manager Run Command。

範例 8：執行使用共用文件的命令

下列 send-command 範例示範在目標執行個體上執行共用文件。

```
aws ssm send-command \
    --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument" \
    --targets "Key=instanceids,Values=i-1234567890abcdef0"
```

如需範例輸出，請參閱範例 1。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Using shared SSM documents。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 SendCommand。

start-associations-once

以下程式碼範例顯示如何使用 start-associations-once。

AWS CLI

立即執行關聯且僅執行一次

下列start-associations-once範例會立即且僅執行一次指定的關聯。如果命令成功，則無輸出訊息。

```
aws ssm start-associations-once \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Viewing association histories。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartAssociationsOnce。

start-automation-execution

以下程式碼範例顯示如何使用 start-automation-execution。

AWS CLI

範例 1：執行 Automation 文件

下列 start-automation-execution 範例示範執行 Automation 文件。

```
aws ssm start-automation-execution \
    --document-name "AWS-UpdateLinuxAmi" \
    --parameters "AutomationAssumeRole=arn:aws:iam::123456789012:role/
SSMAutomationRole,SourceAmiId=ami-EXAMPLE,IamInstanceProfileName=EC2InstanceRole"
```

輸出：

```
{
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Running an Automation Workflow Manually。

範例 2：執行共用 Automation 文件

下列 start-automation-execution 範例示範執行共用的 Automation 文件。

```
aws ssm start-automation-execution \
    --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument"
```

輸出：

```
{
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Using shared SSM documents。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 StartAutomationExecution。

start-change-request-execution

以下程式碼範例顯示如何使用 start-change-request-execution。

AWS CLI

範例 1：啟動變更請求

下列start-change-request-execution範例會以指定的最小選項啟動變更請求。

```
aws ssm start-change-request-execution \
    --change-request-name MyChangeRequest \
    --document-name AWS-HelloWorldChangeTemplate \
    --runbooks '[{"DocumentName": "AWS-HelloWorld","Parameters":
 {"AutomationAssumeRole": ["arn:aws:iam:us-east-2:1112223233444:role/
MyChangeManagerAssumeRole"]}}]' \
    --parameters
 Approver="JohnDoe",ApproverType="IamUser",ApproverSnsTopicArn="arn:aws:sns:us-
east-2:1112223233444:MyNotificationTopic"
```

輸出：

```
{
  "AutomationExecutionId": "9d32a4fc-f944-11e6-4105-0a1b2EXAMPLE"
}
```

範例 2：使用外部 JSON 檔案啟動變更請求

下列start-automation-execution範例使用 JSON 檔案中指定的多個選項啟動變更請求。

```
aws ssm start-change-request-execution \
    --cli-input-json file://MyChangeRequest.json
```

MyChangeRequest.json 的內容：

```
{
    "ChangeRequestName": "MyChangeRequest",
    "DocumentName": "AWS-HelloWorldChangeTemplate",
    "DocumentVersion": "$DEFAULT",
    "ScheduledTime": "2021-12-30T03:00:00",
    "ScheduledEndTime": "2021-12-30T03:05:00",
    "Tags": [
        {
            "Key": "Purpose",
            "Value": "Testing"
        }
    ],
    "Parameters": {
        "Approver": [
            "JohnDoe"
        ],
        "ApproverType": [
            "IamUser"
        ],
        "ApproverSnsTopicArn": [
            "arn:aws:sns:us-east-2:111222333444;:MyNotificationTopic
        ]
    },
    "Runbooks": [
        {
            "DocumentName": "AWS-HelloWorld",
            "DocumentVersion": "1",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Parameters": {
                "AutomationAssumeRole": [
                    "arn:aws:iam::111222333444:role/MyChangeManagerAssumeRole"
                ]
            }
```

```
        }
    ],
    "ChangeDetails": "### Document Name: HelloWorldChangeTemplate\n\n## What does
 this document do?\nThis change template demonstrates the feature set available
 for creating change templates for Change Manager. This template starts a Runbook
 workflow for the Automation document called AWS-HelloWorld.\n\n## Input Parameters
\n* ApproverSnsTopicArn: (Required) Amazon Simple Notification Service ARN for
 approvers.\n* Approver: (Required) The name of the approver to send this request
 to.\n* ApproverType: (Required) The type of reviewer.\n  * Allowed Values: IamUser,
 IamGroup, IamRole, SSOGroup, SSOUser\n\n## Output Parameters\nThis document has no
 outputs \n"
}
```

輸出：

```
{
  "AutomationExecutionId": "9d32a4fc-f944-11e6-4105-0a1b2EXAMPLE"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的建立變更請求。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartChangeRequestExecution。

start-session

以下程式碼範例顯示如何使用 start-session。

AWS CLI

範例 1：啟動 Session Manager 工作階段

此 start-session 範例示範建立與 Session Manager 工作階段執行個體之間的連線。請注意，若要使用此互動式命令，則需要在進行呼叫的用戶端機器上安裝 Session Manager 外掛程式。

```
aws ssm start-session \
    --target "i-1234567890abcdef0"
```

輸出：

```
Starting session with SessionId: Jane-Roe-07a16060613c408b5
```

範例 2：使用 SSH 啟動 Session Manager 工作階段

此 `start-session` 範例示範使用 SSH 建立與 Session Manager 工作階段執行個體之間的連線。請注意，若要使用此互動式命令，則需要在進行呼叫的用戶端機器上安裝 Session Manager 外掛程式；而且該命令會使用執行個體上的預設使用者，例如 Linux EC2 執行個體的 `ec2-user`。

```
ssh -i /path/my-key-pair.pem ec2-user@i-02573cafcfEXAMPLE
```

輸出：

```
Starting session with SessionId: ec2-user-07a16060613c408b5
```

如需詳細資訊，請參閱 Systems Manager 使用者指南中的啟動工作階段和安裝 CLI 的 Session Manager 外掛程式。 AWS AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 StartSession。

`stop-automation-execution`

以下程式碼範例顯示如何使用 `stop-automation-execution`。

AWS CLI

停止自動化執行

下列 `stop-automation-execution` 範例示範停止執行 Automation 文件。

```
aws ssm stop-automation-execution
    --automation-execution-id "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Running an Automation Workflow Manually。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 StopAutomationExecution。

`terminate-session`

以下程式碼範例顯示如何使用 `terminate-session`。

AWS CLI

結束 Session Manager 工作階段

此terminate-session範例會永久結束使用者「Shirley-Rodriguez」建立的工作階段，並關閉執行個體上 Session Manager 用戶端和 SSM Agent 之間的資料連線。

```
aws ssm terminate-session \
    --session-id "Shirley-Rodriguez-07a16060613c408b5"
```

輸出：

```
{
    "SessionId": "Shirley-Rodriguez-07a16060613c408b5"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的終止工作階段。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TerminateSession。

unlabel-parameter-version

以下程式碼範例顯示如何使用 unlabel-parameter-version。

AWS CLI

刪除參數標籤

下列unlabel-parameter-version範例會從指定的參數版本中刪除指定的標籤。

```
aws ssm unlabel-parameter-version \
    --name "parameterName" \
    --parameter-version "version" \
    --labels "label_1" "label_2" "label_3"
```

輸出：

```
{
    "RemovedLabels": [
        "label_1"
        "label_2"
```

```
        "label_3"
    ],
    "InvalidLabels": []
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的刪除參數標籤 (AWS CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UnlabelParameterVersion。

update-association-status

以下程式碼範例顯示如何使用 update-association-status。

AWS CLI

更新關聯狀態

下列 update-association-status 範例示範更新執行個體與文件之間的關聯狀態。

```
aws ssm update-association-status \
    --name "AWS-UpdateSSMAgent" \
    --instance-id "i-1234567890abcdef0" \
    --association-
status "Date=1424421071.939,Name=Pending,Message=temp_status_change,AdditionalInfo=Additiona
Config-Needed"
```

輸出：

```
{
    "AssociationDescription": {
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-1234567890abcdef0",
        "AssociationVersion": "1",
        "Date": 1550507529.604,
        "LastUpdateAssociationDate": 1550507806.974,
        "Status": {
            "Date": 1424421071.0,
            "Name": "Pending",
            "Message": "temp_status_change",
            "AdditionalInfo": "Additional-Config-Needed"
        },
        "Overview": {
            "Status": "Success",
```

```
            "AssociationStatusAggregatedCount": {
                "Success": 1
            }
        },
        "DocumentVersion": "$DEFAULT",
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
        "Targets": [
            {
                "Key": "InstanceIds",
                "Values": [
                    "i-1234567890abcdef0"
                ]
            }
        ],
        "LastExecutionDate": 1550507808.0,
        "LastSuccessfulExecutionDate": 1550507808.0
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with associations in Systems Manager。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateAssociationStatus。

update-association

以下程式碼範例顯示如何使用 update-association。

AWS CLI

範例 1：更新文件關聯

下列 update-association 範例示範以新文件版本更新關聯。

```
aws ssm update-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --document-version "\$LATEST"
```

輸出：

```
{
    "AssociationDescription": {
```

```
        "Name": "AWS-UpdateSSMAgent",
        "AssociationVersion": "2",
        "Date": 1550508093.293,
        "LastUpdateAssociationDate": 1550508106.596,
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "DocumentVersion": "$LATEST",
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
        "Targets": [
            {
                "Key": "tag:Name",
                "Values": [
                    "Linux"
                ]
            }
        ],
        "LastExecutionDate": 1550508094.879,
        "LastSuccessfulExecutionDate": 1550508094.879
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Editing and creating a new version of an association。

範例 2：更新關聯的排程表達式

下列 update-association 範例示範更新指定關聯的排程表達式。

```
aws ssm update-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --schedule-expression "cron(0 0 0/4 1/1 * ? *)"
```

輸出：

```
{
    "AssociationDescription": {
        "Name": "AWS-HelloWorld",
        "AssociationVersion": "2",
        "Date": "2021-02-08T13:54:19.203000-08:00",
        "LastUpdateAssociationDate": "2021-06-29T11:51:07.933000-07:00",
        "Overview": {
```

```
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "DocumentVersion": "$DEFAULT",
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
        "Targets": [
            {
                "Key": "aws:NoOpAutomationTag",
                "Values": [
                    "AWS-NoOpAutomationTarget-Value"
                ]
            }
        ],
        "ScheduleExpression": "cron(0 0 0/4 1/1 * ? *)",
        "LastExecutionDate": "2021-06-26T19:00:48.110000-07:00",
        "ApplyOnlyAtCronInterval": false
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Editing and creating a new version of an association。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateAssociation。

update-document-default-version

以下程式碼範例顯示如何使用 update-document-default-version。

AWS CLI

更新文件的預設版本

下列 update-document-default-version 範例示範更新 Systems Manager 文件的預設版本。

```
aws ssm update-document-default-version \
    --name "Example" \
    --document-version "2"
```

輸出：

```
{
```

```
    "Description": {
        "Name": "Example",
        "DefaultVersion": "2"
    }
}
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Writing SSM Document Content。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateDocumentDefaultVersion。

update-document-metadata

以下程式碼範例顯示如何使用 update-document-metadata。

AWS CLI

範例：核准最新版本的變更範本

以下update-document-metadata提供已提交審核之變更範本最新版本的核准。

```
aws ssm update-document-metadata \
    --name MyChangeManagerTemplate \
    --document-reviews 'Action=Approve,Comment=[{Type=Comment,Content=Approved!}]'
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的檢閱和核准或拒絕變更範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDocumentMetadata。

update-document

以下程式碼範例顯示如何使用 update-document。

AWS CLI

建立新版本的文件

下列 update-document 範例示範在 Windows 電腦上執行時建立新版本的文件。--document 指定的文件必須為 JSON 格式。請注意，必須參考 file://，後面接著內容檔案的路徑。由於 --

document-version 參數開頭的 $，在 Windows 上，必須使用雙引號括住值。在 Linux、MacOS 或 PowerShell 提示中，必須使用單引號括住值。

Windows 版本：

```
aws ssm update-document \
    --name "RunShellScript" \
    --content "file://RunShellScript.json" \
    --document-version "$LATEST"
```

Linux/Mac 版本：

```
aws ssm update-document \
    --name "RunShellScript" \
    --content "file://RunShellScript.json" \
    --document-version '$LATEST'
```

輸出：

```
{
  "DocumentDescription": {
      "Status": "Updating",
      "Hash": "f775e5df4904c6fa46686c4722fae9de1950dace25cd9608ff8d622046b68d9b",
      "Name": "RunShellScript",
      "Parameters": [
          {
              "Type": "StringList",
              "Name": "commands",
              "Description": "(Required) Specify a shell script or a command to
 run."
          }
      ],
      "DocumentType": "Command",
      "PlatformTypes": [
          "Linux"
      ],
      "DocumentVersion": "2",
      "HashType": "Sha256",
      "CreatedDate": 1487899655.152,
      "Owner": "809632081692",
      "SchemaVersion": "2.0",
      "DefaultVersion": "1",
```

```
      "LatestVersion": "2",
      "Description": "Run an updated script"
  }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateDocument。

update-maintenance-window-target

以下程式碼範例顯示如何使用 update-maintenance-window-target。

AWS CLI

更新維護時段目標

下列update-maintenance-window-target範例只會更新維護時段目標的名稱。

```
aws ssm update-maintenance-window-target \
    --window-id "mw-0c5ed765acEXAMPLE" \
    --window-target-id "57e8344e-fe64-4023-8191-6bf05EXAMPLE" \
    --name "NewName" \
    --no-replace
```

輸出：

```
{
    "Description": "",
    "OwnerInformation": "",
    "WindowTargetId": "57e8344e-fe64-4023-8191-6bf05EXAMPLE",
    "WindowId": "mw-0c5ed765acEXAMPLE",
    "Targets": [
        {
            "Values": [
                "i-1234567890EXAMPLE"
            ],
            "Key": "InstanceIds"
        }
    ],
    "Name": "NewName"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的更新維護時段 (AWS CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateMaintenanceWindowTarget。

update-maintenance-window-task

以下程式碼範例顯示如何使用 update-maintenance-window-task。

AWS CLI

更新維護時段任務

下列update-maintenance-window-task範例會更新維護時段任務的服務角色。

```
aws ssm update-maintenance-window-task \
    --window-id "mw-0c5ed765acEXAMPLE" \
    --window-task-id "23d3809e-9fbe-4ddf-b41a-b49d7EXAMPLE" \
    --service-role-arn "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM"
```

輸出：

```
{
    "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
    "MaxErrors": "1",
    "TaskArn": "AWS-UpdateEC2Config",
    "MaxConcurrency": "1",
    "WindowTaskId": "23d3809e-9fbe-4ddf-b41a-b49d7EXAMPLE",
    "TaskParameters": {},
    "Priority": 1,
    "TaskInvocationParameters": {
        "RunCommand": {
            "TimeoutSeconds": 600,
            "Parameters": {
                "allowDowngrade": [
                    "false"
                ]
            }
        }
    },
    "WindowId": "mw-0c5ed765acEXAMPLE",
    "Description": "UpdateEC2Config",
    "Targets": [
        {
```

```
            "Values": [
                "57e8344e-fe64-4023-8191-6bf05EXAMPLE"
            ],
            "Key": "WindowTargetIds"
        }
    ],
    "Name": "UpdateEC2Config"
}
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的更新維護時段 (AWS CLI)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateMaintenanceWindowTask。

update-maintenance-window

以下程式碼範例顯示如何使用 update-maintenance-window。

AWS CLI

範例 1：更新維護時段

下列 update-maintenance-window 範例示範更新維護時段的名稱。

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --name "My-Renamed-MW"
```

輸出：

```
{
    "Cutoff": 1,
    "Name": "My-Renamed-MW",
    "Schedule": "cron(0 16 ? * TUE *)",
    "Enabled": true,
    "AllowUnassociatedTargets": true,
    "WindowId": "mw-1a2b3c4d5e6f7g8h9",
    "Duration": 4
}
```

範例 2：停用維護時段

下列 update-maintenance-window 範例示範停用維護時段。

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --no-enabled
```

範例 3：啟用維護時段

下列 update-maintenance-window 範例示範啟用維護時段。

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --enabled
```

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的更新維護時段 (AWS CLI)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateMaintenanceWindow。

update-managed-instance-role

以下程式碼範例顯示如何使用 update-managed-instance-role。

AWS CLI

更新受管執行個體的 IAM 角色

下列 update-managed-instance-role 範例示範更新受管執行個體的 IAM 執行個體設定檔。

```
aws ssm update-managed-instance-role \
    --instance-id "mi-08ab247cdfEXAMPLE" \
    --iam-role "ExampleRole"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Step 4: Create an IAM Instance Profile for Systems Manager。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateManagedInstanceRole。

update-ops-item

以下程式碼範例顯示如何使用 update-ops-item。

AWS CLI

更新 OpsItem

下列 update-ops-item 範例示範更新 OpsItem 的描述、優先順序和類別。此外，該命令還指定 SNS 主題，在編輯或變更此 OpsItem 時會傳送通知。

```
aws ssm update-ops-item \
    --ops-item-id "oi-287b5EXAMPLE" \
    --description "Primary OpsItem for failover event 2020-01-01-fh398yf" \
    --priority 2 \
    --category "Security" \
    --notifications "Arn=arn:aws:sns:us-east-2:111222333444:my-us-east-2-topic"
```

輸出：

```
This command produces no output.
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Working with OpsItems。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateOpsItem。

update-patch-baseline

以下程式碼範例顯示如何使用 update-patch-baseline。

AWS CLI

範例 1：更新修補基準

下列 update-patch-baseline 範例示範將指定修補程式新增到指定修補基準，兩個修補程式新增為已拒絕，一個修補程式新增為已核准。

```
aws ssm update-patch-baseline \
        --baseline-id "pb-0123456789abcdef0" \
        --rejected-patches "KB2032276" "MS10-048" \
        --approved-patches "KB2124261"
```

輸出：

```
{
```

```
    "BaselineId": "pb-0123456789abcdef0",
    "Name": "WindowsPatching",
    "OperatingSystem": "WINDOWS",
    "GlobalFilters": {
        "PatchFilters": []
    },
    "ApprovalRules": {
        "PatchRules": [
            {
                "PatchFilterGroup": {
                    "PatchFilters": [
                        {
                            "Key": "PRODUCT",
                            "Values": [
                                "WindowsServer2016"
                            ]
                        }
                    ]
                },
                "ComplianceLevel": "CRITICAL",
                "ApproveAfterDays": 0,
                "EnableNonSecurity": false
            }
        ]
    },
    "ApprovedPatches": [
        "KB2124261"
    ],
    "ApprovedPatchesComplianceLevel": "UNSPECIFIED",
    "ApprovedPatchesEnableNonSecurity": false,
    "RejectedPatches": [
        "KB2032276",
        "MS10-048"
    ],
    "RejectedPatchesAction": "ALLOW_AS_DEPENDENCY",
    "CreatedDate": 1550244180.465,
    "ModifiedDate": 1550244180.465,
    "Description": "Patches for Windows Servers",
    "Sources": []
}
```

範例 2：重新命名修補基準

下列 `update-patch-baseline` 範例示範重新命名指定的修補基準。

```
aws ssm update-patch-baseline \
    --baseline-id "pb-0713accee01234567" \
    --name "Windows-Server-2012-R2-Important-and-Critical-Security-Updates"
```

如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Update or Delete a Patch Baseline <https://docs.aws.amazon.com/systems-manager/latest/userguide/patch-baseline-update-or-delete.html>`__。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdatePatchBaseline。

update-resource-data-sync

以下程式碼範例顯示如何使用 update-resource-data-sync。

AWS CLI

更新資源資料同步

下列update-resource-data-sync範例會更新 SyncFromSource 資源資料同步。

```
aws ssm update-resource-data-sync \
    --sync-name exampleSync \
    --sync-type SyncFromSource \
    --sync-source '{"SourceType":"SingleAccountMultiRegions", "SourceRegions":["us-east-1", "us-west-2"]}'
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Systems Manager 使用者指南》中的設定 Systems Manager Explorer 以顯示來自多個帳戶和區域的資料。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateResourceDataSync。

update-service-setting

以下程式碼範例顯示如何使用 update-service-setting。

AWS CLI

更新參數存放區輸送量的服務設定

下列update-service-setting範例會更新指定區域中參數存放區輸送量的目前服務設定，以使用增加的輸送量。

```
aws ssm update-service-setting \
    --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-store/high-throughput-enabled \
    --setting-value true
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的增加參數存放區輸送量。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateServiceSetting。

使用 的 Amazon Textract 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon Textract 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`analyze-document`

以下程式碼範例顯示如何使用 `analyze-document`。

AWS CLI

分析文件中的文字

下列analyze-document範例顯示如何分析文件中的文字。

Linux/macOS：

```
aws textract analyze-document \
```

```
    --document '{"S3Object":{"Bucket":"bucket","Name":"document"}}' \
    --feature-types '["TABLES","FORMS"]'
```

Windows：

```
aws textract analyze-document \
    --document "{\"S3Object\":{\"Bucket\":\"bucket\",\"Name\":\"document\"}}" \
    --feature-types "[\"TABLES\",\"FORMS\"]" \
    --region region-name
```

輸出：

```
{
    "Blocks": [
        {
            "Geometry": {
                "BoundingBox": {
                    "Width": 1.0,
                    "Top": 0.0,
                    "Left": 0.0,
                    "Height": 1.0
                },
                "Polygon": [
                    {
                        "Y": 0.0,
                        "X": 0.0
                    },
                    {
                        "Y": 0.0,
                        "X": 1.0
                    },
                    {
                        "Y": 1.0,
                        "X": 1.0
                    },
                    {
                        "Y": 1.0,
                        "X": 0.0
                    }
                ]
            },
            "Relationships": [
                {
```

```
                    "Type": "CHILD",
                    "Ids": [
                        "87586964-d50d-43e2-ace5-8a890657b9a0",
                        "a1e72126-21d9-44f4-a8d6-5c385f9002ba",
                        "e889d012-8a6b-4d2e-b7cd-7a8b327d876a"
                    ]
                }
            ],
            "BlockType": "PAGE",
            "Id": "c2227f12-b25d-4e1f-baea-1ee180d926b2"
        }
    ],
    "DocumentMetadata": {
        "Pages": 1
    }
}
```

如需詳細資訊，請參閱《Amazon Textract 開發人員指南》中的使用 Amazon Textract 分析文件文字

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AnalyzeDocument。

detect-document-text

以下程式碼範例顯示如何使用 detect-document-text。

AWS CLI

若要偵測文件中的文字

下列detect-document-text範例示範如何偵測文件中的文字。

Linux/macOS：

```
aws textract detect-document-text \
    --document '{"S3Object":{"Bucket":"bucket","Name":"document"}}'
```

Windows：

```
aws textract detect-document-text \
    --document "{\"S3Object\":{\"Bucket\":\"bucket\",\"Name\":\"document\"}}" \
    --region region-name
```

輸出：


```
{
    "Blocks": [
        {
            "Geometry": {
                "BoundingBox": {
                    "Width": 1.0,
                    "Top": 0.0,
                    "Left": 0.0,
                    "Height": 1.0
                },
                "Polygon": [
                    {
                        "Y": 0.0,
                        "X": 0.0
                    },
                    {
                        "Y": 0.0,
                        "X": 1.0
                    },
                    {
                        "Y": 1.0,
                        "X": 1.0
                    },
                    {
                        "Y": 1.0,
                        "X": 0.0
                    }
                ]
            },
            "Relationships": [
                {
                    "Type": "CHILD",
                    "Ids": [
                        "896a9f10-9e70-4412-81ce-49ead73ed881",
                        "0da18623-dc4c-463d-a3d1-9ac050e9e720",
                        "167338d7-d38c-4760-91f1-79a8ec457bb2"
                    ]
                }
            ],
            "BlockType": "PAGE",
            "Id": "21f0535e-60d5-4bc7-adf2-c05dd851fa25"
        },
```

```
        {
            "Relationships": [
                {
                    "Type": "CHILD",
                    "Ids": [
                        "62490c26-37ea-49fa-8034-7a9ff9369c9c",
                        "1e4f3f21-05bd-4da9-ba10-15d01e66604c"
                    ]
                }
            ],
            "Confidence": 89.11581420898438,
            "Geometry": {
                "BoundingBox": {
                    "Width": 0.33642634749412537,
                    "Top": 0.17169663310050964,
                    "Left": 0.13885067403316498,
                    "Height": 0.49159330129623413
                },
                "Polygon": [
                    {
                        "Y": 0.17169663310050964,
                        "X": 0.13885067403316498
                    },
                    {
                        "Y": 0.17169663310050964,
                        "X": 0.47527703642845154
                    },
                    {
                        "Y": 0.6632899641990662,
                        "X": 0.47527703642845154
                    },
                    {
                        "Y": 0.6632899641990662,
                        "X": 0.13885067403316498
                    }
                ]
            },
            "Text": "He llo,",
            "BlockType": "LINE",
            "Id": "896a9f10-9e70-4412-81ce-49ead73ed881"
        },
        {
            "Relationships": [
                {
```

```
                    "Type": "CHILD",
                    "Ids": [
                        "19b28058-9516-4352-b929-64d7cef29daf"
                    ]
                }
            ],
            "Confidence": 85.5694351196289,
            "Geometry": {
                "BoundingBox": {
                    "Width": 0.33182239532470703,
                    "Top": 0.23131252825260162,
                    "Left": 0.5091826915740967,
                    "Height": 0.3766750991344452
                },
                "Polygon": [
                    {
                        "Y": 0.23131252825260162,
                        "X": 0.5091826915740967
                    },
                    {
                        "Y": 0.23131252825260162,
                        "X": 0.8410050868988037
                    },
                    {
                        "Y": 0.607987642288208,
                        "X": 0.8410050868988037
                    },
                    {
                        "Y": 0.607987642288208,
                        "X": 0.5091826915740967
                    }
                ]
            },
            "Text": "worlc",
            "BlockType": "LINE",
            "Id": "0da18623-dc4c-463d-a3d1-9ac050e9e720"
        }
    ],
    "DocumentMetadata": {
        "Pages": 1
    }
}
```

如需詳細資訊，請參閱《Amazon Textract 開發人員指南》中的使用 Amazon Textract 偵測文件文字

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DetectDocumentText。

get-document-analysis

以下程式碼範例顯示如何使用 get-document-analysis。

AWS CLI

取得多頁文件的非同步文字分析結果

下列get-document-analysis範例顯示如何取得多頁文件的非同步文字分析結果。

```
aws textract get-document-analysis \
    --job-id df7cf32ebbd2a5de113535fcf4d921926a701b09b4e7d089f3aebadb41e0712b \
    --max-results 1000
```

輸出：

```
{
    "Blocks": [
        {
            "Geometry": {
                "BoundingBox": {
                    "Width": 1.0,
                    "Top": 0.0,
                    "Left": 0.0,
                    "Height": 1.0
                },
                "Polygon": [
                    {
                        "Y": 0.0,
                        "X": 0.0
                    },
                    {
                        "Y": 0.0,
                        "X": 1.0
                    },
                    {
                        "Y": 1.0,
                        "X": 1.0
```

```
                    },
                    {
                        "Y": 1.0,
                        "X": 0.0
                    }
                ]
            },
            "Relationships": [
                {
                    "Type": "CHILD",
                    "Ids": [
                        "75966e64-81c2-4540-9649-d66ec341cd8f",
                        "bb099c24-8282-464c-a179-8a9fa0a057f0",
                        "5ebf522d-f9e4-4dc7-bfae-a288dc094595"
                    ]
                }
            ],
            "BlockType": "PAGE",
            "Id": "247c28ee-b63d-4aeb-9af0-5f7ea8ba109e",
            "Page": 1
        }
    ],
    "NextToken": "cY1W3eTFvoB0cH7YrKVudI4Gb0H8J0xAYLo8xI/JunCIPWCthaKQ+07n/
ElyutsSy0+1VOImoTRmP1zw4P0RFtaeV9Bzhnfedpx1YqwB4xaGDA==",
    "DocumentMetadata": {
        "Pages": 1
    },
    "JobStatus": "SUCCEEDED"
}
```

如需詳細資訊，請參閱《Amazon Textract 開發人員指南》中的偵測和分析多頁文件中的文字

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDocumentAnalysis。

get-document-text-detection

以下程式碼範例顯示如何使用 get-document-text-detection。

AWS CLI

在多頁文件中取得非同步文字偵測的結果

下列get-document-text-detection範例顯示如何在多頁文件中取得非同步文字偵測的結果。

```
aws textract get-document-text-detection \
    --job-id 57849a3dc627d4df74123dca269d69f7b89329c870c65bb16c9fd63409d200b9 \
    --max-results 1000
```

輸出

```
{
    "Blocks": [
        {
            "Geometry": {
                "BoundingBox": {
                    "Width": 1.0,
                    "Top": 0.0,
                    "Left": 0.0,
                    "Height": 1.0
                },
                "Polygon": [
                    {
                        "Y": 0.0,
                        "X": 0.0
                    },
                    {
                        "Y": 0.0,
                        "X": 1.0
                    },
                    {
                        "Y": 1.0,
                        "X": 1.0
                    },
                    {
                        "Y": 1.0,
                        "X": 0.0
                    }
                ]
            },
            "Relationships": [
                {
                    "Type": "CHILD",
                    "Ids": [
                        "1b926a34-0357-407b-ac8f-ec473160c6a9",
                        "0c35dc17-3605-4c9d-af1a-d9451059df51",
                        "dea3db8a-52c2-41c0-b50c-81f66f4aa758"
                    ]
```

```
                }
            ],
            "BlockType": "PAGE",
            "Id": "84671a5e-8c99-43be-a9d1-6838965da33e",
            "Page": 1
        }
    ],
    "NextToken": "GcqyoAJuZwujOT35EN4LCI3EUzMtiLq3nKyFFHvU5q1SaIdEBcSty+njNgoWwuMP/
muqc96S4o5NzDqehhXvhkodMyVO5OJGyms5lsrCxibWJw==",
    "DocumentMetadata": {
        "Pages": 1
    },
    "JobStatus": "SUCCEEDED"
}
```

如需詳細資訊，請參閱《Amazon Textract 開發人員指南》中的偵測和分析多頁文件中的文字

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDocumentTextDetection。

start-document-analysis

以下程式碼範例顯示如何使用 start-document-analysis。

AWS CLI

開始分析多頁文件中的文字

下列start-document-analysis範例示範如何開始多頁文件中文字的非同步分析。

Linux/macOS：

```
aws textract start-document-analysis \
    --document-location '{"S3Object":{"Bucket":"bucket","Name":"document"}}' \
    --feature-types '["TABLES","FORMS"]' \
    --notification-channel "SNSTopicArn=arn:snsTopic,RoleArn=roleArn"
```

Windows：

```
aws textract start-document-analysis \
    --document-location "{\"S3Object\":{\"Bucket\":\"bucket\",\"Name\":\"document
\"}}" \
    --feature-types "[\"TABLES\", \"FORMS\"]" \
```

```
    --region region-name \
    --notification-channel "SNSTopicArn=arn:snsTopic,RoleArn=roleArn"
```

輸出：

```
{
    "JobId": "df7cf32ebbd2a5de113535fcf4d921926a701b09b4e7d089f3aebadb41e0712b"
}
```

如需詳細資訊，請參閱《Amazon Textract 開發人員指南》中的偵測和分析多頁文件中的文字

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartDocumentAnalysis。

`start-document-text-detection`

以下程式碼範例顯示如何使用 `start-document-text-detection`。

AWS CLI

開始偵測多頁文件中的文字

下列`start-document-text-detection`範例示範如何啟動多頁文件中文字的非同步偵測。

Linux/macOS：

```
aws textract start-document-text-detection \
        --document-location '{"S3Object":{"Bucket":"bucket","Name":"document"}}' \
        --notification-channel "SNSTopicArn=arn:snsTopic,RoleArn=roleARN"
```

Windows：

```
aws textract start-document-text-detection \
    --document-location "{\"S3Object\":{\"Bucket\":\"bucket\",\"Name\":\"document
\"}}" \
    --region region-name \
    --notification-channel "SNSTopicArn=arn:snsTopic,RoleArn=roleArn"
```

輸出：

```
{
```

```
    "JobId": "57849a3dc627d4df74123dca269d69f7b89329c870c65bb16c9fd63409d200b9"
}
```

如需詳細資訊，請參閱《Amazon Textract 開發人員指南》中的偵測和分析多頁文件中的文字

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartDocumentTextDetection。

使用 的 Amazon Transcribe 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon Transcribe 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-language-model`

以下程式碼範例顯示如何使用 `create-language-model`。

AWS CLI

範例 1：使用訓練和調校資料建立自訂語言模型。

下列`create-language-model`範例會建立自訂語言模型。您可以使用自訂語言模型來改善法律、接待、財務和保險等領域的轉錄效能。針對語言代碼，輸入有效的語言代碼。針對 base-model-name，指定最適合您想要使用自訂語言模型轉錄之音訊範例速率的基本模型。對於 model-name，指定您要呼叫自訂語言模型的名稱。

```
aws transcribe create-language-model \
    --language-code language-code \
    --base-model-name base-model-name \
    --model-name cli-clm-example \
```

```
    --input-data-config S3Uri="s3://amzn-s3-demo-bucket/Amazon-S3-Prefix-for-
training-data",TuningDataS3Uri="s3://amzn-s3-demo-bucket/Amazon-S3-Prefix-for-
tuning-data",DataAccessRoleArn="arn:aws:iam::AWS-account-number:role/IAM-role-with-
permissions-to-create-a-custom-language-model"
```

輸出：

```
{
    "LanguageCode": "language-code",
    "BaseModelName": "base-model-name",
    "ModelName": "cli-clm-example",
    "InputDataConfig": {
        "S3Uri": "s3://amzn-s3-demo-bucket/Amazon-S3-Prefix/",
        "TuningDataS3Uri": "s3://amzn-s3-demo-bucket/Amazon-S3-Prefix/",
        "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-with-
permissions-create-a-custom-language-model"
    },
    "ModelStatus": "IN_PROGRESS"
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的使用自訂語言模型改善特定領域的轉錄準確性。

範例 2：僅使用訓練資料建立自訂語言模型。

以下 `create-language-model` 範例會轉錄您的音訊檔案。您可以使用自訂語言模型來改善法律、接待、財務和保險等領域的轉錄效能。針對語言代碼，輸入有效的語言代碼。針對 base-model-name，指定最適合您想要使用自訂語言模型轉錄之音訊範例速率的基本模型。對於 model-name，指定您要呼叫自訂語言模型的名稱。

```
aws transcribe create-language-model \
    --language-code en-US \
    --base-model-name base-model-name \
    --model-name cli-clm-example \
    --input-data-config S3Uri="s3://amzn-s3-demo-bucket/Amazon-S3-Prefix-For-
Training-Data",DataAccessRoleArn="arn:aws:iam::AWS-account-number:role/IAM-role-
with-permissions-to-create-a-custom-language-model"
```

輸出：

```
{
```

```
    "LanguageCode": "en-US",
    "BaseModelName": "base-model-name",
    "ModelName": "cli-clm-example",
    "InputDataConfig": {
        "S3Uri": "s3://amzn-s3-demo-bucket/Amazon-S3-Prefix-For-Training-Data/",
        "DataAccessRoleArn": "arn:aws:iam::your-AWS-account-number:role/IAM-role-
with-permissions-to-create-a-custom-language-model"
    },
    "ModelStatus": "IN_PROGRESS"
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的使用自訂語言模型改善特定領域的轉錄準確性。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLanguageModel。

`create-medical-vocabulary`

以下程式碼範例顯示如何使用 `create-medical-vocabulary`。

AWS CLI

建立醫療自訂詞彙

以下 `create-medical-vocabulary` 範例會建立自訂詞彙。若要建立自訂詞彙，您必須先建立一個文字檔案，其中包含要更準確轉錄的所有字詞。針對 vocabulary-file-uri，指定該文字檔案的 Amazon Simple Storage Service (Amazon S3) URI。針對 language-code，指定對應您的自訂詞彙語言的語言代碼。針對 vocabulary-name，指定您自訂詞彙的稱呼。

```
aws transcribe create-medical-vocabulary \
    --vocabulary-name cli-medical-vocab-example \
    --language-code language-code \
    --vocabulary-file-uri https://amzn-s3-demo-bucket.AWS-Region.amazonaws.com/the-
text-file-for-the-medical-custom-vocabulary.txt
```

輸出：

```
{
    "VocabularyName": "cli-medical-vocab-example",
    "LanguageCode": "language-code",
    "VocabularyState": "PENDING"
```

```
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的醫學自訂詞彙。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateMedicalVocabulary。

`create-vocabulary-filter`

以下程式碼範例顯示如何使用 `create-vocabulary-filter`。

AWS CLI

建立詞彙篩選條件

下列`create-vocabulary-filter`範例會建立詞彙篩選條件，該篩選條件使用文字檔案，其中包含您不想出現在轉錄中的字詞清單。針對語言代碼，指定與詞彙篩選條件語言對應的語言代碼。針對 vocabulary-filter-file-uri，指定文字檔案的 Amazon Simple Storage Service (Amazon S3) URI。針對 vocabulary-filter-name，指定詞彙篩選條件的名稱。

```
aws transcribe create-vocabulary-filter \
    --language-code language-code \
    --vocabulary-filter-file-uri s3://amzn-s3-demo-bucket/vocabulary-filter.txt \
    --vocabulary-filter-name cli-vocabulary-filter-example
```

輸出：

```
{
    "VocabularyFilterName": "cli-vocabulary-filter-example",
    "LanguageCode": "language-code"
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的篩選不需要的字詞。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateVocabularyFilter。

`create-vocabulary`

以下程式碼範例顯示如何使用 `create-vocabulary`。

AWS CLI

建立自訂詞彙

以下 create-vocabulary 範例會建立自訂詞彙。若要建立自訂詞彙，您必須先建立一個文字檔案，其中包含要更準確轉錄的所有字詞。針對 vocabulary-file-uri，指定該文字檔案的 Amazon Simple Storage Service (Amazon S3) URI。針對 language-code，指定對應您的自訂詞彙語言的語言代碼。針對 vocabulary-name，指定您自訂詞彙的稱呼。

```
aws transcribe create-vocabulary \
    --language-code language-code \
    --vocabulary-name cli-vocab-example \
    --vocabulary-file-uri s3://amzn-s3-demo-bucket/Amazon-S3-prefix/the-text-file-for-the-custom-vocabulary.txt
```

輸出：

```
{
    "VocabularyName": "cli-vocab-example",
    "LanguageCode": "language-code",
    "VocabularyState": "PENDING"
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的自訂詞彙。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateVocabulary。

delete-language-model

以下程式碼範例顯示如何使用 delete-language-model。

AWS CLI

刪除自訂語言模型

下列delete-language-model範例會刪除自訂語言模型。

```
aws transcribe delete-language-model \
    --model-name model-name
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的使用自訂語言模型改善特定領域的轉錄準確性。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLanguageModel。

delete-medical-transcription-job

以下程式碼範例顯示如何使用 delete-medical-transcription-job。

AWS CLI

刪除醫學轉錄作業

以下 delete-medical-transcription-job 範例會刪除醫學轉錄作業。

```
aws transcribe delete-medical-transcription-job \
    --medical-transcription-job-name medical-transcription-job-name
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的 DeleteMedicalTranscriptionJob。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteMedicalTranscriptionJob。

delete-medical-vocabulary

以下程式碼範例顯示如何使用 delete-medical-vocabulary。

AWS CLI

刪除醫療自訂詞彙

下列delete-medical-vocabulary範例會刪除醫療自訂詞彙。對於詞彙名稱，請指定醫療自訂詞彙的名稱。

```
aws transcribe delete-vocabulary \
    --vocabulary-name medical-custom-vocabulary-name
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的醫學自訂詞彙。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteMedicalVocabulary。

delete-transcription-job

以下程式碼範例顯示如何使用 delete-transcription-job。

AWS CLI

刪除其中一項轉錄作業

以下 delete-transcription-job 範例會刪除其中一項轉錄作業。

```
aws transcribe delete-transcription-job \
    --transcription-job-name your-transcription-job
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的 DeleteTranscriptionJob。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteTranscriptionJob。

delete-vocabulary-filter

以下程式碼範例顯示如何使用 delete-vocabulary-filter。

AWS CLI

刪除詞彙篩選條件

下列delete-vocabulary-filter範例會刪除詞彙篩選條件。

```
aws transcribe delete-vocabulary-filter \
    --vocabulary-filter-name vocabulary-filter-name
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的篩選不需要的字詞。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteVocabularyFilter。

delete-vocabulary

以下程式碼範例顯示如何使用 delete-vocabulary。

AWS CLI

刪除自訂詞彙

以下 delete-vocabulary 範例會刪除自訂詞彙。

```
aws transcribe delete-vocabulary \
    --vocabulary-name vocabulary-name
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的自訂詞彙。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteVocabulary。

describe-language-model

以下程式碼範例顯示如何使用 describe-language-model。

AWS CLI

取得特定自訂語言模型的相關資訊

下列describe-language-model範例取得特定自訂語言模型的相關資訊。例如，在 BaseModelName 下，您可以查看模型是使用 NarrowBand 還是 WideBand 模型進行訓練。具有 NarrowBand 基礎模型的自訂語言模型可以轉錄取樣率小於 16 kHz 的音訊。使用 WideBand 基礎模型的語言模型可以轉錄取樣率大於 16 kHz 的音訊。S3Uri 參數指出您用來存取訓練資料的 Amazon S3 字首，以建立自訂語言模型。

```
aws transcribe describe-language-model \
    --model-name cli-clm-example
```

輸出：

```
{
    "LanguageModel": {
        "ModelName": "cli-clm-example",
        "CreateTime": "2020-09-25T17:57:38.504000+00:00",
        "LastModifiedTime": "2020-09-25T17:57:48.585000+00:00",
        "LanguageCode": "language-code",
        "BaseModelName": "base-model-name",
        "ModelStatus": "IN_PROGRESS",
```

```
        "UpgradeAvailability": false,
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/Amazon-S3-Prefix/",
            "TuningDataS3Uri": "s3://amzn-s3-demo-bucket/Amazon-S3-Prefix/",
            "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-
with-permissions-to-create-a-custom-language-model"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的使用自訂語言模型改善特定領域的轉錄準確性。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeLanguageModel。

get-medical-transcription-job

以下程式碼範例顯示如何使用 get-medical-transcription-job。

AWS CLI

取得特定醫療轉錄工作的相關資訊

下列get-medical-transcription-job範例取得特定醫療轉錄工作的相關資訊。若要存取轉錄結果，請使用 TranscriptFileUri 參數。如果您已為轉錄任務啟用其他功能，您可以在設定物件中看到這些功能。專科參數顯示提供者的醫療專科。類型參數指出轉錄任務中的語音是醫療對話還是醫療聽寫。

```
aws transcribe get-medical-transcription-job \
    --medical-transcription-job-name vocabulary-dictation-medical-transcription-job
```

輸出：

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "vocabulary-dictation-medical-transcription-
job",
        "TranscriptionJobStatus": "COMPLETED",
        "LanguageCode": "en-US",
        "MediaSampleRateHertz": 48000,
        "MediaFormat": "mp4",
        "Media": {
```

```
            "MediaFileUri": "s3://Amazon-S3-Prefix/your-audio-file.file-extension"
        },
        "Transcript": {
            "TranscriptFileUri": "https://s3.Region.amazonaws.com/Amazon-S3-Prefix/
vocabulary-dictation-medical-transcription-job.json"
        },
        "StartTime": "2020-09-21T21:17:27.045000+00:00",
        "CreationTime": "2020-09-21T21:17:27.016000+00:00",
        "CompletionTime": "2020-09-21T21:17:59.561000+00:00",
        "Settings": {
            "ChannelIdentification": false,
            "ShowAlternatives": false,
            "VocabularyName": "cli-medical-vocab-example"
        },
        "Specialty": "PRIMARYCARE",
        "Type": "DICTATION"
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的批次轉錄。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMedicalTranscriptionJob。

get-medical-vocabulary

以下程式碼範例顯示如何使用 get-medical-vocabulary。

AWS CLI

取得醫療自訂詞彙的相關資訊

下列get-medical-vocabulary範例取得醫療自訂詞彙的相關資訊。您可以使用 VocabularyState 參數來查看詞彙的處理狀態。如果是 READY，您可以在 StartMedicalTranscriptionJob 操作中使用它：

```
aws transcribe get-medical-vocabulary \
    --vocabulary-name medical-vocab-example
```

輸出：

```
{
    "VocabularyName": "medical-vocab-example",
```

```
    "LanguageCode": "en-US",
    "VocabularyState": "READY",
    "LastModifiedTime": "2020-09-19T23:59:04.349000+00:00",
    "DownloadUri": "https://link-to-download-the-text-file-used-to-create-your-
medical-custom-vocabulary"
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的醫學自訂詞彙。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMedicalVocabulary。

get-transcription-job

以下程式碼範例顯示如何使用 get-transcription-job。

AWS CLI

取得特定轉錄作業的相關資訊

以下 get-transcription-job 範例會取得特定轉錄作業的相關資訊。若要存取轉錄結果，請使用 TranscriptFileUri 參數。使用 MediaFileUri 參數來查看您在此作業中轉錄的音訊檔案。您可以使用 Settings 物件來查看您在轉錄作業中已啟用的選用功能。

```
aws transcribe get-transcription-job \
    --transcription-job-name your-transcription-job
```

輸出：

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "your-transcription-job",
        "TranscriptionJobStatus": "COMPLETED",
        "LanguageCode": "language-code",
        "MediaSampleRateHertz": 48000,
        "MediaFormat": "mp4",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.file-
extension"
        },
        "Transcript": {
            "TranscriptFileUri": "https://Amazon-S3-file-location-of-transcription-
output"
```

```
        },
        "StartTime": "2020-09-18T22:27:23.970000+00:00",
        "CreationTime": "2020-09-18T22:27:23.948000+00:00",
        "CompletionTime": "2020-09-18T22:28:21.197000+00:00",
        "Settings": {
            "ChannelIdentification": false,
            "ShowAlternatives": false
        },
        "IdentifyLanguage": true,
        "IdentifiedLanguageScore": 0.8672199249267578
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的入門 (AWS 命令列界面) 。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetTranscriptionJob。

get-vocabulary-filter

以下程式碼範例顯示如何使用 get-vocabulary-filter。

AWS CLI

取得詞彙篩選條件的相關資訊

下列get-vocabulary-filter範例取得詞彙篩選條件的相關資訊。您可以使用 DownloadUri 參數來取得您用來建立詞彙篩選條件的字詞清單。

```
aws transcribe get-vocabulary-filter \
    --vocabulary-filter-name testFilter
```

輸出：

```
{
    "VocabularyFilterName": "testFilter",
    "LanguageCode": "language-code",
    "LastModifiedTime": "2020-05-07T22:39:32.147000+00:00",
    "DownloadUri": "https://Amazon-S3-location-to-download-your-vocabulary-filter"
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的篩選不需要的字詞。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetVocabularyFilter。

get-vocabulary

以下程式碼範例顯示如何使用 get-vocabulary。

AWS CLI

取得自訂詞彙的相關資訊

以下 get-vocabulary 範例會取得先前建立的自訂詞彙的相關資訊。

```
aws transcribe get-vocabulary \
    --vocabulary-name cli-vocab-1
```

輸出：

```
{
    "VocabularyName": "cli-vocab-1",
    "LanguageCode": "language-code",
    "VocabularyState": "READY",
    "LastModifiedTime": "2020-09-19T23:22:32.836000+00:00",
    "DownloadUri": "https://link-to-download-the-text-file-used-to-create-your-
custom-vocabulary"
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的自訂詞彙。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetVocabulary。

list-language-models

以下程式碼範例顯示如何使用 list-language-models。

AWS CLI

列出您的自訂語言模型

下列list-language-models範例列出與 AWS 您的帳戶和區域相關聯的自訂語言模型。您可以使用 S3Uri和 TuningDataS3Uri 參數來尋找做為訓練資料或調校資料的 Amazon S3 字首。BaseModelName 會告訴您是使用 NarrowBand 還是 WideBand 模型來建立自訂語言模型。您可以使用 NarrowBand 基礎模型，使用自訂語言模型轉錄取樣率小於 16 kHz 的音訊。您可以使用 WideBand 基礎模型，使用自訂語言模型轉錄 16 kHz 或更高的音訊。ModelStatus 參數會顯示您是否可以在轉錄任務中使用自訂語言模型。如果值為 COMPLETED，您可以在轉錄任務中使用它。

```
aws transcribe list-language-models
```

輸出：

```
{
    "Models": [
        {
            "ModelName": "cli-clm-2",
            "CreateTime": "2020-09-25T17:57:38.504000+00:00",
            "LastModifiedTime": "2020-09-25T17:57:48.585000+00:00",
            "LanguageCode": "language-code",
            "BaseModelName": "WideBand",
            "ModelStatus": "IN_PROGRESS",
            "UpgradeAvailability": false,
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/clm-training-data/",
                "TuningDataS3Uri": "s3://amzn-s3-demo-bucket/clm-tuning-data/",
                "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-
used-to-create-the-custom-language-model"
            }
        },
        {
            "ModelName": "cli-clm-1",
            "CreateTime": "2020-09-25T17:16:01.835000+00:00",
            "LastModifiedTime": "2020-09-25T17:16:15.555000+00:00",
            "LanguageCode": "language-code",
            "BaseModelName": "WideBand",
            "ModelStatus": "IN_PROGRESS",
            "UpgradeAvailability": false,
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/clm-training-data/",
                "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-
used-to-create-the-custom-language-model"
            }
        },
        {
            "ModelName": "clm-console-1",
            "CreateTime": "2020-09-24T19:26:28.076000+00:00",
            "LastModifiedTime": "2020-09-25T04:25:22.271000+00:00",
            "LanguageCode": "language-code",
            "BaseModelName": "NarrowBand",
            "ModelStatus": "COMPLETED",
            "UpgradeAvailability": false,
```

```
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/clm-training-data/",
                "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-
used-to-create-the-custom-language-model"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的使用自訂語言模型改善特定領域的轉錄準確性。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListLanguageModels。

list-medical-transcription-jobs

以下程式碼範例顯示如何使用 list-medical-transcription-jobs。

AWS CLI

列出醫學轉錄作業

下列list-medical-transcription-jobs範例列出與 AWS 您的帳戶和區域相關聯的醫療轉錄任務。若要取得有關特定轉錄作業的詳細資訊，請複製轉錄輸出中 MedicalTranscriptionJobName 參數的值，然後指定 get-medical-transcription-job 命令的 MedicalTranscriptionJobName 選項值。若要進一步查看轉錄作業，請複製 NextToken 參數的值，再次執行 list-medical-transcription-jobs 命令，然後在 --next-token 選項中指定該值。

```
aws transcribe list-medical-transcription-jobs
```

輸出：

```
{
    "NextToken": "3/PblzkiGhzjER3KHuQt2fmbPLF7cDYafjFMEoGn44ON/
gsuUSTIkGyanvRE6WMXFd/ZTEc2EZj+P9eii/
z1O2FDYli6RLI0WoRX4RwMisVrh9G0Kie0Y8ikBCdtqlZB10Wa9McC+ebOl
+LaDtZPC4u6ttoHLRlEfzqstHXSgapXg3tEBtm9piIaPB6MOM5BB6t86+qtmocTR/
qrteHZBBudhTfbCwhsxaqujHiiUvFdm3BQbKKWIW06yV9b+4f38oD2lVIan
+vfUs3gBYAl5VTDmXXzQPBQOHPjtwmFI+IWX15nSUjWuN3TUylHgPWzDaYT8qBtu0Z+3UG4V6b
+K2CC0XszXg5rBq9hYgNzy4XoFh/6s5DoSnzq49Q9xHgHdT2yBADFmvFK7myZBsj75+2vQZOSVpWUPy3WT/32zFAcoEL
+mFYfUjtTZ8n/jq7aQEjQ42A
```

```
+X/7K6JgOcdVPtEg8PlDr5kgYYG3q3OmYXX37U3FZuJmnTI63VtIXsNnOU5eGoYObtpk00Nq9UkzgSJxqj84ZD5n
+S0EGy9ZUYBJRRcGeYUM3Q4DbSJfUwSAqcFdLIWZdp8qIREMQIBWy7BLwSdyqsQo2vRrd53hm5aWM7SVf6pPq6X/
IXR5+1eUOOD8/coaTT4ES2DerbV6RkV4o0VT1d0SdVX/
MmtkNG8nYj8PqU07w7988quh1ZP6D80veJS1q73tUUR9MjnGernW2tAnvnLNhdefBcD
+sZVfYq3iBMFY7wTy1P1G6NqW9GrYDYoX3tTPWlD7phpbVSyKrh/
PdYrps5UxnsGoA1b7L/FfAXDfUoGrGUB4N3JsPYXX9D++g+6gV1qBBs/
WfF934aKqfD6UTggm/zV3GAOWiBpfvAZRvEb924i6yGHyMC7y54O1ZAwSBupmI
+FFd13CaPO4kN1vJlth6aM5vUPXg4BpyUhtbRhwD/KxCvf9K0tLJGyL1A==",
    "MedicalTranscriptionJobSummaries": [
        {
            "MedicalTranscriptionJobName": "vocabulary-dictation-medical-
transcription-job",
            "CreationTime": "2020-09-21T21:17:27.016000+00:00",
            "StartTime": "2020-09-21T21:17:27.045000+00:00",
            "CompletionTime": "2020-09-21T21:17:59.561000+00:00",
            "LanguageCode": "en-US",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "CUSTOMER_BUCKET",
            "Specialty": "PRIMARYCARE",
            "Type": "DICTATION"
        },
        {
            "MedicalTranscriptionJobName": "alternatives-dictation-medical-
transcription-job",
            "CreationTime": "2020-09-21T21:01:14.569000+00:00",
            "StartTime": "2020-09-21T21:01:14.592000+00:00",
            "CompletionTime": "2020-09-21T21:01:43.606000+00:00",
            "LanguageCode": "en-US",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "CUSTOMER_BUCKET",
            "Specialty": "PRIMARYCARE",
            "Type": "DICTATION"
        },
        {
            "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
            "CreationTime": "2020-09-21T19:09:18.171000+00:00",
            "StartTime": "2020-09-21T19:09:18.199000+00:00",
            "CompletionTime": "2020-09-21T19:10:22.516000+00:00",
            "LanguageCode": "en-US",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "CUSTOMER_BUCKET",
            "Specialty": "PRIMARYCARE",
            "Type": "CONVERSATION"
```

```
        },
        {
            "MedicalTranscriptionJobName": "speaker-id-conversation-medical-
transcription-job",
            "CreationTime": "2020-09-21T18:43:37.157000+00:00",
            "StartTime": "2020-09-21T18:43:37.265000+00:00",
            "CompletionTime": "2020-09-21T18:44:21.192000+00:00",
            "LanguageCode": "en-US",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "CUSTOMER_BUCKET",
            "Specialty": "PRIMARYCARE",
            "Type": "CONVERSATION"
        },
        {
            "MedicalTranscriptionJobName": "multichannel-conversation-medical-
transcription-job",
            "CreationTime": "2020-09-20T23:46:44.053000+00:00",
            "StartTime": "2020-09-20T23:46:44.081000+00:00",
            "CompletionTime": "2020-09-20T23:47:35.851000+00:00",
            "LanguageCode": "en-US",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "CUSTOMER_BUCKET",
            "Specialty": "PRIMARYCARE",
            "Type": "CONVERSATION"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的 https://docs.aws.amazon.com/transcribe/latest/dg/batch-med-transcription.html>。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListMedicalTranscriptionJobs。

`list-medical-vocabularies`

以下程式碼範例顯示如何使用 `list-medical-vocabularies`。

AWS CLI

列出醫療自訂詞彙

下列`list-medical-vocabularies`範例列出與 AWS 您的帳戶和區域相關聯的醫療自訂詞彙。若要取得特定轉錄任務的詳細資訊，請在轉錄輸出中複製`MedicalTranscriptionJobName`參數

的值，並為 get-medical-transcription-job命令的 MedicalTranscriptionJobName選項指定該值。若要查看更多轉錄任務，請複製 NextToken 參數的值，再次執行 list-medical-transcription-jobs命令，然後在 --next-token選項中指定該值。

```
aws transcribe list-medical-vocabularies
```

輸出：

```
{
    "Vocabularies": [
        {
            "VocabularyName": "cli-medical-vocab-2",
            "LanguageCode": "en-US",
            "LastModifiedTime": "2020-09-21T21:44:59.521000+00:00",
            "VocabularyState": "READY"
        },
        {
            "VocabularyName": "cli-medical-vocab-1",
            "LanguageCode": "en-US",
            "LastModifiedTime": "2020-09-19T23:59:04.349000+00:00",
            "VocabularyState": "READY"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的醫學自訂詞彙。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMedicalVocabularies。

list-transcription-jobs

以下程式碼範例顯示如何使用 list-transcription-jobs。

AWS CLI

列出您的轉錄作業

下列list-transcription-jobs範例列出與 AWS 您的帳戶和區域相關聯的轉錄任務。

```
aws transcribe list-transcription-jobs
```

輸出：

```
{
    "NextToken": "NextToken",
    "TranscriptionJobSummaries": [
        {
            "TranscriptionJobName": "speak-id-job-1",
            "CreationTime": "2020-08-17T21:06:15.391000+00:00",
            "StartTime": "2020-08-17T21:06:15.416000+00:00",
            "CompletionTime": "2020-08-17T21:07:05.098000+00:00",
            "LanguageCode": "language-code",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "SERVICE_BUCKET"
        },
        {
            "TranscriptionJobName": "job-1",
            "CreationTime": "2020-08-17T20:50:24.207000+00:00",
            "StartTime": "2020-08-17T20:50:24.230000+00:00",
            "CompletionTime": "2020-08-17T20:52:18.737000+00:00",
            "LanguageCode": "language-code",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "SERVICE_BUCKET"
        },
        {
            "TranscriptionJobName": "sdk-test-job-4",
            "CreationTime": "2020-08-17T20:32:27.917000+00:00",
            "StartTime": "2020-08-17T20:32:27.956000+00:00",
            "CompletionTime": "2020-08-17T20:33:15.126000+00:00",
            "LanguageCode": "language-code",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "SERVICE_BUCKET"
        },
        {
            "TranscriptionJobName": "Diarization-speak-id",
            "CreationTime": "2020-08-10T22:10:09.066000+00:00",
            "StartTime": "2020-08-10T22:10:09.116000+00:00",
            "CompletionTime": "2020-08-10T22:26:48.172000+00:00",
            "LanguageCode": "language-code",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "SERVICE_BUCKET"
        },
        {
            "TranscriptionJobName": "your-transcription-job-name",
            "CreationTime": "2020-07-29T17:45:09.791000+00:00",
```

```
            "StartTime": "2020-07-29T17:45:09.826000+00:00",
            "CompletionTime": "2020-07-29T17:46:20.831000+00:00",
            "LanguageCode": "language-code",
            "TranscriptionJobStatus": "COMPLETED",
            "OutputLocationType": "SERVICE_BUCKET"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的入門 (AWS 命令列界面) 。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTranscriptionJobs。

`list-vocabularies`

以下程式碼範例顯示如何使用 `list-vocabularies`。

AWS CLI

列出您的自訂詞彙

下列`list-vocabularies`範例列出與 AWS 您的帳戶和區域相關聯的自訂詞彙。

```
aws transcribe list-vocabularies
```

輸出：

```
{
    "NextToken": "NextToken",
    "Vocabularies": [
        {
            "VocabularyName": "ards-test-1",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-04-27T22:00:27.330000+00:00",
            "VocabularyState": "READY"
        },
        {
            "VocabularyName": "sample-test",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-04-24T23:04:11.044000+00:00",
            "VocabularyState": "READY"
        },
```

```
        {
            "VocabularyName": "CRLF-to-LF-test-3-1",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-04-24T22:12:22.277000+00:00",
            "VocabularyState": "READY"
        },
        {
            "VocabularyName": "CRLF-to-LF-test-2",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-04-24T21:53:50.455000+00:00",
            "VocabularyState": "READY"
        },
        {
            "VocabularyName": "CRLF-to-LF-1-1",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-04-24T21:39:33.356000+00:00",
            "VocabularyState": "READY"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的自訂詞彙。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListVocabularies。

list-vocabulary-filters

以下程式碼範例顯示如何使用 `list-vocabulary-filters`。

AWS CLI

列出詞彙篩選條件

下列`list-vocabulary-filters`範例列出與 AWS 您的帳戶和區域相關聯的詞彙篩選條件。

```
aws transcribe list-vocabulary-filters
```

輸出：

```
{
    "NextToken": "NextToken": [
        {
            "VocabularyFilterName": "testFilter",
```

```
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-05-07T22:39:32.147000+00:00"
        },
        {
            "VocabularyFilterName": "testFilter2",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-05-21T23:29:35.174000+00:00"
        },
        {
            "VocabularyFilterName": "filter2",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-05-08T20:18:26.426000+00:00"
        },
        {
            "VocabularyFilterName": "filter-review",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-06-03T18:52:30.448000+00:00"
        },
        {
            "VocabularyFilterName": "crlf-filt",
            "LanguageCode": "language-code",
            "LastModifiedTime": "2020-05-22T19:42:42.737000+00:00"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的篩選不需要的字詞。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListVocabularyFilters。

start-medical-transcription-job

以下程式碼範例顯示如何使用 start-medical-transcription-job。

AWS CLI

範例 1：轉錄儲存為音訊檔案的醫學聽寫

以下 start-medical-transcription-job 範例會轉錄音訊檔案。您可在 OutputBucketName 參數中指定轉錄輸出的位置。

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://myfile.json
```

myfile.json 的內容：

```
{
    "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
    }
}
```

輸出：

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-20T00:35:22.256000+00:00",
        "CreationTime": "2020-09-20T00:35:22.218000+00:00",
        "Specialty": "PRIMARYCARE",
        "Type": "DICTATION"
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的批次轉錄概觀。

範例 2：轉錄儲存為音訊檔案的臨床醫師與病患對話

以下 start-medical-transcription-job 範例會轉錄包含臨床醫師與病患對話的音訊檔案。您可在 OutputBucketName 參數中指定轉錄輸出的位置。

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://mysecondfile.json
```

mysecondfile.json 的內容：

```
{
    "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
    }
}
```

輸出：

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "simple-conversation-medical-transcription-
job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-20T23:19:49.965000+00:00",
        "CreationTime": "2020-09-20T23:19:49.941000+00:00",
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的批次轉錄概觀。

範例 3：轉錄臨床醫師與病患對話的多聲道音訊檔案

以下 `start-medical-transcription-job` 範例會轉錄音訊檔案中每個聲道的音訊，並將每個聲道的個別轉錄合併成單一轉錄輸出。您可在 `OutputBucketName` 參數中指定轉錄輸出的位置。

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://mythirdfile.json
```

`mythirdfile.json` 的內容：

```
{
```

```
    "MedicalTranscriptionJobName": "multichannel-conversation-medical-transcription-
job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
        "Media": {
          "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "Settings":{
          "ChannelIdentification": true
        }
}
```

輸出：

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "multichannel-conversation-medical-
transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-20T23:46:44.081000+00:00",
        "CreationTime": "2020-09-20T23:46:44.053000+00:00",
        "Settings": {
            "ChannelIdentification": true
        },
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的聲道識別。

範例 4：轉錄臨床醫生與病患對話的音訊檔案，並識別轉錄輸出中的發言者

以下 `start-medical-transcription-job` 範例會轉錄音訊檔案，並標記轉錄輸出中每一位發言者的語音。您可在 `OutputBucketName` 參數中指定轉錄輸出的位置。

```
aws transcribe start-medical-transcription-job \
```

```
    --cli-input-json file://myfourthfile.json
```

myfourthfile.json 的內容：

```
{
    "MedicalTranscriptionJobName": "speaker-id-conversation-medical-transcription-
job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
    "Settings":{
        "ShowSpeakerLabels": true,
        "MaxSpeakerLabels": 2
        }
}
```

輸出：

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "speaker-id-conversation-medical-
transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-21T18:43:37.265000+00:00",
        "CreationTime": "2020-09-21T18:43:37.157000+00:00",
        "Settings": {
            "ShowSpeakerLabels": true,
            "MaxSpeakerLabels": 2
        },
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的識別發言者。

範例 5：轉錄儲存為音訊檔案且最多有兩個轉錄替代選項的醫學對話

以下 `start-medical-transcription-job` 範例會從單一音訊檔案建立最多兩個替代轉錄。每個轉錄都有相關聯的可信度等級。根據預設，Amazon Transcribe 會傳回最高可信度等級的轉錄。您可以指定 Amazon Transcribe 傳回可信度較低的其他轉錄。您可在 `OutputBucketName` 參數中指定轉錄輸出的位置。

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://myfifthfile.json
```

`myfifthfile.json` 的內容：

```
{
    "MedicalTranscriptionJobName": "alternatives-conversation-medical-transcription-
job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
    },
    "Settings":{
        "ShowAlternatives": true,
        "MaxAlternatives": 2
    }
}
```

輸出：

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-21T19:09:18.199000+00:00",
        "CreationTime": "2020-09-21T19:09:18.171000+00:00",
        "Settings": {
```

```
            "ShowAlternatives": true,
            "MaxAlternatives": 2
        },
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的替代轉錄。

範例 6：轉錄最多有兩個替代轉錄的醫學聽寫的音訊檔案

以下 start-medical-transcription-job 範例會轉錄音訊檔案，並使用詞彙篩選器來為任何不希望出現的字詞加上遮罩。您可在 OutputBucketName 參數中指定轉錄輸出的位置。

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://mysixthfile.json
```

mysixthfile.json 的內容：

```
{
    "MedicalTranscriptionJobName": "alternatives-conversation-medical-transcription-
job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
    },
    "Settings":{
          "ShowAlternatives": true,
          "MaxAlternatives": 2
    }
}
```

輸出：

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "alternatives-dictation-medical-
transcription-job",
```

```
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-21T21:01:14.592000+00:00",
        "CreationTime": "2020-09-21T21:01:14.569000+00:00",
        "Settings": {
            "ShowAlternatives": true,
            "MaxAlternatives": 2
        },
        "Specialty": "PRIMARYCARE",
        "Type": "DICTATION"
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的替代轉錄。

範例 7：使用自訂詞彙以更準確的方式轉錄醫學聽寫的音訊檔案

以下 start-medical-transcription-job 範例會轉錄音訊檔案，並使用您先前建立的醫學自訂詞彙來提高轉錄準確度。您可在 OutputBucketName 參數中指定轉錄輸出的位置。

```
aws transcribe start-transcription-job \
    --cli-input-json file://myseventhfile.json
```

mysixthfile.json 的內容：

```
{
    "MedicalTranscriptionJobName": "vocabulary-dictation-medical-transcription-job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
    },
    "Settings":{
        "VocabularyName": "cli-medical-vocab-1"
    }
}
```

輸出：

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "vocabulary-dictation-medical-transcription-
job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-21T21:17:27.045000+00:00",
        "CreationTime": "2020-09-21T21:17:27.016000+00:00",
        "Settings": {
            "VocabularyName": "cli-medical-vocab-1"
        },
        "Specialty": "PRIMARYCARE",
        "Type": "DICTATION"
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的醫學自訂詞彙。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 StartMedicalTranscriptionJob。

start-transcription-job

以下程式碼範例顯示如何使用 start-transcription-job。

AWS CLI

範例 1：轉錄音訊檔案

以下 start-transcription-job 範例會轉錄您的音訊檔案。

```
aws transcribe start-transcription-job \
    --cli-input-json file://myfile.json
```

myfile.json 的內容：

```
{
    "TranscriptionJobName": "cli-simple-transcription-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
```

```
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-file-
name.file-extension"
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的入門 (AWS 命令列界面)。

範例 2：轉錄多聲道音訊檔案

以下 `start-transcription-job` 範例會轉錄您的多聲道音訊檔案。

```
aws transcribe start-transcription-job \
    --cli-input-json file://mysecondfile.json
```

mysecondfile.json 的內容：

```
{
    "TranscriptionJobName": "cli-channelid-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-file-
name.file-extension"
    },
    "Settings":{
        "ChannelIdentification":true
    }
}
```

輸出：

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-channelid-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "the-language-of-your-transcription-job",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
        },
        "StartTime": "2020-09-17T16:07:56.817000+00:00",
        "CreationTime": "2020-09-17T16:07:56.784000+00:00",
        "Settings": {
            "ChannelIdentification": true
```

```
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的轉錄多聲道音訊。

範例 3：轉錄音訊檔案並識別不同的發言者

以下 start-transcription-job 範例會轉錄音訊檔案，並識別轉錄輸出中的發言者。

```
aws transcribe start-transcription-job \
    --cli-input-json file://mythirdfile.json
```

mythirdfile.json 的內容：

```
{
    "TranscriptionJobName": "cli-speakerid-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-file-
name.file-extension"
    },
    "Settings":{
    "ShowSpeakerLabels": true,
    "MaxSpeakerLabels": 2
    }
}
```

輸出：

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-speakerid-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "the-language-of-your-transcription-job",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
        },
        "StartTime": "2020-09-17T16:22:59.696000+00:00",
        "CreationTime": "2020-09-17T16:22:59.676000+00:00",
        "Settings": {
            "ShowSpeakerLabels": true,
```

```
            "MaxSpeakerLabels": 2
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的識別發言者。

範例 4：轉錄音訊檔案，並在轉錄輸出中為任何不希望出現的字詞加上遮罩

以下 start-transcription-job 範例會轉錄音訊檔案，並使用您先前建立的詞彙篩選器來為任何不希望出現的字詞加上遮罩。

```
aws transcribe start-transcription-job \
    --cli-input-json file://myfourthfile.json
```

myfourthfile.json 的內容：

```
{
    "TranscriptionJobName": "cli-filter-mask-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
          "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "Settings":{
        "VocabularyFilterName": "your-vocabulary-filter",
        "VocabularyFilterMethod": "mask"
    }
}
```

輸出：

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-filter-mask-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "the-language-of-your-transcription-job",
        "Media": {
            "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
        },
        "StartTime": "2020-09-18T16:36:18.568000+00:00",
        "CreationTime": "2020-09-18T16:36:18.547000+00:00",
```

```
        "Settings": {
            "VocabularyFilterName": "your-vocabulary-filter",
            "VocabularyFilterMethod": "mask"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的篩選轉錄。

範例 5：轉錄音訊檔案，並移除轉錄輸出中任何不希望出現的字詞

以下 start-transcription-job 範例會轉錄音訊檔案，並使用您先前建立的詞彙篩選器來為任何不希望出現的字詞加上遮罩。

```
aws transcribe start-transcription-job \
    --cli-input-json file://myfifthfile.json
```

myfifthfile.json 的內容：

```
{
    "TranscriptionJobName": "cli-filter-remove-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-file-
name.file-extension"
    },
    "Settings":{
        "VocabularyFilterName": "your-vocabulary-filter",
        "VocabularyFilterMethod": "remove"
    }
}
```

輸出：

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-filter-remove-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "the-language-of-your-transcription-job",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
```

```
        },
        "StartTime": "2020-09-18T16:36:18.568000+00:00",
        "CreationTime": "2020-09-18T16:36:18.547000+00:00",
        "Settings": {
            "VocabularyFilterName": "your-vocabulary-filter",
            "VocabularyFilterMethod": "remove"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的篩選轉錄。

範例 6：使用自訂詞彙以更準確的方式轉錄音訊檔案

以下 start-transcription-job 範例會轉錄音訊檔案，並使用您先前建立的詞彙篩選器來為任何不希望出現的字詞加上遮罩。

```
aws transcribe start-transcription-job \
    --cli-input-json file://mysixthfile.json
```

mysixthfile.json 的內容：

```
{
    "TranscriptionJobName": "cli-vocab-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-file-
name.file-extension"
    },
    "Settings":{
        "VocabularyName": "your-vocabulary"
    }
}
```

輸出：

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-vocab-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "the-language-of-your-transcription-job",
        "Media": {
```

```
            "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
        },
        "StartTime": "2020-09-18T16:36:18.568000+00:00",
        "CreationTime": "2020-09-18T16:36:18.547000+00:00",
        "Settings": {
            "VocabularyName": "your-vocabulary"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的篩選轉錄。

範例 7：識別音訊檔案的語言並進行轉錄

以下 start-transcription-job 範例會轉錄音訊檔案，並使用您先前建立的詞彙篩選器來為任何不希望出現的字詞加上遮罩。

```
aws transcribe start-transcription-job \
    --cli-input-json file://myseventhfile.json
```

myseventhfile.json 的內容：

```
{
    "TranscriptionJobName": "cli-identify-language-transcription-job",
    "IdentifyLanguage": true,
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-file-
name.file-extension"
    }
}
```

輸出：

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-identify-language-transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
        },
```

```
        "StartTime": "2020-09-18T22:27:23.970000+00:00",
        "CreationTime": "2020-09-18T22:27:23.948000+00:00",
        "IdentifyLanguage": true
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的識別語言。

範例 8：轉錄包含已修訂的個人身分識別資訊的音訊檔案

以下 `start-transcription-job` 範例會轉錄音訊檔案，並修訂轉錄輸出中的任何個人身分識別資訊。

```
aws transcribe start-transcription-job \
    --cli-input-json file://myeighthfile.json
```

myeigthfile.json 的內容：

```
{
    "TranscriptionJobName": "cli-redaction-job",
    "LanguageCode": "language-code",
    "Media": {
        "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
    },
    "ContentRedaction": {
        "RedactionOutput":"redacted",
        "RedactionType":"PII"
    }
}
```

輸出：

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-redaction-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
        },
        "StartTime": "2020-09-25T23:49:13.195000+00:00",
        "CreationTime": "2020-09-25T23:49:13.176000+00:00",
```

```
        "ContentRedaction": {
            "RedactionType": "PII",
            "RedactionOutput": "redacted"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的自動內容修訂。

範例 9：產生包含已修訂個人身分識別資訊 (PII) 的文字記錄和未修訂的文字記錄

以下 start-transcription-job 範例會產生音訊檔案的兩個轉錄，一個包含已修訂的個人身分識別資訊，另一個則不包含任何修訂。

```
aws transcribe start-transcription-job \
    --cli-input-json file://myninthfile.json
```

myninthfile.json 的內容：

```
{
    "TranscriptionJobName": "cli-redaction-job-with-unredacted-transcript",
    "LanguageCode": "language-code",
    "Media": {
          "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
        },
    "ContentRedaction": {
        "RedactionOutput":"redacted_and_unredacted",
        "RedactionType":"PII"
    }
}
```

輸出：

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-redaction-job-with-unredacted-transcript",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
        },
        "StartTime": "2020-09-25T23:59:47.677000+00:00",
        "CreationTime": "2020-09-25T23:59:47.653000+00:00",
```

```
        "ContentRedaction": {
            "RedactionType": "PII",
            "RedactionOutput": "redacted_and_unredacted"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的自動內容修訂。

範例 10：使用您先前建立的自訂語言模型來轉錄音訊檔案。

以下 start-transcription-job 範例會使用您先前建立的自訂語言模型來轉錄音訊檔案。

```
aws transcribe start-transcription-job \
    --cli-input-json file://mytenthfile.json
```

mytenthfile.json 的內容：

```
{
    "TranscriptionJobName": "cli-clm-2-job-1",
    "LanguageCode": "language-code",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.file-extension"
    },
    "ModelSettings": {
        "LanguageModelName":"cli-clm-2"
    }
}
```

輸出：

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-clm-2-job-1",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.file-
extension"
        },
        "StartTime": "2020-09-28T17:56:01.835000+00:00",
        "CreationTime": "2020-09-28T17:56:01.801000+00:00",
        "ModelSettings": {
```

```
            "LanguageModelName": "cli-clm-2"
        }
    }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的使用自訂語言模型改善特定領域的轉錄準確性。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 StartTranscriptionJob。

update-medical-vocabulary

以下程式碼範例顯示如何使用 update-medical-vocabulary。

AWS CLI

使用新詞彙更新醫療自訂詞彙。

下列update-medical-vocabulary範例會將醫療自訂詞彙中使用的詞彙取代為新的詞彙。先決條件：若要取代醫療自訂詞彙中的詞彙，您需要具有新詞彙的檔案。

```
aws transcribe update-medical-vocabulary \
    --vocabulary-file-uri s3://amzn-s3-demo-bucket/Amazon-S3-Prefix/medical-custom-vocabulary.txt \
    --vocabulary-name medical-custom-vocabulary \
    --language-code language
```

輸出：

```
{
    "VocabularyName": "medical-custom-vocabulary",
    "LanguageCode": "en-US",
    "VocabularyState": "PENDING"
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的醫學自訂詞彙。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateMedicalVocabulary。

update-vocabulary-filter

以下程式碼範例顯示如何使用 update-vocabulary-filter。

AWS CLI

若要取代詞彙篩選條件中的字詞

下列update-vocabulary-filter範例會將詞彙篩選條件中的單字取代為新的單字。先決條件：若要使用新單字更新詞彙篩選條件，您必須將這些單字儲存為文字檔案。

```
aws transcribe update-vocabulary-filter \
    --vocabulary-filter-file-uri s3://amzn-s3-demo-bucket/Amazon-S3-Prefix/your-text-file-to-update-your-vocabulary-filter.txt \
    --vocabulary-filter-name vocabulary-filter-name
```

輸出：

```
{
    "VocabularyFilterName": "vocabulary-filter-name",
    "LanguageCode": "language-code",
    "LastModifiedTime": "2020-09-23T18:40:35.139000+00:00"
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的篩選不需要的字詞。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateVocabularyFilter。

update-vocabulary

以下程式碼範例顯示如何使用 update-vocabulary。

AWS CLI

使用新用語更新自訂詞彙。

以下 update-vocabulary 範例會以您提供的新用語覆寫用來建立自訂詞彙的用語。先決條件：若要取代自訂詞彙中的用語，您需有包含新用語的檔案。

```
aws transcribe update-vocabulary \
    --vocabulary-file-uri s3://amzn-s3-demo-bucket/Amazon-S3-Prefix/custom-vocabulary.txt \
    --vocabulary-name custom-vocabulary \
    --language-code language-code
```

輸出：

```
{
    "VocabularyName": "custom-vocabulary",
    "LanguageCode": "language",
    "VocabularyState": "PENDING"
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的自訂詞彙。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateVocabulary。

使用 的 Amazon Translate 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon Translate 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`import-terminology`

以下程式碼範例顯示如何使用 `import-terminology`。

AWS CLI

從檔案匯入自訂術語

下列`import-terminology`範例`MyTestTerminology`會從 `test-terminology.csv` 檔案建立名為 的術語：

```
aws translate import-terminology \
    --name MyTestTerminology \
    --description "Creating a test terminology in AWS Translate" \
    --merge-strategy OVERWRITE \
    --data-file fileb://test-terminology.csv \
```

```
    --terminology-data Format=CSV
```

test-terminology.csv 的內容：

en，fr，es，zh Hello world！，Bonjour tout le monde！，Hola Mundo！，？？？？？ Amazon、Amazon、Amazon、Amazon

輸出：

```
{
    "TerminologyProperties": {
        "SourceLanguageCode": "en",
        "Name": "MyTestTerminology",
        "TargetLanguageCodes": [
            "fr",
            "es",
            "zh"
        ],
        "SizeBytes": 97,
        "LastUpdatedAt": 1571089500.851,
        "CreatedAt": 1571089500.851,
        "TermCount": 6,
        "Arn": "arn:aws:translate:us-west-2:123456789012:terminology/
MyTestTerminology/LATEST",
        "Description": "Creating a test terminology in AWS Translate"
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ImportTerminology。

Trusted Advisor 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 Trusted Advisor。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`get-organization-recommendation`

以下程式碼範例顯示如何使用 `get-organization-recommendation`。

AWS CLI

取得組織建議

下列`get-organization-recommendation`範例會依其識別符取得組織建議。

```
aws trustedadvisor get-organization-recommendation \
    --organization-recommendation-identifier arn:aws:trustedadvisor:::organization-recommendation/9534ec9b-bf3a-44e8-8213-2ed68b39d9d5
```

輸出：

```
{
    "organizationRecommendation": {
        "arn": "arn:aws:trustedadvisor:::organization-recommendation/9534ec9b-bf3a-44e8-8213-2ed68b39d9d5",
        "name": "Lambda Runtime Deprecation Warning",
        "description": "One or more lambdas are using a deprecated runtime",
        "awsServices": [
            "lambda"
        ],
        "checkArn": "arn:aws:trustedadvisor:::check/L4dfs2Q4C5",
        "id": "9534ec9b-bf3a-44e8-8213-2ed68b39d9d5",
        "lifecycleStage": "resolved",
        "pillars": [
            "security"
        ],
        "resourcesAggregates": {
            "errorCount": 0,
            "okCount": 0,
            "warningCount": 0
        },
        "source": "ta_check",
        "status": "warning",
        "type": "priority"
    }
}
```

如需詳細資訊，請參閱 Trusted Advisor 使用者指南中的 Trusted Advisor API 入門。 AWS Trusted Advisor

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetOrganizationRecommendation。

get-recommendation

以下程式碼範例顯示如何使用 get-recommendation。

AWS CLI

取得建議

下列get-recommendation範例會依其識別符取得建議。

```
aws trustedadvisor get-recommendation \
    --recommendation-
identifier arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578
```

輸出：

```
{
    "recommendation": {
        "arn": "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
        "name": "MFA Recommendation",
        "description": "Enable multi-factor authentication",
        "awsServices": [
            "iam"
        ],
        "checkArn": "arn:aws:trustedadvisor:::check/7DAFEmoDos",
        "id": "55fa4d2e-bbb7-491a-833b-5773e9589578",
        "lastUpdatedAt": "2023-11-01T15:57:58.673Z",
        "pillarSpecificAggregates": {
            "costOptimizing": {
                "estimatedMonthlySavings": 0.0,
                "estimatedPercentMonthlySavings": 0.0
            }
        },
        "pillars": [
            "security"
        ],
```

```
        "resourcesAggregates": {
            "errorCount": 1,
            "okCount": 0,
            "warningCount": 0
        },
        "source": "ta_check",
        "status": "error",
        "type": "standard"
    }
}
```

如需詳細資訊，請參閱 Trusted Advisor 使用者指南中的 Trusted Advisor API 入門。 AWS Trusted Advisor

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRecommendation。

`list-checks`

以下程式碼範例顯示如何使用 `list-checks`。

AWS CLI

列出 Trusted Advisor 檢查

下列`list-checks`範例列出所有 Trusted Advisor 檢查。

```
aws trustedadvisor list-checks
```

輸出：

```
{
    "checkSummaries": [
        {
            "arn": "arn:aws:trustedadvisor:::check/1iG5NDGVre",
            "awsServices": [
                "EC2"
            ],
            "description": "Checks security groups for rules that allow unrestricted
 access to a resource. Unrestricted access increases opportunities for malicious
 activity (hacking, denial-of-service attacks, loss of data)",
            "id": "1iG5NDGVre",
            "metadata": {
                "0": "Region",
```

```
                "1": "Security Group Name",
                "2": "Security Group ID",
                "3": "Protocol",
                "4": "Port",
                "5": "Status",
                "6": "IP Range"
            },
            "name": "Security Groups - Unrestricted Access",
            "pillars": [
                "security"
            ],
            "source": "ta_check"
        },
        {
            "arn": "arn:aws:trustedadvisor:::check/1qazXsw23e",
            "awsServices": [
                "RDS"
            ],
            "description": "Checks your usage of RDS and provides recommendations
on purchase of Reserved Instances to help reduce costs incurred from using RDS
On-Demand. AWS generates these recommendations by analyzing your On-Demand usage
for the past 30 days. We then simulate every combination of reservations in the
generated category of usage in order to identify the best number of each type
of Reserved Instance to purchase to maximize your savings. This check covers
recommendations based on partial upfront payment option with 1-year or 3-year
commitment. This check is not available to accounts linked in Consolidated Billing.
Recommendations are only available for the Paying Account.",
            "id": "1qazXsw23e",
            "metadata": {
                "0": "Region",
                "1": "Family",
                "2": "Instance Type",
                "3": "License Model",
                "4": "Database Edition",
                "5": "Database Engine",
                "6": "Deployment Option",
                "7": "Recommended number of Reserved Instances to purchase",
                "8": "Expected Average Reserved Instance Utilization",
                "9": "Estimated Savings with Recommendation (monthly)"
                "10": "Upfront Cost of Reserved Instances",
                "11": "Estimated cost of Reserved Instances (monthly)",
                "12": "Estimated On-Demand Cost Post Recommended Reserved Instance
Purchase (monthly)",
                "13": "Estimated Break Even (months)",
```

```
                "14": "Lookback Period (days)",
                "15": "Term (years)"
            },
            "name": "Amazon Relational Database Service (RDS) Reserved Instance
Optimization",
            "pillars": [
                "cost_optimizing"
            ],
            "source": "ta_check"
        },
        {
            "arn": "arn:aws:trustedadvisor:::check/1qw23er45t",
            "awsServices": [
                "Redshift"
            ],
            "description": "Checks your usage of Redshift and provides
recommendations on purchase of Reserved Nodes to help reduce costs incurred from
using Redshift On-Demand. AWS generates these recommendations by analyzing your
On-Demand usage for the past 30 days. We then simulate every combination of
reservations in the generated category of usage in order to identify the best
number of each type of Reserved Nodes to purchase to maximize your savings. This
check covers recommendations based on partial upfront payment option with 1-year or
3-year commitment. This check is not available to accounts linked in Consolidated
Billing. Recommendations are only available for the Paying Account.",
            "id": "1qw23er45t",
            "metadata": {
                "0": "Region",
                "1": "Family",
                "2": "Node Type",
                "3": "Recommended number of Reserved Nodes to purchase",
                "4": "Expected Average Reserved Node Utilization",
                "5": "Estimated Savings with Recommendation (monthly)",
                "6": "Upfront Cost of Reserved Nodes",
                "7": "Estimated cost of Reserved Nodes (monthly)",
                "8": "Estimated On-Demand Cost Post Recommended Reserved Nodes
Purchase (monthly)",
                "9": "Estimated Break Even (months)",
                "10": "Lookback Period (days)",
                "11": "Term (years)",
            },
            "name": "Amazon Redshift Reserved Node Optimization",
            "pillars": [
                "cost_optimizing"
            ],
```

```
            "source": "ta_check"
        },
    ],
    "nextToken": "REDACTED"
}
```

如需詳細資訊，請參閱 Trusted Advisor 使用者指南中的 Trusted Advisor API 入門。 AWS Trusted Advisor

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListChecks。

`list-organization-recommendation-accounts`

以下程式碼範例顯示如何使用 `list-organization-recommendation-accounts`。

AWS CLI

列出組織建議帳戶

下列`list-organization-recommendation-accounts`範例會依其識別符列出組織建議的所有帳戶建議摘要。

```
aws trustedadvisor list-organization-recommendation-accounts \
    --organization-recommendation-identifier arn:aws:trustedadvisor:::organization-recommendation/9534ec9b-bf3a-44e8-8213-2ed68b39d9d5
```

輸出：

```
{
    "accountRecommendationLifecycleSummaries": [{
        "accountId": "000000000000",
        "accountRecommendationArn":
 "arn:aws:trustedadvisor::000000000000:recommendation/9534ec9b-
bf3a-44e8-8213-2ed68b39d9d5",
        "lifecycleStage": "resolved",
        "updateReason": "Resolved issue",
        "updateReasonCode": "valid_business_case",
        "lastUpdatedAt": "2023-01-17T18:25:44.552Z"
    }],
    "nextToken": "REDACTED"
}
```

如需詳細資訊，請參閱 Trusted Advisor 使用者指南中的 Trusted Advisor API 入門。 AWS Trusted Advisor

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListOrganizationRecommendationAccounts。

`list-organization-recommendation-resources`

以下程式碼範例顯示如何使用 `list-organization-recommendation-resources`。

AWS CLI

列出組織建議資源

下列`list-organization-recommendation-resources`範例會依組織識別符列出組織建議的所有資源。

```
aws trustedadvisor list-organization-recommendation-resources \
    --organization-recommendation-identifier arn:aws:trustedadvisor:::organization-recommendation/5a694939-2e54-45a2-ae72-730598fa89d0
```

輸出：

```
{
    "organizationRecommendationResourceSummaries": [
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-resource/5a694939-2e54-45a2-ae72-730598fa89d0/bb38affc0ce0681d9a6cd13f30238ba03a8f63dfe7a379dc403c619119d86af",
            "awsResourceId": "database-1-instance-1",
            "id": "bb38affc0ce0681d9a6cd13f302383ba03a8f63dfe7a379dc403c619119d86af",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
                "0": "14",
                "1": "208.79999999999998",
                "2": "database-1-instance-1",
                "3": "db.r5.large",
                "4": "false",
                "5": "us-west-2",
                "6": "arn:aws:rds:us-west-2:000000000000:db:database-1-instance-1",
                "7": "1"
            },
```

```
            "recommendationArn": "arn:aws:trustedadvisor:::organization-
recommendation/5a694939-2e54-45a2-ae72-730598fa89d0",
            "regionCode": "us-west-2",
            "status": "warning"
        },
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/5a694939-2e54-45a2-
ae72-730598fa89d0/51fded4d7a3278818df9cfe344ff5762cec46c095a6763d1ba1ba53bd0e1b0e6",
            "awsResourceId": "database-1",
            "id":
 "51fded4d7a3278818df9cfe344ff5762cec46c095a6763d1ba1ba53bd0e1b0e6",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
                "0": "14",
                "1": "31.679999999999996",
                "2": "database-1",
                "3": "db.t3.small",
                "4": "false",
                "5": "us-west-2",
                "6": "arn:aws:rds:us-west-2:000000000000:db:database-1",
                "7": "20"
            },
            "recommendationArn": "arn:aws:trustedadvisor:::organization-
recommendation/5a694939-2e54-45a2-ae72-730598fa89d0",
            "regionCode": "us-west-2",
            "status": "warning"
        },
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/5a694939-2e54-45a2-ae72-730598fa89d0/
f4d01bd20f4cd5372062aafc8786c489e48f0ead7cdab121463bf9f89e40a36b",
            "awsResourceId": "database-2-instance-1-us-west-2a",
            "id":
 "f4d01bd20f4cd5372062aafc8786c489e48f0ead7cdab121463bf9f89e40a36b",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
                "0": "14",
                "1": "187.20000000000002",
                "2": "database-2-instance-1-us-west-2a",
                "3": "db.r6g.large",
                "4": "true",
                "5": "us-west-2",
```

```
                "6": "arn:aws:rds:us-west-2:000000000000:db:database-2-instance-1-
us-west-2a",
                "7": "1"
            },
            "recommendationArn": "arn:aws:trustedadvisor:::organization-
recommendation/5a694939-2e54-45a2-ae72-730598fa89d0",
            "regionCode": "us-west-2",
            "status": "warning"
        },
    ],
    "nextToken": "REDACTED"
}
```

如需詳細資訊，請參閱 Trusted Advisor 使用者指南中的 Trusted Advisor API 入門。 AWS Trusted Advisor

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListOrganizationRecommendationResources。

`list-organization-recommendations`

以下程式碼範例顯示如何使用 `list-organization-recommendations`。

AWS CLI

範例 1：列出組織建議

下列`list-organization-recommendations`範例列出所有組織建議，但不包含篩選條件。

```
aws trustedadvisor list-organization-recommendations
```

輸出：

```
{
    "organizationRecommendationSummaries": [
        {
            "arn": "arn:aws:trustedadvisor:::organization-recommendation/9534ec9b-
bf3a-44e8-8213-2ed68b39d9d5",
            "name": "Lambda Runtime Deprecation Warning",
            "awsServices": [
                "lambda"
            ],
```

```
            "checkArn": "arn:aws:trustedadvisor:::check/L4dfs2Q4C5",
            "id": "9534ec9b-bf3a-44e8-8213-2ed68b39d9d5",
            "lifecycleStage": "resolved",
            "pillars": [
                "security"
            ],
            "resourcesAggregates": {
                "errorCount": 0,
                "okCount": 0,
                "warningCount": 0
            },
            "source": "ta_check",
            "status": "warning",
            "type": "priority"
        },
        {
            "arn": "arn:aws:trustedadvisor:::organization-
recommendation/4ecff4d4-1bc1-4c99-a5b8-0fff9ee500d6",
            "name": "Lambda Runtime Deprecation Warning",
            "awsServices": [
                "lambda"
            ],
            "checkArn": "arn:aws:trustedadvisor:::check/L4dfs2Q4C5",
            "id": "4ecff4d4-1bc1-4c99-a5b8-0fff9ee500d6",
            "lifecycleStage": "resolved",
            "pillars": [
                "security"
            ],
            "resourcesAggregates": {
                "errorCount": 0,
                "okCount": 0,
                "warningCount": 0
            },
            "source": "ta_check",
            "status": "warning",
            "type": "priority"
        },
    ],
    "nextToken": "REDACTED"
}
```

如需詳細資訊，請參閱 Trusted Advisor 使用者指南中的 Trusted Advisor API 入門。 AWS Trusted Advisor

範例 2：使用篩選條件列出組織建議

下列list-organization-recommendations範例會篩選並傳回最多一個屬於「安全」支柱的組織建議。

```
aws trustedadvisor list-organization-recommendations \
    --pillar security \
    --max-items 100
```

輸出：

```
{
    "organizationRecommendationSummaries": [{
        "arn": "arn:aws:trustedadvisor:::organization-recommendation/9534ec9b-
bf3a-44e8-8213-2ed68b39d9d5",
        "name": "Lambda Runtime Deprecation Warning",
        "awsServices": [
            "lambda"
        ],
        "checkArn": "arn:aws:trustedadvisor:::check/L4dfs2Q4C5",
        "id": "9534ec9b-bf3a-44e8-8213-2ed68b39d9d5",
        "lifecycleStage": "resolved",
        "pillars": [
            "security"
        ],
        "resourcesAggregates": {
            "errorCount": 0,
            "okCount": 0,
            "warningCount": 0
        },
        "source": "ta_check",
        "status": "warning",
        "type": "priority"
    }],
    "nextToken": "REDACTED"
}
```

如需詳細資訊，請參閱 Trusted Advisor 使用者指南中的 Trusted Advisor API 入門。 AWS Trusted Advisor

範例 3：使用分頁字符列出組織建議

下列list-organization-recommendations範例使用先前請求傳回的「nextToken」來擷取組織建議的下一頁。

```
aws trustedadvisor list-organization-recommendations \
    --pillar security \
    --max-items 100 \
    --starting-token <next-token>
```

輸出：

```
{
    "organizationRecommendationSummaries": [{
        "arn": "arn:aws:trustedadvisor:::organization-
recommendation/4ecff4d4-1bc1-4c99-a5b8-0fff9ee500d6",
        "name": "Lambda Runtime Deprecation Warning",
        "awsServices": [
            "lambda"
        ],
        "checkArn": "arn:aws:trustedadvisor:::check/L4dfs2Q4C5",
        "id": "4ecff4d4-1bc1-4c99-a5b8-0fff9ee500d6",
        "lifecycleStage": "resolved",
        "pillars": [
            "security"
        ],
        "resourcesAggregates": {
            "errorCount": 0,
            "okCount": 0,
            "warningCount": 0
        },
        "source": "ta_check",
        "status": "warning",
        "type": "priority"
    }]
}
```

如需詳細資訊，請參閱 Trusted Advisor 使用者指南中的 Trusted Advisor API 入門。 AWS Trusted Advisor

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListOrganizationRecommendations。

list-recommendation-resources

以下程式碼範例顯示如何使用 list-recommendation-resources。

AWS CLI

列出建議資源

下列list-recommendation-resources範例會依其識別符列出建議的所有資源。

```
aws trustedadvisor list-recommendation-resources \
    --recommendation-
identifier arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578
```

輸出：

```
{
    "recommendationResourceSummaries": [
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/55fa4d2e-
bbb7-491a-833b-5773e9589578/18959a1f1973cff8e706e9d9bde28bba36cd602a6b2cb86c8b61252835236010
            "id":
 "18959a1f1973cff8e706e9d9bde28bba36cd602a6b2cb86c8b61252835236010",
            "awsResourceId": "webcms-dev-01",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
                "0": "14",
                "1": "123.12000000000002",
                "2": "webcms-dev-01",
                "3": "db.m6i.large",
                "4": "false",
                "5": "us-east-1",
                "6": "arn:aws:rds:us-east-1:000000000000:db:webcms-dev-01",
                "7": "20"
            },
            "recommendationArn":
 "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
            "regionCode": "us-east-1",
            "status": "warning"
        },
        {
```

```
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/55fa4d2e-bbb7-491a-833b-5773e9589578/
e6367ff500ac90db8e4adeb4892e39ee9c36bbf812dcbce4b9e4fefcec9eb63e",
            "id":
 "e6367ff500ac90db8e4adeb4892e39ee9c36bbf812dcbce4b9e4fefcec9eb63e",
            "awsResourceId": "aws-dev-db-stack-instance-1",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
                "0": "14",
                "1": "29.52",
                "2": "aws-dev-db-stack-instance-1",
                "3": "db.t2.small",
                "4": "false",
                "5": "us-east-1",
                "6": "arn:aws:rds:us-east-1:000000000000:db:aws-dev-db-stack-
instance-1",
                "7": "1"
            },
            "recommendationArn":
 "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
            "regionCode": "us-east-1",
            "status": "warning"
        },
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/55fa4d2e-
bbb7-491a-833b-5773e9589578/31aa78ba050a5015d2d38cca7f5f1ce88f70857c4e1c3ad03f8f9fd95dad7459
            "id":
 "31aa78ba050a5015d2d38cca7f5f1ce88f70857c4e1c3ad03f8f9fd95dad7459",
            "awsResourceId": "aws-awesome-apps-stack-db",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
                "0": "14",
                "1": "114.48000000000002",
                "2": "aws-awesome-apps-stack-db",
                "3": "db.m6g.large",
                "4": "false",
                "5": "us-east-1",
                "6": "arn:aws:rds:us-east-1:000000000000:db:aws-awesome-apps-stack-
db",
                "7": "100"
            },
```

```
            "recommendationArn":
 "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
            "regionCode": "us-east-1",
            "status": "warning"
        }
    ],
    "nextToken": "REDACTED"
}
```

如需詳細資訊，請參閱 Trusted Advisor 使用者指南中的 Trusted Advisor API 入門。 AWS Trusted Advisor

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRecommendationResources。

`list-recommendations`

以下程式碼範例顯示如何使用 `list-recommendations`。

AWS CLI

範例 1：列出建議

下列`list-recommendations`範例列出所有建議，但不包含篩選條件。

```
aws trustedadvisor list-recommendations
```

輸出：

```
{
    "recommendationSummaries": [
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
            "name": "MFA Recommendation",
            "awsServices": [
                "iam"
            ],
            "checkArn": "arn:aws:trustedadvisor:::check/7DAFEmoDos",
            "id": "55fa4d2e-bbb7-491a-833b-5773e9589578",
            "lastUpdatedAt": "2023-11-01T15:57:58.673Z",
            "pillarSpecificAggregates": {
                "costOptimizing": {
```

```
                    "estimatedMonthlySavings": 0.0,
                    "estimatedPercentMonthlySavings": 0.0
                }
            },
            "pillars": [
                "security"
            ],
            "resourcesAggregates": {
                "errorCount": 1,
                "okCount": 0,
                "warningCount": 0
            },
            "source": "ta_check",
            "status": "error",
            "type": "standard"
        },
        {
            "arn":
 "arn:aws:trustedadvisor::000000000000:recommendation/8b602b6f-452d-4cb2-8a9e-
c7650955d9cd",
            "name": "RDS clusters quota warning",
            "awsServices": [
                "rds"
            ],
            "checkArn": "arn:aws:trustedadvisor:::check/gjqMBn6pjz",
            "id": "8b602b6f-452d-4cb2-8a9e-c7650955d9cd",
            "lastUpdatedAt": "2023-11-01T15:58:17.397Z",
            "pillarSpecificAggregates": {
                "costOptimizing": {
                    "estimatedMonthlySavings": 0.0,
                    "estimatedPercentMonthlySavings": 0.0
                }
            },
            "pillars": [
                "service_limits"
            ],
            "resourcesAggregates": {
                "errorCount": 0,
                "okCount": 3,
                "warningCount": 6
            },
            "source": "ta_check",
            "status": "warning",
            "type": "standard"
```

```
        }
    ],
    "nextToken": "REDACTED"
}
```

如需詳細資訊，請參閱 Trusted Advisor 使用者指南中的 Trusted Advisor API 入門。 AWS Trusted Advisor

範例 2：使用篩選條件列出建議

下列list-recommendations範例列出建議並包含篩選條件。

```
aws trustedadvisor list-recommendations \
    --aws-service iam \
    --max-items 100
```

輸出：

```
{
    "recommendationSummaries": [{
        "arn": "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
        "name": "MFA Recommendation",
        "awsServices": [
            "iam"
        ],
        "checkArn": "arn:aws:trustedadvisor:::check/7DAFEmoDos",
        "id": "55fa4d2e-bbb7-491a-833b-5773e9589578",
        "lastUpdatedAt": "2023-11-01T15:57:58.673Z",
        "pillarSpecificAggregates": {
            "costOptimizing": {
                "estimatedMonthlySavings": 0.0,
                "estimatedPercentMonthlySavings": 0.0
            }
        },
        "pillars": [
            "security"
        ],
        "resourcesAggregates": {
            "errorCount": 1,
            "okCount": 0,
            "warningCount": 0
```

```
        },
        "source": "ta_check",
        "status": "error",
        "type": "standard"
    }],
    "nextToken": "REDACTED"
}
```

如需詳細資訊，請參閱 Trusted Advisor 使用者指南中的 Trusted Advisor API 入門。 AWS Trusted Advisor

範例 3：使用分頁字符列出建議

下列list-recommendations範例使用先前請求傳回的 "nextToken" 來擷取篩選的建議的下一頁。

```
aws trustedadvisor list-recommendations \
    --aws-service rds \
    --max-items 100 \
    --starting-token <next-token>
```

輸出：

```
{
    "recommendationSummaries": [{
        "arn":
 "arn:aws:trustedadvisor::000000000000:recommendation/8b602b6f-452d-4cb2-8a9e-
c7650955d9cd",
        "name": "RDS clusters quota warning",
        "awsServices": [
            "rds"
        ],
        "checkArn": "arn:aws:trustedadvisor:::check/gjqMBn6pjz",
        "id": "8b602b6f-452d-4cb2-8a9e-c7650955d9cd",
        "lastUpdatedAt": "2023-11-01T15:58:17.397Z",
        "pillarSpecificAggregates": {
            "costOptimizing": {
                "estimatedMonthlySavings": 0.0,
                "estimatedPercentMonthlySavings": 0.0
            }
        },
        "pillars": [
```

```
            "service_limits"
        ],
        "resourcesAggregates": {
            "errorCount": 0,
            "okCount": 3,
            "warningCount": 6
        },
        "source": "ta_check",
        "status": "warning",
        "type": "standard"
    }]
}
```

如需詳細資訊，請參閱《Trusted Advisor 使用者指南》中的 Trusted Advisor API 入門。 AWS Trusted Advisor

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRecommendations。

update-organization-recommendation-lifecycle

以下程式碼範例顯示如何使用 update-organization-recommendation-lifecycle。

AWS CLI

更新組織建議生命週期

下列update-organization-recommendation-lifecycle範例會透過其識別符更新組織建議的生命週期。

```
aws trustedadvisor update-organization-recommendation-lifecycle \
    --organization-recommendation-identifier arn:aws:trustedadvisor:::organization-recommendation/96b5e5ca-7930-444c-90c6-06d386128100 \
    --lifecycle-stage dismissed \
    --update-reason-code not_applicable
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Trusted Advisor 使用者指南》中的 Trusted Advisor API 入門。 AWS Trusted Advisor

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateOrganizationRecommendationLifecycle。

update-recommendation-lifecycle

以下程式碼範例顯示如何使用 update-recommendation-lifecycle。

AWS CLI

更新建議生命週期

下列update-recommendation-lifecycle範例會依其識別符更新建議的生命週期。

```
aws trustedadvisor update-recommendation-lifecycle \
    --recommendation-
identifier arn:aws:trustedadvisor::000000000000:recommendation/861c9c6e-
f169-405a-8b59-537a8caccd7a \
    --lifecycle-stage resolved \
    --update-reason-code valid_business_case
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Trusted Advisor 使用者指南中的 Trusted Advisor API 入門。 AWS Trusted Advisor

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRecommendationLifecycle。

使用 的已驗證許可範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Verified Permissions 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

create-identity-source

以下程式碼範例顯示如何使用 create-identity-source。

AWS CLI

建立身分來源

下列create-identity-source範例會建立身分來源，可讓您參考存放在指定 Amazon Cognito 使用者集區中的身分。這些身分在 Verified Permissions 中可用為類型 的實體User。

```
aws verifiedpermissions create-identity-source \
    --configuration file://config.txt \
    --principal-entity-type "User" \
    --policy-store-id PSEXAMPLEabcdefg111111
```

config.txt 的內容：

```
{
    "cognitoUserPoolConfiguration": {
        "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_1a2b3c4d5",
        "clientIds":["a1b2c3d4e5f6g7h8i9j0kalbmc"]
    }
}
```

輸出：

```
{
    "createdDate": "2023-05-19T20:30:28.214829+00:00",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

如需身分來源的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的將 Amazon Verified Permissions 與身分提供者搭配使用。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateIdentitySource。

create-policy-store

以下程式碼範例顯示如何使用 create-policy-store。

AWS CLI

建立政策存放區

下列create-policy-store範例會在目前 AWS 區域中建立政策存放區。

```
aws verifiedpermissions create-policy-store \
    --validation-settings "mode=STRICT"
```

輸出：

```
{
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
    "createdDate": "2023-05-16T17:41:29.103459+00:00",
    "lastUpdatedDate": "2023-05-16T17:41:29.103459+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

如需政策存放區的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的 Amazon Verified Permissions 政策存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePolicyStore。

create-policy-template

以下程式碼範例顯示如何使用 create-policy-template。

AWS CLI

建立政策範本

下列create-policy-template範例會使用包含委託人預留位置的陳述式來建立政策範本。

```
aws verifiedpermissions create-policy-template \
    --statement file://template1.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

template1.txt 的內容：

```
permit(
```

```
    principal in ?principal,
    action == Action::"view",
    resource == Photo::"VacationPhoto94.jpg"
);
```

輸出：

```
{
    "createdDate": "2023-06-12T20:47:42.804511+00:00",
    "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111"
}
```

如需政策範本的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的 Amazon Verified Permissions 政策範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreatePolicyTemplate。

create-policy

以下程式碼範例顯示如何使用 create-policy。

AWS CLI

範例 1：建立靜態政策

下列create-policy範例會建立具有指定委託人和資源之政策範圍的靜態政策。

```
aws verifiedpermissions create-policy \
    --definition file://definition1.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

檔案 的內容definition1.txt：

```
{
    "static": {
        "description":  "Grant everyone of janeFriends UserGroup access to the
 vacationFolder Album",
        "statement": "permit(principal in UserGroup::\"janeFriends\", action,
 resource in Album::\"vacationFolder\" );"
    }
```

```
}
```

輸出：

```
{
    "createdDate": "2023-06-12T20:33:37.382907+00:00",
    "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
    "policyId": "SPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "STATIC",
    "principal": {
        "entityId": "janeFriends",
        "entityType": "UserGroup"
    },
    "resource": {
        "entityId": "vacationFolder",
        "entityType": "Album"
    }
}
```

範例 2：建立將資源存取權授予每個人的靜態政策

下列create-policy範例會建立具有僅指定資源之政策範圍的靜態政策。

```
aws verifiedpermissions create-policy \
    --definition file://definition2.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

檔案 的內容definition2.txt：

```
{
    "static": {
        "description":  "Grant everyone access to the publicFolder Album",
        "statement": "permit(principal, action, resource in Album::\"publicFolder
\");"
    }
}
```

輸出：

```
{
    "createdDate": "2023-06-12T20:39:44.975897+00:00",
```

```
    "lastUpdatedDate": "2023-06-12T20:39:44.975897+00:00",
    "policyId": "PbfR73F8oh5MMfr9uRtFDB",
    "policyStoreId": "PSEXAMPLEabcdefg222222",
    "policyType": "STATIC",
    "resource": {
        "entityId": "publicFolder",
        "entityType": "Album"
    }
}
```

範例 3：建立與指定範本相關聯的範本連結政策

下列create-policy範例會使用指定的政策範本建立範本連結政策，並將指定的委託人與新範本連結政策搭配使用。

```
aws verifiedpermissions create-policy \
    --definition file://definition.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

definition.txt 的內容：

```
{
    "templateLinked": {
        "policyTemplateId": "PTEXAMPLEabcdefg111111",
        "principal": {
            "entityType": "User",
            "entityId": "alice"
        }
    }
}
```

輸出：

```
{
    "createdDate": "2023-06-12T20:49:51.490211+00:00",
    "lastUpdatedDate": "2023-06-12T20:49:51.490211+00:00",
    "policyId": "TPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "TEMPLATE_LINKED",
    "principal": {
        "entityId": "alice",
        "entityType": "User"
```

```
    },
    "resource": {
        "entityId": "VacationPhoto94.jpg",
        "entityType": "Photo"
    }
}
```

如需政策的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的 Amazon Verified Permissions 政策。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreatePolicy。

delete-identity-source

以下程式碼範例顯示如何使用 delete-identity-source。

AWS CLI

刪除身分來源

下列delete-identity-source範例會刪除具有指定 ID 的身分來源。

```
aws verifiedpermissions delete-identity-source \
    --identity-source-id ISEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

此命令不會產生輸出。

如需身分來源的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的將 Amazon Verified Permissions 與身分提供者搭配使用。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteIdentitySource。

delete-policy-store

以下程式碼範例顯示如何使用 delete-policy-store。

AWS CLI

刪除政策存放區

下列delete-policy-store範例會刪除具有指定 ID 的政策存放區。

```
aws verifiedpermissions delete-policy-store \
    --policy-store-id PSEXAMPLEabcdefg111111
```

此命令不會產生輸出。

如需政策存放區的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的 Amazon Verified Permissions 政策存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePolicyStore。

delete-policy-template

以下程式碼範例顯示如何使用 delete-policy-template。

AWS CLI

刪除政策範本

下列delete-policy-template範例會刪除具有指定 ID 的政策範本。

```
aws verifiedpermissions delete-policy \
    --policy-template-id PTEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

此命令不會產生輸出。

如需政策範本的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的 Amazon Verified Permissions 政策範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeletePolicyTemplate。

delete-policy

以下程式碼範例顯示如何使用 delete-policy。

AWS CLI

刪除靜態或範本連結政策

下列delete-policy範例會刪除具有指定 ID 的政策。

```
aws verifiedpermissions delete-policy \
```

```
    --policy-id SPEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

此命令不會產生輸出。

如需政策的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的 Amazon Verified Permissions 政策。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeletePolicy。

get-identity-source

以下程式碼範例顯示如何使用 get-identity-source。

AWS CLI

擷取身分來源的詳細資訊

下列get-identity-source範例顯示具有指定 ID 之身分來源的詳細資訊。

```
aws verifiedpermissions get-identity-source \
    --identity-source  ISEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

輸出：

```
{
    "createdDate": "2023-06-12T22:27:49.150035+00:00",
    "details": {
        "clientIds": [ "a1b2c3d4e5f6g7h8i9j0kalbmc" ],
        "discoveryUrl": "https://cognito-idp.us-west-2.amazonaws.com/us-
west-2_1a2b3c4d5",
        "openIdIssuer": "COGNITO",
        "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_1a2b3c4d5"
    },
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-06-12T22:27:49.150035+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "principalEntityType": "User"
}
```

如需身分來源的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的將 Amazon Verified Permissions 與身分提供者搭配使用。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIdentitySource。

get-policy-store

以下程式碼範例顯示如何使用 get-policy-store。

AWS CLI

擷取政策存放區的詳細資訊

下列get-policy-store範例顯示具有指定 ID 的政策存放區詳細資訊。

```
aws verifiedpermissions get-policy-store \
    --policy-store-id PSEXAMPLEabcdefg111111
```

輸出：

```
{
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
    "createdDate": "2023-06-05T20:16:46.225598+00:00",
    "lastUpdatedDate": "2023-06-08T20:40:23.173691+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "validationSettings": { "mode": "OFF" }
}
```

如需政策存放區的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的 Amazon Verified Permissions 政策存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPolicyStore。

get-policy-template

以下程式碼範例顯示如何使用 get-policy-template。

AWS CLI

擷取政策範本的詳細資訊

下列get-policy-template範例顯示具有指定 ID 的政策範本詳細資訊。

```
aws verifiedpermissions get-policy-template \
    --policy-template-id  PTEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

輸出：

```
{
    "createdDate": "2023-06-12T20:47:42.804511+00:00",
    "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111",
    "statement": "permit(\n    principal in ?principal,\n    action == Action::
\"view\",\n    resource == Photo::\"VacationPhoto94.jpg\"\n);"
}
```

如需政策範本的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的 Amazon Verified Permissions 政策範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetPolicyTemplate。

get-policy

以下程式碼範例顯示如何使用 get-policy。

AWS CLI

擷取政策的詳細資訊

下列get-policy範例顯示具有指定 ID 的政策詳細資訊。

```
aws verifiedpermissions get-policy \
    --policy-id PSEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

輸出：

```
{
    "createdDate": "2023-06-12T20:33:37.382907+00:00",
    "definition": {
```

```
        "static": {
            "description": "Grant everyone of janeFriends UserGroup access to the
 vacationFolder Album",
            "statement": "permit(principal in UserGroup::\"janeFriends\", action,
 resource in Album::\"vacationFolder\" );"
        }
    },
    "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
    "policyId": "SPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "STATIC",
    "principal": {
        "entityId": "janeFriends",
        "entityType": "UserGroup"
    },
    "resource": {
        "entityId": "vacationFolder",
        "entityType": "Album"
    }
}
```

如需政策的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的 Amazon Verified Permissions 政策。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetPolicy。

get-schema

以下程式碼範例顯示如何使用 get-schema。

AWS CLI

擷取政策存放區中的結構描述

下列get-schema範例顯示指定政策存放區中結構描述的詳細資訊。

```
aws verifiedpermissions get-schema \
    --policy-store-id PSEXAMPLEabcdefg111111
```

輸出：

```
{
```

```
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "schema": "{\"MySampleNamespace\":{\"entityTypes\":{\"Employee\":{\"shape
\":{\"attributes\":{\"jobLevel\":{\"type\":\"Long\"},\"name\":{\"type\":\"String
\"}},\"type\":\"Record\"}}},\"actions\":{\"remoteAccess\":{\"appliesTo\":
{\"principalTypes\":[\"Employee\"]}}}}",
    "createdDate": "2023-06-14T17:47:13.999885+00:00",
    "lastUpdatedDate": "2023-06-14T17:47:13.999885+00:00"
}
```

如需結構描述的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的政策存放區結構描述。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSchema。

is-authorized-with-token

以下程式碼範例顯示如何使用 is-authorized-with-token。

AWS CLI

範例 1：請求使用者請求的授權決策 （ 允許 ）

下列is-authorized-with-token範例會為經過 Amazon Cognito 驗證的使用者請求授權決策。請求使用 Cognito 提供的身分字符，而不是存取字符。在此範例中，指定的資訊存放區設定為將主體傳回為類型 的實體CognitoUser。

```
aws verifiedpermissions is-authorized-with-token \
    --action actionId="View",actionType="Action" \
    --resource entityId="vacationPhoto94.jpg",entityType="Photo" \
    --policy-store-id PSEXAMPLEabcdefg111111 \
    --identity-token "AbCdE12345...long.string...54321EdCbA"
```

政策存放區包含具有下列陳述式的政策，可接受來自指定 Cognito 使用者集區和應用程式 ID 的身分。

```
permit(
    principal == CognitoUser::"us-east-1_1a2b3c4d5|a1b2c3d4e5f6g7h8i9j0kalbmc",
    action,
    resource == Photo::"VacationPhoto94.jpg"
);
```

輸出：

```
{
    "decision":"Allow",
    "determiningPolicies":[
        {
        "determiningPolicyId":"SPEXAMPLEabcdefg111111"
        }
    ],
    "errors":[]
}
```

如需從 Cognito 使用者集區使用身分的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的使用 Amazon Verified Permissions 搭配身分提供者。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 IsAuthorizedWithToken。

`is-authorized`

以下程式碼範例顯示如何使用 `is-authorized`。

AWS CLI

範例 1：請求使用者請求的授權決策（允許）

下列`is-authorized`範例會針對`User`名為 的委託人請求授權決策`Alice`，而委託人想要對`Photo`名為 的 資源執行 `updatePhoto`操作`VacationPhoto94.jpg`。

回應顯示一個政策允許請求。

```
aws verifiedpermissions is-authorized \
    --principal entityType=User,entityId=alice \
    --action actionType=Action,actionId=view \
    --resource entityType=Photo,entityId=VactionPhoto94.jpg \
    --policy-store-id PSEXAMPLEabcdefg111111
```

輸出：

```
{
    "decision": "ALLOW",
    "determiningPolicies": [
```

```
        {
            "policyId": "SPEXAMPLEabcdefg111111"
        }
    ],
    "errors": []
}
```

範例 2：請求使用者請求的授權決策（拒絕）

下列範例與先前的範例相同，但委託人是 User::"Bob"。政策存放區不包含允許該使用者存取 的任何政策Album::"alice_folder"。

輸出指出 Deny是隱含的，因為 的清單DeterminingPolicies是空的。

```
aws verifiedpermissions create-policy \
    --definition file://definition2.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

輸出：

```
{
    "decision": "DENY",
    "determiningPolicies": [],
    "errors": []
}
```

如需詳細資訊，請參閱 Amazon Verified Permissions 使用者指南。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 IsAuthorized。

list-identity-sources

以下程式碼範例顯示如何使用 list-identity-sources。

AWS CLI

列出可用的身分來源

下列list-identity-sources範例列出指定政策存放區中的所有身分來源。

```
aws verifiedpermissions list-identity-sources \
```

```
    --policy-store-id PSEXAMPLEabcdefg111111
```

輸出：

```
{
    "identitySources": [
        {
            "createdDate": "2023-06-12T22:27:49.150035+00:00",
            "details": {
                "clientIds": [ "a1b2c3d4e5f6g7h8i9j0kalbmc" ],
                "discoveryUrl": "https://cognito-idp.us-west-2.amazonaws.com/us-
west-2_1a2b3c4d5",
                "openIdIssuer": "COGNITO",
                "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/
us-west-2_1a2b3c4d5"
            },
            "identitySourceId": "ISEXAMPLEabcdefg111111",
            "lastUpdatedDate": "2023-06-12T22:27:49.150035+00:00",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "principalEntityType": "User"
        }
    ]
}
```

如需身分來源的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的將 Amazon Verified Permissions 與身分提供者搭配使用。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListIdentitySources。

list-policies

以下程式碼範例顯示如何使用 list-policies。

AWS CLI

列出可用的政策

下列list-policies範例列出指定政策存放區中的所有政策。

```
aws verifiedpermissions list-policies \
    --policy-store-id PSEXAMPLEabcdefg111111
```

輸出：

```
{
    "policies": [
        {
            "createdDate": "2023-06-12T20:33:37.382907+00:00",
            "definition": {
                "static": {
                    "description": "Grant everyone of janeFriends UserGroup access
 to the vacationFolder Album"
                }
            },
            "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
            "policyId": "SPEXAMPLEabcdefg111111",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "policyType": "STATIC",
            "principal": {
                "entityId": "janeFriends",
                "entityType": "UserGroup"
            },
            "resource": {
                "entityId": "vacationFolder",
                "entityType": "Album"
            }
        },
        {
            "createdDate": "2023-06-12T20:39:44.975897+00:00",
            "definition": {
                "static": {
                    "description": "Grant everyone access to the publicFolder Album"
                }
            },
            "lastUpdatedDate": "2023-06-12T20:39:44.975897+00:00",
            "policyId": "SPEXAMPLEabcdefg222222",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "policyType": "STATIC",
            "resource": {
                "entityId": "publicFolder",
                "entityType": "Album"
            }
        },
        {
            "createdDate": "2023-06-12T20:49:51.490211+00:00",
            "definition": {
```

```
                "templateLinked": {
                    "policyTemplateId": "PTEXAMPLEabcdefg111111"
                }
            },
            "lastUpdatedDate": "2023-06-12T20:49:51.490211+00:00",
            "policyId": "SPEXAMPLEabcdefg333333",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "policyType": "TEMPLATE_LINKED",
            "principal": {
                "entityId": "alice",
                "entityType": "User"
            },
            "resource": {
                "entityId": "VacationPhoto94.jpg",
                "entityType": "Photo"
            }
        }
    ]
}
```

如需政策的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的 Amazon Verified Permissions 政策。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListPolicies。

`list-policy-stores`

以下程式碼範例顯示如何使用 `list-policy-stores`。

AWS CLI

列出可用的政策存放區

下列`list-policy-stores`範例列出 AWS 區域中的所有政策存放區。Verified Permissions 的所有命令，但 `create-policy-store`和`list-policy-stores`需要您指定要使用的政策存放區的 ID。

```
aws verifiedpermissions list-policy-stores
```

輸出：

```
{
```

```
    "policyStores": [
        {
            "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
            "createdDate": "2023-06-05T20:16:46.225598+00:00",
            "policyStoreId": "PSEXAMPLEabcdefg111111"
        },
        {
            "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg222222",
            "createdDate": "2023-06-08T18:09:37.364356+00:00",
            "policyStoreId": "PSEXAMPLEabcdefg222222"
        },
        {
            "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg333333",
            "createdDate": "2023-06-08T18:09:46.920600+00:00",
            "policyStoreId": "PSEXAMPLEabcdefg333333"
        }
    ]
}
```

如需政策存放區的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的 Amazon Verified Permissions 政策存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPolicyStores。

`list-policy-templates`

以下程式碼範例顯示如何使用 `list-policy-templates`。

AWS CLI

列出可用的政策範本

下列`list-policy-templates`範例列出指定政策存放區中的所有政策範本。

```
aws verifiedpermissions list-policy-templates \
    --policy-store-id PSEXAMPLEabcdefg111111
```

輸出：

```
{
```

```
    "policyTemplates": [
        {
            "createdDate": "2023-06-12T20:47:42.804511+00:00",
            "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "policyTemplateId": "PTEXAMPLEabcdefg111111"
        }
    ]
}
```

如需政策範本的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的 Amazon Verified Permissions 政策範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListPolicyTemplates。

put-schema

以下程式碼範例顯示如何使用 put-schema。

AWS CLI

將結構描述儲存到政策存放區

下列put-schema範例會在指定的政策存放區中建立或取代結構描述。

輸入檔案中的 cedarJson 參數採用 JSON 物件的字串表示法。它包含最外部引號對內的內嵌引號 (")。這需要您將 JSON 轉換為字串，方法是將所有內嵌引號前面加上反斜線字元 (\")，並將所有行合併為沒有換行符號的單一文字行。

為便於閱讀，您可以在這裡顯示跨多行包裝的範例字串，但操作需要以單行字串形式提交參數。

aws verifiedpermissions put-schema --definition file：//schema.txt --policy-store-id PSEXAMPLEabcdefg111111

schema.txt 的內容：

```
{
    "cedarJson": "{\"MySampleNamespace\": {\"actions\": {\"remoteAccess\": {
            \"appliesTo\": {\"principalTypes\": [\"Employee\"]}},\"entityTypes\": {
            \"Employee\": {\"shape\": {\"attributes\": {\"jobLevel\": {\"type\":
            \"Long\"},\"name\": {\"type\": \"String\"}},\"type\": \"Record\"}}}}"
```

```
}
```

輸出：

```
{
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "namespaces": [
        "MySampleNamespace"
    ],
    "createdDate": "2023-06-14T17:47:13.999885+00:00",
    "lastUpdatedDate": "2023-06-14T17:47:13.999885+00:00"
}
```

如需結構描述的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的政策存放區結構描述。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutSchema。

update-identity-source

以下程式碼範例顯示如何使用 update-identity-source。

AWS CLI

更新身分來源

下列update-identity-source範例提供新的 Cognito 使用者集區組態，並變更身分來源傳回的實體類型，以修改指定的身分來源。

```
aws verifiedpermissions update-identity-source
    --identity-source-id ISEXAMPLEabcdefg111111 \
    --update-configuration file://config.txt \
    --principal-entity-type "Employee" \
    --policy-store-id PSEXAMPLEabcdefg111111
```

config.txt 的內容：

```
{
        "cognitoUserPoolConfiguration": {
                "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/
us-west-2_1a2b3c4d5",
```

```
                "clientIds":["a1b2c3d4e5f6g7h8i9j0kalbmc"]
        }
}
```

輸出：

```
{
    "createdDate": "2023-05-19T20:30:28.214829+00:00",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

如需身分來源的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的將 Amazon Verified Permissions 與身分提供者搭配使用。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateIdentitySource。

update-policy-store

以下程式碼範例顯示如何使用 update-policy-store。

AWS CLI

更新政策存放區

下列update-policy-store範例會透過變更政策存放區驗證設定來修改政策存放區。

```
aws verifiedpermissions update-policy-store \
    --validation-settings "mode=STRICT" \
    --policy-store-id PSEXAMPLEabcdefg111111
```

輸出：

```
{
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
    "createdDate": "2023-05-16T17:41:29.103459+00:00",
    "lastUpdatedDate": "2023-05-16T17:41:29.103459+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

如需政策存放區的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的 Amazon Verified Permissions 政策存放區。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePolicyStore。

update-policy-template

以下程式碼範例顯示如何使用 update-policy-template。

AWS CLI

範例 1：更新政策範本

下列update-policy-template範例會修改指定的範本連結政策，以取代其政策陳述式。

```
aws verifiedpermissions update-policy-template \
    --policy-template-id PTEXAMPLEabcdefg111111 \
    --statement file://template1.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

檔案 的內容template1.txt：

```
permit(
    principal in ?principal,
    action == Action::"view",
    resource == Photo::"VacationPhoto94.jpg"
);
```

輸出：

```
{
    "createdDate": "2023-06-12T20:47:42.804511+00:00",
    "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111"
}
```

如需政策範本的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的 Amazon Verified Permissions 政策範本。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePolicyTemplate。

update-policy

以下程式碼範例顯示如何使用 update-policy。

AWS CLI

更新靜態政策

下列update-policy範例透過更新其描述和陳述式來修改現有的靜態政策。

```
aws verifiedpermissions update-policy \
    --policy-id SPEXAMPLEabcdefg111111 \
    --definition file://updated-definition.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

statement 參數會取得 JSON 物件的字串表示法。它包含最外部引號對內的內嵌引號 (")。這需要您將 JSON 轉換為字串，方法是將所有內嵌引號前面加上反斜線字元 (\")，並將所有行合併為單一文字行，且沒有換行符號。

您可以顯示跨多行包裝的範例字串，以提高可讀性，但 操作需要以單行字串形式提交參數。

檔案 的內容updated-definition.txt：

```
{
    "static": {
        "description": "Updated policy to grant janeFriends UserGroup access to the
 vacationFolder Album with view action only",
        "statement": "permit(principal in UserGroup::\"janeFriends\", action ==
 Action::\"view\", resource in Album::\"vacationFolder\" );"
    }
}
```

輸出：

```
{
    "createdDate": "2023-06-12T20:33:37.382907+00:00",
    "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
    "policyId": "SPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "STATIC",
    "principal": {
        "entityId": "janeFriends",
```

```
        "entityType": "UserGroup"
    },
    "resource": {
        "entityId": "vacationFolder",
        "entityType": "Album"
    }
}
```

如需政策的詳細資訊，請參閱《Amazon Verified Permissions 使用者指南》中的 Amazon Verified Permissions 政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePolicy。

使用 的 VPC Lattice 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 VPC Lattice 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-listener`

以下程式碼範例顯示如何使用 `create-listener`。

AWS CLI

建立接聽程式

下列`create-listener`範例會使用預設規則建立 HTTPS 接聽程式，將流量轉送至指定的 VPC Lattice 目標群組。

```
aws vpc-lattice create-listener \
    --name my-service-listener \
```

```
    --protocol HTTPS \
    --port 443 \
    --service-identifier svc-0285b53b2eEXAMPLE \
    --default-action file://listener-config.json
```

listener-config.json 的內容：

```
{
    "forward": {
        "targetGroups": [
            {
                "targetGroupIdentifier": "tg-0eaa4b9ab4EXAMPLE"
            }
        ]
    }
}
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE/listener/listener-07cc7fb0abEXAMPLE",
    "defaultAction": {
        "forward": {
            "targetGroups": [
                {
                    "targetGroupIdentifier": "tg-0eaa4b9ab4EXAMPLE",
                    "weight": 100
                }
            ]
        }
    },
    "id": "listener-07cc7fb0abEXAMPLE",
    "name": "my-service-listener",
    "port": 443,
    "protocol": "HTTPS",
    "serviceArn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
    "serviceId": "svc-0285b53b2eEXAMPLE"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的接聽程式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateListener。

create-resource-configuration

以下程式碼範例顯示如何使用 create-resource-configuration。

AWS CLI

建立資源組態

下列create-resource-configuration範例會建立指定單一 IPv4 地址的資源組態。

```
aws vpc-lattice create-resource-configuration \
    --name my-resource-config \
    --type SINGLE \
    --resource-gateway-identifier rgw-0bba03f3d56060135 \
    --resource-configuration-definition 'ipResource={ipAddress=10.0.14.85}'
```

輸出：

```
{
    "allowAssociationToShareableServiceNetwork": true,
    "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourceconfiguration/
rcfg-07129f3acded87625",
    "id": "rcfg-07129f3acded87625",
    "name": "my-resource-config",
    "portRanges": [
        "1-65535"
    ],
    "protocol": "TCP",
    "resourceConfigurationDefinition": {
        "ipResource": {
            "ipAddress": "10.0.14.85"
        }
    },
    "resourceGatewayId": "rgw-0bba03f3d56060135",
    "status": "ACTIVE",
    "type": "SINGLE"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的 VPC 資源的資源組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateResourceConfiguration。

create-resource-gateway

以下程式碼範例顯示如何使用 create-resource-gateway。

AWS CLI

建立資源閘道

下列create-resource-gateway範例會為指定的子網路建立資源閘道。

```
aws vpc-lattice create-resource-gateway \
    --name my-resource-gateway \
    --vpc-identifier vpc-0bf4c2739bc05a69 \
    --subnet-ids subnet-08e8943905b63a683
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourcegateway/
rgw-0bba03f3d56060135",
    "id": "rgw-0bba03f3d56060135",
    "ipAddressType": "IPV4",
    "name": "my-resource-gateway",
    "securityGroupIds": [
        "sg-087ffd596c5fe962c"
    ],
    "status": "ACTIVE",
    "subnetIds": [
        "subnet-08e8943905b63a683"
    ],
    "vpcIdentifier": "vpc-0bf4c2739bc05a694"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的 VPC Lattice 中的資源閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateResourceGateway。

create-service-network-service-association

以下程式碼範例顯示如何使用 create-service-network-service-association。

AWS CLI

建立服務關聯

下列create-service-network-service-association範例會將指定的服務與指定的服務網路建立關聯。

```
aws vpc-lattice create-service-network-service-association \
    --service-identifier svc-0285b53b2eEXAMPLE \
    --service-network-identifier sn-080ec7dc93EXAMPLE
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-
east-2:123456789012:servicenetworkserviceassociation/snsa-0e16955a8cEXAMPLE",
    "createdBy": "123456789012",
    "dnsEntry": {
        "domainName": "my-lattice-service-0285b53b2eEXAMPLE.7d67968.vpc-lattice-
svcs.us-east-2.on.aws",
        "hostedZoneId": "Z09127221KTH2CEXAMPLE"
    },
    "id": "snsa-0e16955a8cEXAMPLE",
    "status": "CREATE_IN_PROGRESS"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的管理服務關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateServiceNetworkServiceAssociation。

create-service-network-vpc-association

以下程式碼範例顯示如何使用 create-service-network-vpc-association。

AWS CLI

建立 VPC 關聯

下列create-service-network-vpc-association範例會將指定的 vpc 與指定的服務網路建立關聯。指定的安全群組會控制 VPC 中的哪些資源可以存取服務網路及其服務。

```
aws vpc-lattice create-service-network-vpc-association \
```

```
    --vpc-identifier vpc-0a1b2c3d4eEXAMPLE \
    --service-network-identifier sn-080ec7dc93EXAMPLE \
    --security-group-ids sg-0aee16bc6cEXAMPLE
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/
snva-0821fc8631EXAMPLE",
    "createdBy": "123456789012",
    "id": "snva-0821fc8631EXAMPLE",
    "securityGroupIds": [
        "sg-0aee16bc6cEXAMPLE"
    ],
    "status": "CREATE_IN_PROGRESS"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的管理 VPC 關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateServiceNetworkVpcAssociation。

create-service-network

以下程式碼範例顯示如何使用 create-service-network。

AWS CLI

建立服務網路

下列create-service-network範例會建立具有指定名稱的服務網路。

```
aws vpc-lattice create-service-network \
    --name my-service-network
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
    "authType": "NONE",
    "id": "sn-080ec7dc93EXAMPLE",
```

```
    "name": "my-service-network"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的服務網路。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateServiceNetwork。

create-service

以下程式碼範例顯示如何使用 create-service。

AWS CLI

建立服務

下列create-service範例會建立具有指定名稱的服務。

```
aws vpc-lattice create-service \
    --name my-lattice-service
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
    "authType": "NONE",
    "dnsEntry": {
        "domainName": "my-lattice-service-0285b53b2eEXAMPLE.1a2b3c4.vpc-lattice-
svcs.us-east-2.on.aws",
        "hostedZoneId": "Z09127221KTH2CEXAMPLE"
    },
    "id": "svc-0285b53b2eEXAMPLE",
    "name": "my-lattice-service",
    "status": "CREATE_IN_PROGRESS"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的 VPC Lattice 中的服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateService。

create-target-group

以下程式碼範例顯示如何使用 create-target-group。

AWS CLI

範例 1：建立類型為 INSTANCE 的目標群組

下列create-target-group範例會建立具有指定名稱、類型和組態的目標群組。

```
aws vpc-lattice create-target-group \
    --name my-lattice-target-group-instance \
    --type INSTANCE \
    --config file://tg-config.json
```

tg-config.json 的內容：

```
{
    "port": 443,
    "protocol": "HTTPS",
    "protocolVersion": "HTTP1",
    "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
}
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
    "config": {
        "healthCheck": {
            "enabled": true,
            "healthCheckIntervalSeconds": 30,
            "healthCheckTimeoutSeconds": 5,
            "healthyThresholdCount": 5,
            "matcher": {
                "httpCode": "200"
            },
            "path": "/",
            "protocol": "HTTPS",
            "protocolVersion": "HTTP1",
            "unhealthyThresholdCount": 2
        },
        "port": 443,
        "protocol": "HTTPS",
        "protocolVersion": "HTTP1",
        "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
```

```
    },
    "id": "tg-0eaa4b9ab4EXAMPLE",
    "name": "my-lattice-target-group-instance",
    "status": "CREATE_IN_PROGRESS",
    "type": "INSTANCE"
}
```

範例 2：建立 IP 類型的目標群組

下列create-target-group範例會建立具有指定名稱、類型和組態的目標群組。

```
aws vpc-lattice create-target-group \
    --name my-lattice-target-group-ip \
    --type IP \
    --config file://tg-config.json
```

tg-config.json 的內容：

```
{
    "ipAddressType": "IPV4",
    "port": 443,
    "protocol": "HTTPS",
    "protocolVersion": "HTTP1",
    "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
}
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
    "config": {
        "healthCheck": {
            "enabled": true,
            "healthCheckIntervalSeconds": 30,
            "healthCheckTimeoutSeconds": 5,
            "healthyThresholdCount": 5,
            "matcher": {
                "httpCode": "200"
            },
            "path": "/",
            "protocol": "HTTPS",
```

```
            "protocolVersion": "HTTP1",
            "unhealthyThresholdCount": 2
        },
        "ipAddressType": "IPV4",
        "port": 443,
        "protocol": "HTTPS",
        "protocolVersion": "HTTP1",
        "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
    },
    "id": "tg-0eaa4b9ab4EXAMPLE",
    "name": "my-lattice-target-group-ip",
    "status": "CREATE_IN_PROGRESS",
    "type": "IP"
}
```

範例 3：建立 LAMBDA 類型的目標群組

下列create-target-group範例會建立具有指定名稱、類型和組態的目標群組。

```
aws vpc-lattice create-target-group \
    --name my-lattice-target-group-lambda \
    --type LAMBDA
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
    "id": "tg-0eaa4b9ab4EXAMPLE",
    "name": "my-lattice-target-group-lambda",
    "status": "CREATE_IN_PROGRESS",
    "type": "LAMBDA"
}
```

範例 4：建立類型為 ALB 的目標群組

下列create-target-group範例會建立具有指定名稱、類型和組態的目標群組。

```
aws vpc-lattice create-target-group \
    --name my-lattice-target-group-alb \
    --type ALB \
    --config file://tg-config.json
```

tg-config.json 的內容：

```
{
    "port": 443,
    "protocol": "HTTPS",
    "protocolVersion": "HTTP1",
    "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
}
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
    "config": {
        "port": 443,
        "protocol": "HTTPS",
        "protocolVersion": "HTTP1",
        "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
    },
    "id": "tg-0eaa4b9ab4EXAMPLE",
    "name": "my-lattice-target-group-alb",
    "status": "CREATE_IN_PROGRESS",
    "type": "ALB"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的目標群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateTargetGroup。

delete-auth-policy

以下程式碼範例顯示如何使用 delete-auth-policy。

AWS CLI

刪除身分驗證政策

下列delete-auth-policy範例會刪除指定服務的身分驗證政策。

```
aws vpc-lattice delete-auth-policy \
    --resource-identifier svc-0285b53b2eEXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的身分驗證政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAuthPolicy。

delete-listener

以下程式碼範例顯示如何使用 delete-listener。

AWS CLI

刪除接聽程式

下列delete-listener範例會刪除指定的接聽程式。

```
aws vpc-lattice delete-listener \
    --listener-identifier listener-07cc7fb0abEXAMPLE \
    --service-identifier svc-0285b53b2eEXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的接聽程式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteListener。

delete-resource-configuration

以下程式碼範例顯示如何使用 delete-resource-configuration。

AWS CLI

刪除資源組態

下列delete-resource-configuration範例會刪除指定的資源組態。

```
aws vpc-lattice delete-resource-configuration \
    --resource-configuration-identifier rcfg-07129f3acded87625
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的 VPC Lattice 中的資源閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteResourceConfiguration。

delete-resource-gateway

以下程式碼範例顯示如何使用 delete-resource-gateway。

AWS CLI

刪除資源閘道

下列delete-resource-gateway範例會刪除指定的資源閘道。

```
aws vpc-lattice delete-resource-gateway \
    --resource-gateway-identifier rgw-0bba03f3d56060135
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourcegateway/
rgw-0bba03f3d56060135",
    "id": "rgw-0bba03f3d56060135",
    "name": "my-resource-gateway",
    "status": "DELETE_IN_PROGRESS"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的 VPC Lattice 中的資源閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteResourceGateway。

delete-service-network-service-association

以下程式碼範例顯示如何使用 delete-service-network-service-association。

AWS CLI

刪除服務關聯

下列delete-service-network-service-association範例會取消指定服務關聯的關聯。

```
aws vpc-lattice delete-service-network-service-association \
```

```
    --service-network-service-association-identifier snsa-031fabb4d8EXAMPLE
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-
east-2:123456789012:servicenetworkserviceassociation/snsa-031fabb4d8EXAMPLE",
    "id": "snsa-031fabb4d8EXAMPLE",
    "status": "DELETE_IN_PROGRESS"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的管理服務關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteServiceNetworkServiceAssociation。

delete-service-network-vpc-association

以下程式碼範例顯示如何使用 delete-service-network-vpc-association。

AWS CLI

刪除 VPC 關聯

下列delete-service-network-vpc-association範例會取消與指定 VPC 關聯的關聯。

```
aws vpc-lattice delete-service-network-vpc-association \
    --service-network-vpc-association-identifier snva-0821fc8631EXAMPLE
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/
snva-0821fc8631EXAMPLE",
    "id": "snva-0821fc8631EXAMPLE",
    "status": "DELETE_IN_PROGRESS"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的管理 VPC 關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteServiceNetworkVpcAssociation。

delete-service-network

以下程式碼範例顯示如何使用 delete-service-network。

AWS CLI

刪除服務網路

下列delete-service-network範例會刪除指定的服務網路。

```
aws vpc-lattice delete-service-network \
    --service-network-identifier sn-080ec7dc93EXAMPLE
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的服務網路。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteServiceNetwork。

delete-service

以下程式碼範例顯示如何使用 delete-service。

AWS CLI

刪除服務

下列delete-service範例會刪除指定的服務。

```
aws vpc-lattice delete-service \
    --service-identifier svc-0285b53b2eEXAMPLE
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-west-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
    "id": "svc-0285b53b2eEXAMPLE",
    "name": "my-lattice-service",
    "status": "DELETE_IN_PROGRESS"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的 VPC Lattice 中的服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteService。

delete-target-group

以下程式碼範例顯示如何使用 delete-target-group。

AWS CLI

刪除目標群組

下列delete-target-group範例會刪除指定的目標群組。

```
aws vpc-lattice delete-target-group \
    --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
    "id": "tg-0eaa4b9ab4EXAMPLE",
    "status": "DELETE_IN_PROGRESS"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的目標群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTargetGroup。

deregister-targets

以下程式碼範例顯示如何使用 deregister-targets。

AWS CLI

取消註冊目標

下列deregister-targets範例會從指定的目標群組取消註冊指定的目標。

```
aws vpc-lattice deregister-targets \
    --targets i-07dd579bc5EXAMPLE \
```

```
    --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

輸出：

```
{
    "successful": [
        {
            "id": "i-07dd579bc5EXAMPLE",
            "port": 443
        }
    ],
    "unsuccessful": []
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的註冊目標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterTargets。

get-auth-policy

以下程式碼範例顯示如何使用 get-auth-policy。

AWS CLI

取得身分驗證政策的相關資訊

下列get-auth-policy範例取得指定服務之身分驗證政策的相關資訊。

```
aws vpc-lattice get-auth-policy \
    --resource-identifier svc-0285b53b2eEXAMPLE
```

輸出：

```
{
    "createdAt": "2023-06-07T03:51:20.266Z",
    "lastUpdatedAt": "2023-06-07T04:39:27.082Z",
    "policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:role/my-clients\"},
\"Action\":\"vpc-lattice-svcs:Invoke\",\"Resource\":\"arn:aws:vpc-lattice:us-
east-2:123456789012:service/svc-0285b53b2eEXAMPLE\"}]}",
    "state": "Active"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的身分驗證政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAuthPolicy。

get-listener

以下程式碼範例顯示如何使用 get-listener。

AWS CLI

取得服務接聽程式的相關資訊

下列get-listener範例取得指定服務之指定接聽程式的相關資訊。

```
aws vpc-lattice get-listener \
    --listener-identifier listener-0ccf55918cEXAMPLE \
    --service-identifier svc-0285b53b2eEXAMPLE
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE/listener/listener-0ccf55918cEXAMPLE",
    "createdAt": "2023-05-07T05:08:45.192Z",
    "defaultAction": {
        "forward": {
            "targetGroups": [
                {
                    "targetGroupIdentifier": "tg-0ff213abb6EXAMPLE",
                    "weight": 1
                }
            ]
        }
    },
    "id": "listener-0ccf55918cEXAMPLE",
    "lastUpdatedAt": "2023-05-07T05:08:45.192Z",
    "name": "http-80",
    "port": 80,
    "protocol": "HTTP",
    "serviceArn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
    "serviceId": "svc-0285b53b2eEXAMPLE"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的定義路由。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetListener。

get-resource-configuration

以下程式碼範例顯示如何使用 get-resource-configuration。

AWS CLI

取得資源組態的相關資訊

下列get-resource-configuration範例取得指定資源組態的相關資訊。

```
aws vpc-lattice get-resource-configuration \
    --resource-configuration-identifier rcfg-07129f3acded87625
```

輸出：

```
{
    "allowAssociationToShareableServiceNetwork": true,
    "amazonManaged": false,
    "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourceconfiguration/
rcfg-07129f3acded87625",
    "createdAt": "2025-02-01T00:57:35.871000+00:00",
    "id": "rcfg-07129f3acded87625",
    "lastUpdatedAt": "2025-02-01T00:57:46.874000+00:00",
    "name": "my-resource-config",
    "portRanges": [
        "1-65535"
    ],
    "protocol": "TCP",
    "resourceConfigurationDefinition": {
        "ipResource": {
            "ipAddress": "10.0.14.85"
        }
    },
    "resourceGatewayId": "rgw-0bba03f3d56060135",
    "status": "ACTIVE",
    "type": "SINGLE"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的 VPC Lattice 中的資源閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResourceConfiguration。

get-resource-gateway

以下程式碼範例顯示如何使用 get-resource-gateway。

AWS CLI

取得資源閘道的相關資訊

下列get-resource-gateway範例取得指定資源閘道的相關資訊。

```
aws vpc-lattice get-resource-gateway \
    --resource-gateway-identifier rgw-0bba03f3d56060135
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourcegateway/
rgw-0bba03f3d56060135",
    "createdAt": "2025-02-01T00:57:33.241000+00:00",
    "id": "rgw-0bba03f3d56060135",
    "ipAddressType": "IPV4",
    "lastUpdatedAt": "2025-02-01T00:57:44.351000+00:00",
    "name": "my-resource-gateway",
    "securityGroupIds": [
        "sg-087ffd596c5fe962c"
    ],
    "status": "ACTIVE",
    "subnetIds": [
        "subnet-08e8943905b63a683"
    ],
    "vpcId": "vpc-0bf4c2739bc05a694"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的 VPC Lattice 中的資源閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResourceGateway。

get-service-network-service-association

以下程式碼範例顯示如何使用 get-service-network-service-association。

AWS CLI

取得服務關聯的相關資訊

下列get-service-network-service-association範例會取得指定服務關聯的相關資訊。

```
aws vpc-lattice get-service-network-service-association \
    --service-network-service-association-identifier snsa-031fabb4d8EXAMPLE
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-
east-2:123456789012:servicenetworkserviceassociation/snsa-031fabb4d8EXAMPLE",
    "createdAt": "2023-05-05T21:48:16.076Z",
    "createdBy": "123456789012",
    "dnsEntry": {
        "domainName": "my-lattice-service-0285b53b2eEXAMPLE.7d67968.vpc-lattice-
svcs.us-east-2.on.aws",
        "hostedZoneId": "Z09127221KTH2CEXAMPLE"
    },
    "id": "snsa-031fabb4d8EXAMPLE",
    "serviceArn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
    "serviceId": "svc-0285b53b2eEXAMPLE",
    "serviceName": "my-lattice-service",
    "serviceNetworkArn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
    "serviceNetworkId": "sn-080ec7dc93EXAMPLE",
    "serviceNetworkName": "my-service-network",
    "status": "ACTIVE"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的管理服務關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetServiceNetworkServiceAssociation。

get-service-network-vpc-association

以下程式碼範例顯示如何使用 get-service-network-vpc-association。

AWS CLI

取得 VPC 關聯的相關資訊

下列get-service-network-vpc-association範例取得指定 VPC 關聯的相關資訊。

```
aws vpc-lattice get-service-network-vpc-association \
    --service-network-vpc-association-identifier snva-0821fc8631EXAMPLE
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/
snva-0821fc8631EXAMPLE",
    "createdAt": "2023-06-06T23:41:08.421Z",
    "createdBy": "123456789012",
    "id": "snva-0c5dcb60d6EXAMPLE",
    "lastUpdatedAt": "2023-06-06T23:41:08.421Z",
    "securityGroupIds": [
        "sg-0aee16bc6cEXAMPLE"
    ],
    "serviceNetworkArn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
    "serviceNetworkId": "sn-080ec7dc93EXAMPLE",
    "serviceNetworkName": "my-service-network",
    "status": "ACTIVE",
    "vpcId": "vpc-0a1b2c3d4eEXAMPLE"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的管理 VPC 關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetServiceNetworkVpcAssociation。

get-service-network

以下程式碼範例顯示如何使用 get-service-network。

AWS CLI

取得服務網路的相關資訊

下列get-service-network範例取得指定服務網路的相關資訊。

```
aws vpc-lattice get-service-network \
    --service-network-identifier sn-080ec7dc93EXAMPLE
```

輸出：

```
{
   "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
   "authType": "AWS_IAM",
   "createdAt": "2023-05-05T15:26:08.417Z",
   "id": "sn-080ec7dc93EXAMPLE",
   "lastUpdatedAt": "2023-05-05T15:26:08.417Z",
   "name": "my-service-network",
   "numberOfAssociatedServices": 2,
   "numberOfAssociatedVPCs": 3
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的服務網路。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetServiceNetwork。

get-service

以下程式碼範例顯示如何使用 get-service。

AWS CLI

取得服務的相關資訊

下列get-service範例會取得指定服務的相關資訊。

```
aws vpc-lattice get-service \
    --service-identifier svc-0285b53b2eEXAMPLE
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
    "authType": "AWS_IAM",
    "createdAt": "2023-05-05T21:35:29.339Z",
```

```
    "dnsEntry": {
        "domainName": "my-lattice-service-0285b53b2eEXAMPLE.7d67968.vpc-lattice-
svcs.us-east-2.on.aws",
        "hostedZoneId": "Z09127221KTH2CFUOHIZH"
    },
    "id": "svc-0285b53b2eEXAMPLE",
    "lastUpdatedAt": "2023-05-05T21:35:29.339Z",
    "name": "my-lattice-service",
    "status": "ACTIVE"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetService。

get-target-group

以下程式碼範例顯示如何使用 get-target-group。

AWS CLI

取得目標群組的相關資訊

下列get-target-group範例取得指定目標群組的相關資訊，其目標類型為 INSTANCE。

```
aws vpc-lattice get-target-group \
    --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

輸出：

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
    "config": {
        "healthCheck": {
            "enabled": true,
            "healthCheckIntervalSeconds": 30,
            "healthCheckTimeoutSeconds": 5,
            "healthyThresholdCount": 5,
            "matcher": {
                "httpCode": "200"
            },
```

```
            "path": "/",
            "protocol": "HTTPS",
            "protocolVersion": "HTTP1",
            "unhealthyThresholdCount": 2
        },
        "port": 443,
        "protocol": "HTTPS",
        "protocolVersion": "HTTP1",
        "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
    },
    "createdAt": "2023-05-06T04:41:04.122Z",
    "id": "tg-0eaa4b9ab4EXAMPLE",
    "lastUpdatedAt": "2023-05-06T04:41:04.122Z",
    "name": "my-target-group",
    "serviceArns": [
        "arn:aws:vpc-lattice:us-east-2:123456789012:service/svc-0285b53b2eEXAMPLE"
    ],
    "status": "ACTIVE",
    "type": "INSTANCE"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的目標群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTargetGroup。

list-listeners

以下程式碼範例顯示如何使用 list-listeners。

AWS CLI

列出服務接聽程式

下列list-listeners範例列出指定服務的接聽程式。

```
aws vpc-lattice list-listeners \
    --service-identifier svc-0285b53b2eEXAMPLE
```

輸出：

```
{
    "items": [
```

```
        {
            "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE/listener/listener-0ccf55918cEXAMPLE",
            "createdAt": "2023-05-07T05:08:45.192Z",
            "id": "listener-0ccf55918cEXAMPLE",
            "lastUpdatedAt": "2023-05-07T05:08:45.192Z",
            "name": "http-80",
            "port": 80,
            "protocol": "HTTP"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的定義路由。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListListeners。

`list-resource-configurations`

以下程式碼範例顯示如何使用 `list-resource-configurations`。

AWS CLI

列出您的資源組態

下列`list-resource-configurations`範例列出您的資源組態。

```
aws vpc-lattice list-resource-configurations
```

輸出：

```
{
    "items": [
        {
            "amazonManaged": false,
            "arn": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceconfiguration/rcfg-07129f3acded87625",
            "createdAt": "2025-02-01T00:57:35.871000+00:00",
            "id": "rcfg-07129f3acded87625",
            "lastUpdatedAt": "2025-02-01T00:57:46.874000+00:00",
            "name": "my-resource-config",
            "resourceGatewayId": "rgw-0bba03f3d56060135",
```

```
            "status": "ACTIVE",
            "type": "SINGLE"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的資源組態。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourceConfigurations。

`list-resource-endpoint-associations`

以下程式碼範例顯示如何使用 `list-resource-endpoint-associations`。

AWS CLI

列出 VPC 端點關聯

下列`list-resource-endpoint-associations`範例列出與指定資源組態相關聯的 VPC 端點。

```
aws vpc-lattice list-resource-endpoint-associations \
    --resource-configuration-identifier rcfg-07129f3acded87625
```

輸出：

```
{
    "items": [
        {
            "arn": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceendpointassociation/rea-0956a7435baf89326",
            "createdAt": "2025-02-01T00:57:38.998000+00:00",
            "id": "rea-0956a7435baf89326",
            "resourceConfigurationArn": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceconfiguration/rcfg-07129f3acded87625",
            "resourceConfigurationId": "rcfg-07129f3acded87625",
            "vpcEndpointId": "vpce-019b90d6f16d4f958",
            "vpcEndpointOwner": "123456789012"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的管理 VPC Lattice 資源組態的關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourceEndpointAssociations。

`list-resource-gateways`

以下程式碼範例顯示如何使用 `list-resource-gateways`。

AWS CLI

列出您的資源閘道

下列`list-resource-gateways`範例列出您的資源閘道。

```
aws vpc-lattice list-resource-gateways
```

輸出：

```
{
    "items": [
        {
            "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourcegateway/
rgw-0bba03f3d56060135",
            "createdAt": "2025-02-01T00:57:33.241000+00:00",
            "id": "rgw-0bba03f3d56060135",
            "ipAddressType": "IPV4",
            "lastUpdatedAt": "2025-02-01T00:57:44.351000+00:00",
            "name": "my-resource-gateway",
            "seurityGroupIds": [
                "sg-087ffd596c5fe962c"
            ],
            "status": "ACTIVE",
            "subnetIds": [
                "subnet-08e8943905b63a683"
            ],
            "vpcIdentifier": "vpc-0bf4c2739bc05a694"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的 VPC Lattice 中的資源閘道。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourceGateways。

list-service-network-service-associations

以下程式碼範例顯示如何使用 list-service-network-service-associations。

AWS CLI

列出服務關聯

下列list-service-network-service-associations範例列出指定服務網路的服務關聯。--query 選項會將輸出範圍限制為服務關聯的 IDs。

```
aws vpc-lattice list-service-network-service-associations \
    --service-network-identifier sn-080ec7dc93EXAMPLE \
    --query items[*].id
```

輸出：

```
[
    "snsa-031fabb4d8EXAMPLE",
    "snsa-0e16955a8cEXAMPLE"
]
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的管理服務關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListServiceNetworkServiceAssociations。

list-service-network-vpc-associations

以下程式碼範例顯示如何使用 list-service-network-vpc-associations。

AWS CLI

列出 VPC 關聯

下列list-service-network-vpc-associations範例列出指定服務網路的 VPC 關聯。--query 選項會將輸出範圍限制為 VPC IDs。

```
aws vpc-lattice list-service-network-vpc-associations \
```

```
    --service-network-identifier sn-080ec7dc93EXAMPLE \
    --query items[*].id
```

輸出：

```
[
    "snva-0821fc8631EXAMPLE",
    "snva-0c5dcb60d6EXAMPLE"
]
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的管理 VPC 關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListServiceNetworkVpcAssociations。

list-service-network-vpc-endpoint-associations

以下程式碼範例顯示如何使用 list-service-network-vpc-endpoint-associations。

AWS CLI

列出 VPC 端點關聯

下列list-service-network-vpc-endpoint-associations範例列出與特定服務網路相關聯的 VPC 端點。

```
aws vpc-lattice list-service-network-vpc-endpoint-associations \
    --service-network-identifier sn-0808d1748faee0c1e
```

輸出：

```
{
    "items": [
        {
            "createdAt": "2025-02-01T01:21:36.667000+00:00",
            "serviceNetworkArn": "arn:aws:vpc-lattice:us-
east-1:123456789012:servicenetwork/sn-0808d1748faee0c1e",
            "state": "ACTIVE",
            "vpcEndpointId": "vpce-0cc199f605eaeace7",
            "vpcEndpointOwnerId": "123456789012"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的管理 VPC Lattice 服務網路的關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListServiceNetworkVpcEndpointAssociations。

`list-service-networks`

以下程式碼範例顯示如何使用 `list-service-networks`。

AWS CLI

列出您的服務網路

下列`list-service-networks`範例列出與呼叫帳戶擁有或共用的服務網路。`--query` 選項會將結果範圍限定為服務網路的 Amazon Resource Name (ARN)。

```
aws vpc-lattice list-service-networks \
    --query items[*].arn
```

輸出：

```
[
    "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
    "arn:aws:vpc-lattice:us-east-2:111122223333:servicenetwork/sn-0ec4d436cfEXAMPLE"
]
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的服務網路。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListServiceNetworks。

`list-services`

以下程式碼範例顯示如何使用 `list-services`。

AWS CLI

列出您的 服務

下列`list-services`範例列出與呼叫帳戶擁有或共用的 服務。`--query` 選項會將結果範圍限定為服務的 Amazon Resource Name (ARN)。

```
aws vpc-lattice list-services \
    --query items[*].arn
```

輸出：

```
[
    "arn:aws:vpc-lattice:us-east-2:123456789012:service/svc-0285b53b2eEXAMPLE",
    "arn:aws:vpc-lattice:us-east-2:111122223333:service/svc-0b8ac96550EXAMPLE"
]
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的服務。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListServices。

list-target-groups

以下程式碼範例顯示如何使用 list-target-groups。

AWS CLI

列出您的目標群組

下列list-target-groups範例列出目標類型為 的目標群組LAMBDA。

```
aws vpc-lattice list-target-groups \
    --target-group-type LAMBDA
```

輸出：

```
{
    "items": [
        {
            "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-045c1b7d9dEXAMPLE",
            "createdAt": "2023-05-06T05:22:16.637Z",
            "id": "tg-045c1b7d9dEXAMPLE",
            "lastUpdatedAt": "2023-05-06T05:22:16.637Z",
            "name": "my-target-group-lam",
            "serviceArns": [
                "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE"
            ],
```

```
            "status": "ACTIVE",
            "type": "LAMBDA"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的目標群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTargetGroups。

`list-targets`

以下程式碼範例顯示如何使用 `list-targets`。

AWS CLI

列出目標群組的目標

下列`list-targets`範例列出指定目標群組的目標。

```
aws vpc-lattice list-targets \
    --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

輸出：

```
{
    "items": [
        {
            "id": "i-07dd579bc5EXAMPLE",
            "port": 443,
            "status": "HEALTHY"
        },
        {
            "id": "i-047b3c9078EXAMPLE",
            "port": 443,
            "reasonCode": "HealthCheckFailed",
            "status": "UNHEALTHY"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的目標群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListTargets。

put-auth-policy

以下程式碼範例顯示如何使用 put-auth-policy。

AWS CLI

為服務建立身分驗證政策

下列put-auth-policy範例會授予來自使用指定 IAM 角色之任何已驗證委託人的請求存取權。資源是政策所連接之服務的 ARN。

```
aws vpc-lattice put-auth-policy \
    --resource-identifier svc-0285b53b2eEXAMPLE \
    --policy file://auth-policy.json
```

auth-policy.json 的內容：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/my-clients"
            },
            "Action": "vpc-lattice-svcs:Invoke",
            "Resource": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE"
        }
    ]
}
```

輸出：

```
{
    "policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:role/my-clients\"},
\"Action\":\"vpc-lattice-svcs:Invoke\",\"Resource\":\"arn:aws:vpc-lattice:us-
east-2:123456789012:service/svc-0285b53b2eEXAMPLE\"}]}",
```

```
    "state": "Active"
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的身分驗證政策。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutAuthPolicy。

register-targets

以下程式碼範例顯示如何使用 register-targets。

AWS CLI

註冊目標

下列register-targets範例會向指定的目標群組註冊指定的目標。

```
aws vpc-lattice register-targets \
    --targets id=i-047b3c9078EXAMPLE id=i-07dd579bc5EXAMPLE \
    --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

輸出：

```
{
    "successful": [
        {
            "id": "i-07dd579bc5EXAMPLE",
            "port": 443
        }
    ],
    "unsuccessful": [
        {
            "failureCode": "UnsupportedTarget",
            "failureMessage": "Instance targets must be in the same VPC as their
 target group",
            "id": "i-047b3c9078EXAMPLE",
            "port": 443
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon VPC Lattice 使用者指南》中的註冊目標。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterTargets。

AWS WAF Classic 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS WAF Classic。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`put-logging-configuration`

以下程式碼範例顯示如何使用 `put-logging-configuration`。

AWS CLI

使用指定的 Kinesis Firehose 串流 ARN 建立 Web ACL ARN 的記錄組態

下列`put-logging-configuration`範例顯示 WAF 與 CloudFront 的記錄組態。

```
aws waf put-logging-configuration \
    --logging-configuration ResourceArn=arn:aws:waf::123456789012:webacl/3bffd3ed-fa2e-445e-869f-a6a7cf153fd3,LogDestinationConfigs=arn:aws:firehose:us-east-1:123456789012:deliverystream/aws-waf-logs-firehose-stream,RedactedFields=[]
```

輸出：

```
{
    "LoggingConfiguration": {
        "ResourceArn": "arn:aws:waf::123456789012:webacl/3bffd3ed-fa2e-445e-869f-a6a7cf153fd3",
        "LogDestinationConfigs": [
            "arn:aws:firehose:us-east-1:123456789012:deliverystream/aws-waf-logs-firehose-stream"
```

```
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutLoggingConfiguration。

update-byte-match-set

以下程式碼範例顯示如何使用 update-byte-match-set。

AWS CLI

更新位元組比對集

下列update-byte-match-set命令會刪除 ByteMatchSet 中的 ByteMatchTuple 物件 （篩選條件）： ByteMatchSet

```
aws waf update-byte-match-set --byte-match-set-id a123fae4-
b567-8e90-1234-5ab67ac8ca90 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --
updates
 Action="DELETE",ByteMatchTuple={FieldToMatch={Type="HEADER",Data="referer"},TargetString="b
```

如需詳細資訊，請參閱 AWS WAF 開發人員指南中的使用字串比對條件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateByteMatchSet。

update-ip-set

以下程式碼範例顯示如何使用 update-ip-set。

AWS CLI

更新 IP 集

下列update-ip-set命令會使用 IPv4 地址更新 IPSet，並刪除 IPv6 地址：

```
aws waf update-ip-set --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90
 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates
 Action="INSERT",IPSetDescriptor={Type="IPV4",Value="12.34.56.78/16"},Action="DELETE",IPSetD
```

或者，您可以使用 JSON 檔案來指定輸入。例如：

```
aws waf update-ip-set --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --change-
token 12cs345-67cd-890b-1cd2-c3a4567d89f1  --updates file://change.json
```

JSON 檔案的內容為：

```
[
{
"Action": "INSERT",
"IPSetDescriptor":
{
"Type": "IPV4",
"Value": "12.34.56.78/16"
}
},
{
"Action": "DELETE",
"IPSetDescriptor":
{
"Type": "IPV6",
"Value": "1111:0000:0000:0000:0000:0000:0000:0111/128"
}
}
]
```

如需詳細資訊，請參閱 AWS WAF 開發人員指南中的使用 IP 比對條件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateIpSet。

update-rule

以下程式碼範例顯示如何使用 update-rule。

AWS CLI

更新規則

下列update-rule命令會刪除規則中的述詞物件：

```
aws waf update-rule --rule-id a123fae4-b567-8e90-1234-5ab67ac8ca90
 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates
 Action="DELETE",Predicate={Negated=false,Type="ByteMatch",DataId="MyByteMatchSetID"}
```

如需詳細資訊，請參閱 AWS WAF 開發人員指南中的使用規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRule。

update-size-constraint-set

以下程式碼範例顯示如何使用 update-size-constraint-set。

AWS CLI

更新大小限制集

下列update-size-constraint-set命令會刪除大小限制集中的 SizeConstraint 物件 （篩選條件）：

```
aws waf update-size-constraint-set --size-constraint-set-id a123fae4-
b567-8e90-1234-5ab67ac8ca90 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --
updates
 Action="DELETE",SizeConstraint={FieldToMatch={Type="QUERY_STRING"},TextTransformation="NONE
```

如需詳細資訊，請參閱 AWS WAF 開發人員指南中的使用大小限制條件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSizeConstraintSet。

update-sql-injection-match-set

以下程式碼範例顯示如何使用 update-sql-injection-match-set。

AWS CLI

更新 SQL Injection Match Set

下列update-sql-injection-match-set命令會刪除 SQL Injection 比對集中的 SqlInjectionMatchTuple 物件 （篩選條件）：

```
aws waf update-sql-injection-match-set --sql-injection-
match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --
change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates
 Action="DELETE",SqlInjectionMatchTuple={FieldToMatch={Type="QUERY_STRING"},TextTransformati
```

如需詳細資訊，請參閱 AWS WAF 開發人員指南中的使用 SQL Injection Match 條件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSqlInjectionMatchSet。

update-web-acl

以下程式碼範例顯示如何使用 update-web-acl。

AWS CLI

更新 Web ACL

下列update-web-acl命令會刪除 WebACL 中的ActivatedRule物件。

aws waf update-web-acl --web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 -updates Action="DELETE"，ActivatedRule='{Priority=1，RuleId="WAFRule-1-Example"，Action={Type="ALLOW"}，Type="REGULAR"}'

輸出：

```
{
    "ChangeToken": "12cs345-67cd-890b-1cd2-c3a4567d89f1"
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide 中的使用 Web ACLs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateWebAcl。

update-xss-match-set

以下程式碼範例顯示如何使用 update-xss-match-set。

AWS CLI

更新 XSSMatchSet

下列update-xss-match-set命令會刪除 XssMatchSet 中的 XssMatchTuple 物件 （篩選條件）： XssMatchSet

```
aws waf update-xss-match-set --xss-match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90
 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates
 Action="DELETE",XssMatchTuple={FieldToMatch={Type="QUERY_STRING"},TextTransformation="URL_D
```

如需詳細資訊，請參閱 AWS WAF 開發人員指南中的使用跨網站指令碼比對條件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateXssMatchSet。

AWS WAF Classic 區域性 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS WAF Classic 區域性。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-web-acl`

以下程式碼範例顯示如何使用 `associate-web-acl`。

AWS CLI

將 Web ACL 與資源建立關聯

下列`associate-web-acl`命令會將 web-acl-id 指定的 Web ACL 與 resource-arn 指定的資源建立關聯。資源 ARN 可以參考應用程式負載平衡器或 API Gateway：

```
aws waf-regional associate-web-acl \
    --web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --resource-arn 12cs345-67cd-890b-1cd2-c3a4567d89f1
```

如需詳細資訊，請參閱 AWS WAF 開發人員指南中的使用 Web ACLs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateWebAcl。

`put-logging-configuration`

以下程式碼範例顯示如何使用 `put-logging-configuration`。

AWS CLI

使用指定的 Kinesis Firehose 串流 ARN 建立 Web ACL ARN 的記錄組態

下列put-logging-configuration範例顯示區域 中 WAF 與 ALB/APIGateway 的記錄組態us-east-1。

```
aws waf-regional put-logging-configuration \
    --logging-configuration ResourceArn=arn:aws:waf-regional:us-east-1:123456789012:webacl/3bffd3ed-fa2e-445e-869f-a6a7cf153fd3,LogDestinationConfigs=arn:aws:firehose:us-east-1:123456789012:deliverystream/aws-waf-logs-firehose-stream,RedactedFields=[] \
    --region us-east-1
```

輸出：

```
{
    "LoggingConfiguration": {
        "ResourceArn": "arn:aws:waf-regional:us-east-1:123456789012:webacl/3bffd3ed-fa2e-445e-869f-a6a7cf153fd3",
        "LogDestinationConfigs": [
            "arn:aws:firehose:us-east-1:123456789012:deliverystream/aws-waf-logs-firehose-stream"
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutLoggingConfiguration。

update-byte-match-set

以下程式碼範例顯示如何使用 update-byte-match-set。

AWS CLI

更新位元組比對集

下列update-byte-match-set命令會刪除 中的ByteMatchTuple物件 （篩選條件）ByteMatchSet。由於該updates值具有內嵌雙引號，因此您必須以單引號括住該值。

```
aws waf-regional update-byte-match-set \
```

```
    --byte-match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
    --updates
 'Action="DELETE",ByteMatchTuple={FieldToMatch={Type="HEADER",Data="referer"},TargetString="
```

如需詳細資訊，請參閱 AWS WAF 開發人員指南中的使用字串比對條件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateByteMatchSet。

update-ip-set

以下程式碼範例顯示如何使用 update-ip-set。

AWS CLI

更新 IP 集

下列update-ip-set命令會使用 IPv4 地址更新 IPSet，並刪除 IPv6 地址。執行 get-change-token命令change-token以取得 的值。由於更新的 值包含內嵌雙引號，因此您必須以單引號括住該值。

```
aws waf update-ip-set \
    --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
    --updates
 'Action="INSERT",IPSetDescriptor={Type="IPV4",Value="12.34.56.78/16"},Action="DELETE",IPSet
```

或者，您可以使用 JSON 檔案來指定輸入。例如：

```
aws waf-regional update-ip-set \
    --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1  \
    --updates file://change.json
```

的內容 change.json

```
[
    {
        "Action": "INSERT",
        "IPSetDescriptor":
        {
```

```
            "Type": "IPV4",
            "Value": "12.34.56.78/16"
        }
    },
    {
        "Action": "DELETE",
        "IPSetDescriptor":
        {
            "Type": "IPV6",
            "Value": "1111:0000:0000:0000:0000:0000:0000:0111/128"
        }
    }
]
```

如需詳細資訊，請參閱 AWS WAF 開發人員指南中的使用 IP 比對條件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateIpSet。

update-rule

以下程式碼範例顯示如何使用 update-rule。

AWS CLI

更新規則

下列update-rule命令會刪除規則中的Predicate物件。由於updates值具有內嵌雙引號，因此您必須以單引號括住整個值。

```
aws waf-regional update-rule \
    --rule-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
    --updates
 'Action="DELETE",Predicate={Negated=false,Type="ByteMatch",DataId="MyByteMatchSetID"}'
```

如需詳細資訊，請參閱 AWS WAF 開發人員指南中的使用規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRule。

update-size-constraint-set

以下程式碼範例顯示如何使用 update-size-constraint-set。

AWS CLI

更新大小限制集

下列update-size-constraint-set命令會刪除大小限制集中的 SizeConstraint` 物件 （ 篩選條件 ）。由於 updates值包含內嵌雙引號，您必須以單引號括住整個值。

```
aws waf-regional update-size-constraint-set \
    --size-constraint-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
    --updates
 'Action="DELETE",SizeConstraint={FieldToMatch={Type="QUERY_STRING"},TextTransformation="NON
```

如需詳細資訊，請參閱 AWS WAF 開發人員指南中的使用大小限制條件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSizeConstraintSet。

update-sql-injection-match-set

以下程式碼範例顯示如何使用 update-sql-injection-match-set。

AWS CLI

更新 SQL Injection Match Set

下列update-sql-injection-match-set命令會刪除 SQL Injection 比對集中的SqlInjectionMatchTuple物件 （ 篩選條件 ）。由於 updates值包含內嵌雙引號，您必須以單引號括住整個值。 ：

aws waf-regional update-sql-injection-match-set --sql-injection-match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 ---change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates
'Action="DELETE"，SqlInjectionMatchTuple={FieldToMatch={Type="QUERY_STRING"}，TextTransform

如需詳細資訊，請參閱 AWS WAF 開發人員指南中的使用 SQL Injection Match 條件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSqlInjectionMatchSet。

update-web-acl

以下程式碼範例顯示如何使用 update-web-acl。

AWS CLI

更新 Web ACL

下列update-web-acl命令會刪除 WebACL 中的ActivatedRule物件。由於updates值包含內嵌雙引號，您必須以單引號括住整個值。

```
aws waf-regional update-web-acl \
    --web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
    --updates Action="DELETE",ActivatedRule='{Priority=1,RuleId="WAFRule-1-
Example",Action={Type="ALLOW"},Type="ALLOW"}'
```

如需詳細資訊，請參閱 AWS WAF 開發人員指南中的使用 Web ACLs。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateWebAcl。

update-xss-match-set

以下程式碼範例顯示如何使用 update-xss-match-set。

AWS CLI

更新 XSSMatchSet

下列update-xss-match-set命令會刪除 中的XssMatchTuple物件 （ 篩選條件 ）XssMatchSet。由於 updates值包含內嵌雙引號，因此您必須以單引號括住整個值。

```
aws waf-regional update-xss-match-set \
    --xss-match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
    --updates
 'Action="DELETE",XssMatchTuple={FieldToMatch={Type="QUERY_STRING"},TextTransformation="URL_
```

如需詳細資訊，請參閱 AWS WAF 開發人員指南中的使用跨網站指令碼比對條件。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateXssMatchSet。

AWS WAFV2 使用 的範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 來執行動作和實作常見案例 AWS WAFV2。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-web-acl`

以下程式碼範例顯示如何使用 `associate-web-acl`。

AWS CLI

將 Web ACL 與區域 AWS 資源建立關聯

下列`associate-web-acl`範例會將指定的 Web ACL 與 Application Load Balancer 建立關聯。

```
aws wafv2 associate-web-acl \
    --web-acl-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test-cli/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/
app/waf-cli-alb/1ea17125f8b25a2a \
    --region us-west-2
```

此命令不會產生輸出。

如需詳細資訊，請參閱AWS 《WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide》中的將 Web ACL 與 AWS 資源建立關聯或取消關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateWebAcl。

`check-capacity`

以下程式碼範例顯示如何使用 `check-capacity`。

AWS CLI

取得一組規則所使用的容量

以下內容會check-capacity擷取規則集的容量需求，其中包含速率型規則陳述式，以及包含巢狀規則的 AND 規則陳述式。

```
aws wafv2 check-capacity \
    --scope REGIONAL \
    --rules file://waf-rule-list.json \
    --region us-west-2
```

檔案的內容：//waf-rule-list.json：

```
[
    {
        "Name":"basic-rule",
        "Priority":0,
        "Statement":{
            "AndStatement":{
                "Statements":[
                    {
                        "ByteMatchStatement":{
                            "SearchString":"example.com",
                            "FieldToMatch":{
                                "SingleHeader":{
                                    "Name":"host"
                                }
                            },
                            "TextTransformations":[
                                {
                                    "Priority":0,
                                    "Type":"LOWERCASE"
                                }
                            ],
                            "PositionalConstraint":"EXACTLY"
                        }
                    },
                    {
                        "GeoMatchStatement":{
                            "CountryCodes":[
                                "US",
                                "IN"
                            ]
                        }
                    }
                ]
```

```
            }
        },
        "Action":{
            "Allow":{

            }
        },
        "VisibilityConfig":{
            "SampledRequestsEnabled":true,
            "CloudWatchMetricsEnabled":true,
            "MetricName":"basic-rule"
        }
    },
    {
        "Name":"rate-rule",
        "Priority":1,
        "Statement":{
            "RateBasedStatement":{
                "Limit":1000,
                "AggregateKeyType":"IP"
            }
        },
        "Action":{
            "Block":{

            }
        },
        "VisibilityConfig":{
            "SampledRequestsEnabled":true,
            "CloudWatchMetricsEnabled":true,
            "MetricName":"rate-rule"
        }
    }
]
```

輸出：

```
{
    "Capacity":15
}
```

如需詳細資訊，請參閱 AWS WAF、Firewall Manager 和 Shield Advanced Developer Guide 中的 WAF Web ACL 容量單位 (WCU)。 AWS AWS AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CheckCapacity。

create-ip-set

以下程式碼範例顯示如何使用 create-ip-set。

AWS CLI

建立用於 Web ACLs IP 集

下列create-ip-set命令會建立具有單一地址範圍規格的 IP 集。

```
aws wafv2 create-ip-set \
    --name testip \
    --scope REGIONAL \
    --ip-address-version IPV4 \
    --addresses 198.51.100.0/16
```

輸出：

```
{
    "Summary":{
        "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/ipset/testip/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Description":"",
        "Name":"testip",
        "LockToken":"447e55ac-0000-0000-0000-86b67c17f8b5",
        "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager AWS 和 Shield Advanced Developer Guide 中的 IP 集和 Regex 模式集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateIpSet。

create-regex-pattern-set

以下程式碼範例顯示如何使用 create-regex-pattern-set。

AWS CLI

建立用於 Web ACLs regex 模式集

下列create-regex-pattern-set命令會建立指定兩個 regex 模式的 regex 模式集。

```
aws wafv2 create-regex-pattern-set \
    --name regexPatterSet01 \
    --scope REGIONAL \
    --description 'Test web-acl' \
    --regular-expression-list '[{"RegexString": "/[0-9]*/"},{"RegexString": "/[a-z]*/"}]'
```

輸出：

```
{
    "Summary":{
        "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/regexpatternset/regexPatterSet01/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Description":"Test web-acl",
        "Name":"regexPatterSet01",
        "LockToken":"0bc01e21-03c9-4b98-9433-6229cbf1ef1c",
        "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager AWS 和 Shield Advanced Developer Guide 中的 IP 集和 Regex 模式集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRegexPatternSet。

create-rule-group

以下程式碼範例顯示如何使用 create-rule-group。

AWS CLI

建立要在 Web ACLs 中使用的自訂規則群組

下列create-rule-group命令會建立自訂規則群組以供區域使用。群組的規則陳述式是以 JSON 格式的檔案提供。

```
aws wafv2 create-rule-group \
    --name "TestRuleGroup" \
    --scope REGIONAL \
    --capacity 250 \
    --rules file://waf-rule.json \
    --visibility-
config SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=TestRuleGroupMet
 \
    --region us-west-2
```

檔案的內容：//waf-rule.json：

```
[
    {
        "Name":"basic-rule",
        "Priority":0,
        "Statement":{
            "AndStatement":{
                "Statements":[
                    {
                        "ByteMatchStatement":{
                            "SearchString":"example.com",
                            "FieldToMatch":{
                                "SingleHeader":{
                                    "Name":"host"
                                }
                            },
                            "TextTransformations":[
                                {
                                    "Priority":0,
                                    "Type":"LOWERCASE"
                                }
                            ],
                            "PositionalConstraint":"EXACTLY"
                        }
                    },
                    {
                        "GeoMatchStatement":{
                            "CountryCodes":[
                                "US",
                                "IN"
                            ]
                        }
```

```
                    }
                ]
            }
        },
        "Action":{
            "Allow":{

            }
        },
        "VisibilityConfig":{
            "SampledRequestsEnabled":true,
            "CloudWatchMetricsEnabled":true,
            "MetricName":"basic-rule"
        }
    }
]
```

輸出：

```
{
    "Summary":{
        "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/rulegroup/
TestRuleGroup/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Description":"",
        "Name":"TestRuleGroup",
        "LockToken":"7b3bcec2-374e-4c5a-b2b9-563bf47249f0",
        "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide 中的管理您自己的規則群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateRuleGroup。

`create-web-acl`

以下程式碼範例顯示如何使用 `create-web-acl`。

AWS CLI

建立 Web ACL

下列create-web-acl命令會建立 Web ACL 以供區域使用。Web ACL 的規則陳述式是以 JSON 格式的檔案提供。

```
aws wafv2 create-web-acl \
    --name TestWebAcl \
    --scope REGIONAL \
    --default-action Allow={} \
    --visibility-
config SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=TestWebAclMetric
 \
    --rules file://waf-rule.json \
    --region us-west-2
```

檔案的內容：//waf-rule.json：

```
[
    {
        "Name":"basic-rule",
        "Priority":0,
        "Statement":{
            "AndStatement":{
                "Statements":[
                    {
                        "ByteMatchStatement":{
                            "SearchString":"example.com",
                            "FieldToMatch":{
                                "SingleHeader":{
                                    "Name":"host"
                                }
                            },
                            "TextTransformations":[
                                {
                                    "Priority":0,
                                    "Type":"LOWERCASE"
                                }
                            ],
                            "PositionalConstraint":"EXACTLY"
                        }
                    },
                    {
                        "GeoMatchStatement":{
                            "CountryCodes":[
                                "US",
```

```
                                "IN"
                            ]
                        }
                    }
                ]
            }
        },
        "Action":{
            "Allow":{

            }
        },
        "VisibilityConfig":{
            "SampledRequestsEnabled":true,
            "CloudWatchMetricsEnabled":true,
            "MetricName":"basic-rule"
        }
    }
]
```

輸出：

```
{
    "Summary":{
        "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/TestWebAcl/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Description":"",
        "Name":"TestWebAcl",
        "LockToken":"2294b3a1-eb60-4aa0-a86f-a3ae04329de9",
        "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager AWS 和 Shield Advanced Developer Guide 中的管理和使用 Web 存取控制清單 (Web ACL)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateWebAcl。

delete-ip-set

以下程式碼範例顯示如何使用 delete-ip-set。

AWS CLI

刪除 IP 集

以下內容會delete-ip-set刪除指定的 IP 集。此呼叫需要一個 ID，您可以從呼叫取得，list-ip-sets以及一個鎖定字符，您可以從呼叫取得，list-ip-sets以及 get-ip-set。

```
aws wafv2 delete-ip-set \
    --name test1 \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --lock-token 46851772-db6f-459d-9385-49428812e357
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager AWS 和 Shield Advanced Developer Guide 中的 IP 集和 Regex 模式集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteIpSet。

delete-logging-configuration

以下程式碼範例顯示如何使用 delete-logging-configuration。

AWS CLI

停用 Web ACL 的記錄

以下內容會從指定的 Web ACL delete-logging-configuration移除任何記錄組態。

```
aws wafv2 delete-logging-configuration \
    --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide 中的記錄 Web ACL 流量資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLoggingConfiguration。

delete-regex-pattern-set

以下程式碼範例顯示如何使用 delete-regex-pattern-set。

AWS CLI

刪除 regex 模式集

以下delete-regex-pattern-set更新指定 regex 模式集的設定。此呼叫需要一個 ID，您可以從呼叫取得，list-regex-pattern-sets以及一個鎖定字符，您可以從呼叫list-regex-pattern-sets或呼叫取得get-regex-pattern-set。

```
aws wafv2 delete-regex-pattern-set \
    --name regexPatterSet01 \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --lock-token 0bc01e21-03c9-4b98-9433-6229cbf1ef1c
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager AWS 和 Shield Advanced Developer Guide 中的 IP 集和 Regex 模式集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRegexPatternSet。

delete-rule-group

以下程式碼範例顯示如何使用 delete-rule-group。

AWS CLI

刪除自訂規則群組

以下內容會delete-rule-group刪除指定的自訂規則群組。此呼叫需要一個 ID，您可以從呼叫取得，list-rule-groups以及一個鎖定字符，您可以從呼叫list-rule-groups或呼叫取得get-rule-group。

```
aws wafv2 delete-rule-group \
    --name TestRuleGroup \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --lock-token 7b3bcec2-0000-0000-0000-563bf47249f0
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide 中的管理您自己的規則群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteRuleGroup。

delete-web-acl

以下程式碼範例顯示如何使用 delete-web-acl。

AWS CLI

刪除 Web ACL

以下內容會從您的帳戶delete-web-acl刪除指定的 Web ACL。只有當 Web ACL 與任何資源沒有關聯時，才能將其刪除。此呼叫需要一個 ID，您可以從呼叫取得，list-web-acls以及一個鎖定字符，您可以從呼叫list-web-acls或呼叫取得get-web-acl。

```
aws wafv2 delete-web-acl \
    --name test \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --lock-token ebab4ed2-155e-4c9a-9efb-e4c45665b1f5
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager AWS 和 Shield Advanced Developer Guide 中的管理和使用 Web 存取控制清單 (Web ACL)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteWebAcl。

describe-managed-rule-group

以下程式碼範例顯示如何使用 describe-managed-rule-group。

AWS CLI

擷取受管規則群組的描述

以下內容會describe-managed-rule-group擷取 AWS 受管規則群組的描述。

```
aws wafv2 describe-managed-rule-group \
    --vendor-name AWS \
    --name AWSManagedRulesCommonRuleSet \
    --scope REGIONAL
```

輸出：

```
{
    "Capacity": 700,
    "Rules": [
        {
            "Name": "NoUserAgent_HEADER",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "UserAgent_BadBots_HEADER",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "SizeRestrictions_QUERYSTRING",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "SizeRestrictions_Cookie_HEADER",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "SizeRestrictions_BODY",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "SizeRestrictions_URIPATH",
            "Action": {
```

```
                "Block": {}
            }
        },
        {
            "Name": "EC2MetaDataSSRF_BODY",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "EC2MetaDataSSRF_COOKIE",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "EC2MetaDataSSRF_URIPATH",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "EC2MetaDataSSRF_QUERYARGUMENTS",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "GenericLFI_QUERYARGUMENTS",
            "Action": {
                "Block": {}
            }
        },
        {
            }
            "Name": "GenericLFI_URIPATH",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "GenericLFI_BODY",
            "Action": {
                "Block": {}
```

```
            }
        },
        {
            "Name": "RestrictedExtensions_URIPATH",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "RestrictedExtensions_QUERYARGUMENTS",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "GenericRFI_QUERYARGUMENTS",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "GenericRFI_BODY",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "GenericRFI_URIPATH",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "CrossSiteScripting_COOKIE",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "CrossSiteScripting_QUERYARGUMENTS",
            "Action": {
                "Block": {}
            }
        },
```

```
        {
            "Name": "CrossSiteScripting_BODY",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "CrossSiteScripting_URIPATH",
            "Action": {
                "Block": {}
            }
        }
    ]
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide 中的受管規則群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeManagedRuleGroup。

`disassociate-web-acl`

以下程式碼範例顯示如何使用 `disassociate-web-acl`。

AWS CLI

取消 Web ACL 與區域 AWS 資源的關聯

下列`disassociate-web-acl`範例會從指定的 Application Load Balancer 移除任何現有的 Web ACL 關聯。

```
aws wafv2 disassociate-web-acl \
    --resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/waf-cli-alb/1ea17125f8b25a2a \
    --region us-west-2
```

此命令不會產生輸出。

如需詳細資訊，請參閱AWS 《WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide》中的將 Web ACL 與 AWS 資源建立關聯或取消關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateWebAcl。

get-ip-set

以下程式碼範例顯示如何使用 get-ip-set。

AWS CLI

擷取特定 IP 集

以下內容會get-ip-set擷取具有指定名稱、範圍和 ID 的 IP 集。您可以從命令 create-ip-set和 取得 IP 集的 IDlist-ip-sets。

```
aws wafv2 get-ip-set \
    --name testip \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "IPSet":{
        "Description":"",
        "Name":"testip",
        "IPAddressVersion":"IPV4",
        "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE1111",
        "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/ipset/testip/
a1b2c3d4-5678-90ab-cdef-EXAMPLE1111",
        "Addresses":[
            "192.0.2.0/16"
        ]
    },
    "LockToken":"447e55ac-2396-4c6d-b9f9-86b67c17f8b5"
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager AWS 和 Shield Advanced Developer Guide 中的 IP 集和 Regex 模式集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetIpSet。

get-logging-configuration

以下程式碼範例顯示如何使用 get-logging-configuration。

AWS CLI

擷取 Web ACL 的記錄組態

以下內容會get-logging-configuration擷取指定 Web ACL 的記錄組態。

```
aws wafv2 get-logging-configuration \
    --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 \
    --region us-west-2
```

輸出：

```
{
    "LoggingConfiguration":{
        "ResourceArn":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "RedactedFields":[
            {
                "Method":{

                }
            }
        ],
        "LogDestinationConfigs":[
            "arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-logs-
custom-transformation"
        ]
    }
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide 中的記錄 Web ACL 流量資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetLoggingConfiguration。

get-rate-based-statement-managed-keys

以下程式碼範例顯示如何使用 get-rate-based-statement-managed-keys。

AWS CLI

擷取速率型規則封鎖的 IP 地址清單

以下內容會get-rate-based-statement-managed-keys擷取目前由用於區域應用程式的速率型規則所封鎖的 IP 地址。

```
aws wafv2 get-rate-based-statement-managed-keys \
    --scope REGIONAL \
    --web-acl-name testwebacl2 \
    --web-acl-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --rule-name ratebasedtest
```

輸出：

```
{
    "ManagedKeysIPV4":{
        "IPAddressVersion":"IPV4",
        "Addresses":[
            "198.51.100.0/32"
        ]
    },
    "ManagedKeysIPV6":{
        "IPAddressVersion":"IPV6",
        "Addresses":[

        ]
    }
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide 中的以速率為基礎的規則陳述式。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRateBasedStatementManagedKeys。

get-regex-pattern-set

以下程式碼範例顯示如何使用 get-regex-pattern-set。

AWS CLI

擷取特定 regex 模式集

以下內容會get-regex-pattern-set擷取具有指定名稱、範圍、區域和 ID 的 regex 模式集。您可以從命令 create-regex-pattern-set和 取得 regex 模式集的 IDlist-regex-pattern-sets。

```
aws wafv2 get-regex-pattern-set \
    --name regexPatterSet01 \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --region us-west-2
```

輸出：

```
{
    "RegexPatternSet":{
        "Description":"Test web-acl",
        "RegularExpressionList":[
            {
                "RegexString":"/[0-9]*/"
            },
            {
                "RegexString":"/[a-z]*/"
            }
        ],
        "Name":"regexPatterSet01",
        "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/regexpatternset/
regexPatterSet01/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    },
    "LockToken":"c8abf33f-b6fc-46ae-846e-42f994d57b29"
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager AWS 和 Shield Advanced Developer Guide 中的 IP 集和 Regex 模式集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRegexPatternSet。

get-rule-group

以下程式碼範例顯示如何使用 get-rule-group。

AWS CLI

擷取特定自訂規則群組

以下內容會get-rule-group擷取具有指定名稱、範圍和 ID 的自訂規則群組。您可以從命令create-rule-group和 取得規則群組的 IDlist-rule-groups。

```
aws wafv2 get-rule-group \
    --name ff \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "RuleGroup":{
        "Capacity":1,
        "Description":"",
        "Rules":[
            {
                "Priority":0,
                "Action":{
                    "Block":{

                    }
                },
                "VisibilityConfig":{
                    "SampledRequestsEnabled":true,
                    "CloudWatchMetricsEnabled":true,
                    "MetricName":"jj"
                },
                "Name":"jj",
                "Statement":{
                    "SizeConstraintStatement":{
                        "ComparisonOperator":"LE",
                        "TextTransformations":[
                            {
                                "Priority":0,
                                "Type":"NONE"
                            }
                        ],
                        "FieldToMatch":{
                            "UriPath":{
```

```
                        }
                    },
                    "Size":7
                }
            }
        }
    ],
    "VisibilityConfig":{
        "SampledRequestsEnabled":true,
        "CloudWatchMetricsEnabled":true,
        "MetricName":"ff"
    },
    "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/rulegroup/ff/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Name":"ff"
},
"LockToken":"485458c9-1830-4234-af31-ec4d52ced1b3"
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide 中的管理您自己的規則群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRuleGroup。

get-sampled-requests

以下程式碼範例顯示如何使用 get-sampled-requests。

AWS CLI

擷取 Web ACL 的 Web 請求範例

以下內容會get-sampled-requests擷取指定 Web ACL、規則指標和時間範圍的範例 Web 請求。

```
aws wafv2 get-sampled-requests \
    --web-acl-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test-cli/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --rule-metric-name AWS-AWSManagedRulesSQLiRuleSet \
    --scope=REGIONAL \
    --time-window StartTime=2020-02-12T20:00Z,EndTime=2020-02-12T21:10Z \
```

```
    --max-items 100
```

輸出：

```
{
    "TimeWindow": {
    "EndTime": 1581541800.0,
    "StartTime": 1581537600.0
    },
    "SampledRequests": [
        {
            "Action": "BLOCK",
            "Timestamp": 1581541799.564,
            "RuleNameWithinRuleGroup": "AWS#AWSManagedRulesSQLiRuleSet#SQLi_BODY",
            "Request": {
                "Country": "US",
                "URI": "/",
                "Headers": [
                    {
                        "Name": "Host",
                        "Value": "alb-test-1EXAMPLE1.us-east-1.elb.amazonaws.com"
                    },
                    {
                        "Name": "Content-Length",
                        "Value": "7456"
                    },
                    {
                        "Name": "User-Agent",
                        "Value": "curl/7.53.1"
                    },
                    {
                        "Name": "Accept",
                        "Value": "/"
                    },
                    {
                        "Name": "Content-Type",
                        "Value": "application/x-www-form-urlencoded"
                    }
                ],
                "ClientIP": "198.51.100.08",
                "Method": "POST",
                "HTTPVersion": "HTTP/1.1"
            },
```

```
            "Weight": 1
        },
        {
            "Action": "BLOCK",
            "Timestamp": 1581541799.988,
            "RuleNameWithinRuleGroup": "AWS#AWSManagedRulesSQLiRuleSet#SQLi_BODY",
            "Request": {
                "Country": "US",
                "URI": "/",
                "Headers": [
                    {
                        "Name": "Host",
                        "Value": "alb-test-1EXAMPLE1.us-east-1.elb.amazonaws.com"
                    },
                    {
                        "Name": "Content-Length",
                        "Value": "7456"
                    },
                    {
                        "Name": "User-Agent",
                        "Value": "curl/7.53.1"
                    },
                    {
                        "Name": "Accept",
                        "Value": "/"
                    },
                    {
                        "Name": "Content-Type",
                        "Value": "application/x-www-form-urlencoded"
                    }
                ],
                "ClientIP": "198.51.100.08",
                "Method": "POST",
                "HTTPVersion": "HTTP/1.1"
            },
            "Weight": 3
        },
        {
            "Action": "BLOCK",
            "Timestamp": 1581541799.846,
            "RuleNameWithinRuleGroup": "AWS#AWSManagedRulesSQLiRuleSet#SQLi_BODY",
            "Request": {
                "Country": "US",
                "URI": "/",
```

```
                "Headers": [
                    {
                        "Name": "Host",
                        "Value": "alb-test-1EXAMPLE1.us-east-1.elb.amazonaws.com"
                    },
                    {
                        "Name": "Content-Length",
                        "Value": "7456"
                    },
                    {
                        "Name": "User-Agent",
                        "Value": "curl/7.53.1"
                    },
                    {
                        "Name": "Accept",
                        "Value": "/"
                    },
                    {
                        "Name": "Content-Type",
                        "Value": "application/x-www-form-urlencoded"
                    }
                ],
                "ClientIP": "198.51.100.08",
                "Method": "POST",
                "HTTPVersion": "HTTP/1.1"
            },
            "Weight": 1
        },
        {
            "Action": "BLOCK",
            "Timestamp": 1581541799.4,
            "RuleNameWithinRuleGroup": "AWS#AWSManagedRulesSQLiRuleSet#SQLi_BODY",
            "Request": {
                "Country": "US",
                "URI": "/",
                "Headers": [
                    {
                        "Name": "Host",
                        "Value": "alb-test-1EXAMPLE1.us-east-1.elb.amazonaws.com"
                    },
                    {
                        "Name": "Content-Length",
                        "Value": "7456"
                    },
```

```
                    {
                        "Name": "User-Agent",
                        "Value": "curl/7.53.1"
                    },
                    {
                        "Name": "Accept",
                        "Value": "/"
                    },
                    {
                        "Name": "Content-Type",
                        "Value": "application/x-www-form-urlencoded"
                    }
                ],
                "ClientIP": "198.51.100.08",
                "Method": "POST",
                "HTTPVersion": "HTTP/1.1"
            },
            "Weight": 1
        }
    ],
    "PopulationSize": 4
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager 和 Shield Advanced Developer Guide AWS 中的檢視 Web 請求範例。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSampledRequests。

get-web-acl-for-resource

以下程式碼範例顯示如何使用 get-web-acl-for-resource。

AWS CLI

擷取與 AWS 資源相關聯的 Web ACL

以下內容會get-web-acl-for-resource擷取與指定資源相關聯之 Web ACL 的 JSON。

```
aws wafv2 get-web-acl-for-resource \
    --resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/waf-cli-alb/1ea17125f8b25a2a
```

輸出：

```
{
    "WebACL":{
        "Capacity":3,
        "Description":"",
        "Rules":[
            {
                "Priority":1,
                "Action":{
                    "Block":{

                    }
                },
                "VisibilityConfig":{
                    "SampledRequestsEnabled":true,
                    "CloudWatchMetricsEnabled":true,
                    "MetricName":"testrule01"
                },
                "Name":"testrule01",
                "Statement":{
                    "AndStatement":{
                        "Statements":[
                            {
                                "ByteMatchStatement":{
                                    "PositionalConstraint":"EXACTLY",
                                    "TextTransformations":[
                                        {
                                            "Priority":0,
                                            "Type":"NONE"
                                        }
                                    ],
                                    "SearchString":"dGVzdHN0cmluZw==",
                                    "FieldToMatch":{
                                        "UriPath":{

                                        }
                                    }
                                }
                            },
                            {
                                "SizeConstraintStatement":{
                                    "ComparisonOperator":"EQ",
                                    "TextTransformations":[
                                        {
```

```
                                        "Priority":0,
                                        "Type":"NONE"
                                    }
                                ],
                                "FieldToMatch":{
                                    "QueryString":{

                                   }
                                },
                                "Size":0
                            }
                        }
                    ]
                }
            }
        }
    ],
    "VisibilityConfig":{
        "SampledRequestsEnabled":true,
        "CloudWatchMetricsEnabled":true,
        "MetricName":"test01"
    },
    "DefaultAction":{
        "Allow":{

        }
    },
    "Id":"9a1b2c3d4-5678-90ab-cdef-EXAMPLE11111  ",
    "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test01/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111  ",
    "Name":"test01"
  }
}
```

如需詳細資訊，請參閱AWS 《WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide》中的將 Web ACL 與 AWS 資源建立關聯或取消關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetWebAclForResource。

get-web-acl

以下程式碼範例顯示如何使用 get-web-acl。

AWS CLI

擷取 Web ACL

以下內容會get-web-acl擷取具有指定名稱、範圍和 ID 的 Web ACL。您可以從命令 create-web-acl和 取得 Web ACL 的 IDlist-web-acls。

```
aws wafv2 get-web-acl \
    --name test01 \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "WebACL":{
        "Capacity":3,
        "Description":"",
        "Rules":[
            {
                "Priority":1,
                "Action":{
                    "Block":{

                    }
                },
                "VisibilityConfig":{
                   "SampledRequestsEnabled":true,
                    "CloudWatchMetricsEnabled":true,
                    "MetricName":"testrule01"
                },
                "Name":"testrule01",
                "Statement":{
                    "AndStatement":{
                        "Statements":[
                            {
                                "ByteMatchStatement":{
                                    "PositionalConstraint":"EXACTLY",
                                    "TextTransformations":[
                                        {
                                            "Priority":0,
                                            "Type":"NONE"
                                        }
```

```
                                    ],
                                    "SearchString":"dGVzdHN0cmluZw==",
                                    "FieldToMatch":{
                                        "UriPath":{

                                        }
                                    }
                                }
                            },
                            {
                                "SizeConstraintStatement":{
                                    "ComparisonOperator":"EQ",
                                    "TextTransformations":[
                                        {
                                            "Priority":0,
                                            "Type":"NONE"
                                        }
                                    ],
                                    "FieldToMatch":{
                                        "QueryString":{

                                        }
                                    },
                                    "Size":0
                                }
                            }
                        ]
                    }
                }
            }
        ],
        "VisibilityConfig":{
            "SampledRequestsEnabled":true,
            "CloudWatchMetricsEnabled":true,
            "MetricName":"test01"
        },
        "DefaultAction":{
            "Allow":{

            }
        },
        "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test01/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```
        "Name":"test01"
    },
    "LockToken":"e3db7e2c-d58b-4ee6-8346-6aec5511c6fb"
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager AWS 和 Shield Advanced Developer Guide 中的管理和使用 Web 存取控制清單 (Web ACL)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetWebAcl。

list-available-managed-rule-groups

以下程式碼範例顯示如何使用 list-available-managed-rule-groups。

AWS CLI

擷取受管規則群組

以下內容會list-available-managed-rule-groups傳回目前可用於 Web ACLs 的所有受管規則群組清單。

```
aws wafv2 list-available-managed-rule-groups \
    --scope REGIONAL
```

輸出：

```
{
   "ManagedRuleGroups": [
       {
           "VendorName": "AWS",
           "Name": "AWSManagedRulesCommonRuleSet",
           "Description": "Contains rules that are generally applicable to web
applications. This provides protection against exploitation of a wide range of
vulnerabilities, including those described in OWASP publications and common Common
Vulnerabilities and Exposures (CVE)."
       },
       {
           "VendorName": "AWS",
           "Name": "AWSManagedRulesAdminProtectionRuleSet",
           "Description": "Contains rules that allow you to block external access
to exposed admin pages. This may be useful if you are running third-party software
or would like to reduce the risk of a malicious actor gaining administrative access
to your application."
```

```
        },
        {
            "VendorName": "AWS",
            "Name": "AWSManagedRulesKnownBadInputsRuleSet",
            "Description": "Contains rules that allow you to block request patterns
that are known to be invalid and are associated with exploitation or discovery of
vulnerabilities. This can help reduce the risk of a malicious actor discovering a
vulnerable application."
        },
        {
            "VendorName": "AWS",
            "Name": "AWSManagedRulesSQLiRuleSet",
            "Description": "Contains rules that allow you to block request patterns
associated with exploitation of SQL databases, like SQL injection attacks. This can
help prevent remote injection of unauthorized queries."
        },
        {
            "VendorName": "AWS",
            "Name": "AWSManagedRulesLinuxRuleSet",
            "Description": "Contains rules that block request patterns associated
with exploitation of vulnerabilities specific to Linux, including LFI attacks. This
can help prevent attacks that expose file contents or execute code for which the
attacker should not have had access."
        },
        {
            "VendorName": "AWS",
            "Name": "AWSManagedRulesUnixRuleSet",
            "Description": "Contains rules that block request patterns associated
with exploiting vulnerabilities specific to POSIX/POSIX-like OS, including LFI
attacks. This can help prevent attacks that expose file contents or execute code
for which access should not been allowed."
        },
        {
            "VendorName": "AWS",
            "Name": "AWSManagedRulesWindowsRuleSet",
            "Description": "Contains rules that block request patterns associated
with exploiting vulnerabilities specific to Windows, (e.g., PowerShell commands).
This can help prevent exploits that allow attacker to run unauthorized commands or
execute malicious code."
        },
        {
            "VendorName": "AWS",
            "Name": "AWSManagedRulesPHPRuleSet",
```

```
            "Description": "Contains rules that block request patterns associated
 with exploiting vulnerabilities specific to the use of the PHP, including injection
 of unsafe PHP functions. This can help prevent exploits that allow an attacker to
 remotely execute code or commands."
        },
        {
            "VendorName": "AWS",
            "Name": "AWSManagedRulesWordPressRuleSet",
            "Description": "The WordPress Applications group contains rules that
 block request patterns associated with the exploitation of vulnerabilities specific
 to WordPress sites."
        },
        {
            "VendorName": "AWS",
            "Name": "AWSManagedRulesAmazonIpReputationList",
            "Description": "This group contains rules that are based on Amazon
 threat intelligence. This is useful if you would like to block sources associated
 with bots or other threats."
        }
    ]
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide 中的受管規則群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAvailableManagedRuleGroups。

list-ip-sets

以下程式碼範例顯示如何使用 `list-ip-sets`。

AWS CLI

擷取 IP 集清單

以下內容會`list-ip-sets`擷取具有區域範圍之帳戶的所有 IP 集。

```
aws wafv2 list-ip-sets \
    --scope REGIONAL
```

輸出：

```
{
```

```
    "IPSets":[
        {
            "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/ipset/testip/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Description":"",
            "Name":"testip",
            "LockToken":"0674c84b-0304-47fe-8728-c6bff46af8fc",
            "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111  "
        }
    ],
    "NextMarker":"testip"
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager AWS 和 Shield Advanced Developer Guide 中的 IP 集和 Regex 模式集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListIpSets。

`list-logging-configurations`

以下程式碼範例顯示如何使用 `list-logging-configurations`。

AWS CLI

擷取區域的所有記錄組態清單

以下內容會`list-logging-configurations`擷取範圍限定為區域使用之 Web ACLs 的所有記錄組態`us-west-2`。

```
aws wafv2 list-logging-configurations \
    --scope REGIONAL \
    --region us-west-2
```

輸出：

```
{
    "LoggingConfigurations":[
        {
            "ResourceArn":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/
test-2/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "RedactedFields":[
                {
```

```
                    "QueryString":{

                    }
                }
            ],
            "LogDestinationConfigs":[
                "arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-
logs-test"
            ]
        },
        {
            "ResourceArn":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/
test/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "RedactedFields":[
                {
                    "Method":{

                    }
                }
            ],
            "LogDestinationConfigs":[
                "arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-
logs-custom-transformation"
            ]
        }
    ]
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide 中的記錄 Web ACL 流量資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListLoggingConfigurations。

`list-regex-pattern-sets`

以下程式碼範例顯示如何使用 `list-regex-pattern-sets`。

AWS CLI

擷取 regex 模式集的清單

以下內容會`list-regex-pattern-sets`擷取區域 中定義之帳戶的所有 regex 模式集`us-west-2`。

```
aws wafv2 list-regex-pattern-sets \
--scope REGIONAL \
--region us-west-2
```

輸出：

```
{
    "NextMarker":"regexPatterSet01",
    "RegexPatternSets":[
        {
            "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/regexpatternset/
regexPatterSet01/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Description":"Test web-acl",
            "Name":"regexPatterSet01",
            "LockToken":"f17743f7-0000-0000-0000-19a8b93bfb01",
            "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager AWS 和 Shield Advanced Developer Guide 中的 IP 集和 Regex 模式集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRegexPatternSets。

list-resources-for-web-acl

以下程式碼範例顯示如何使用 `list-resources-for-web-acl`。

AWS CLI

擷取與 Web ACL 相關聯的資源

以下內容會`list-resources-for-web-acl`擷取目前與區域 中指定的 Web ACL 相關聯的 API Gateway REST API 資源`us-west-2`。

```
aws wafv2 list-resources-for-web-acl \
    --web-acl-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/TestWebAcl/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --resource-type API_GATEWAY \
    --region us-west-2
```

輸出：

```
{
    "ResourceArns": [
        "arn:aws:apigateway:us-west-2::/restapis/EXAMPLE111/stages/testing"
    ]
}
```

如需詳細資訊，請參閱AWS 《WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide》中的將 Web ACL 與 AWS 資源建立關聯或取消關聯。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourcesForWebAcl。

list-rule-groups

以下程式碼範例顯示如何使用 list-rule-groups。

AWS CLI

擷取自訂規則群組的清單

以下內容會list-rule-groups擷取為指定範圍和區域位置的帳戶定義的所有自訂規則群組。

```
aws wafv2 list-rule-groups \
    --scope REGIONAL \
    --region us-west-2
```

輸出：

```
{
    "RuleGroups":[
        {
            "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/rulegroup/
TestRuleGroup/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Description":"",
            "Name":"TestRuleGroup",
            "LockToken":"1eb5ec48-0000-0000-0000-ee9b906c541e",
            "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        },
        {
            "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/rulegroup/test/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "Description":"",
```

```
            "Name":"test",
            "LockToken":"b0f4583e-998b-4880-9069-3fbe45738b43",
            "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
        }
    ],
    "NextMarker":"test"
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide 中的管理您自己的規則群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListRuleGroups。

`list-tags-for-resource`

以下程式碼範例顯示如何使用 `list-tags-for-resource`。

AWS CLI

擷取 AWS WAF 資源的所有標籤

以下內容會`list-tags-for-resource`擷取指定 Web ACL 的所有標籤索引鍵值對清單。

```
aws wafv2 list-tags-for-resource \
    --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/testwebacl2/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

輸出：

```
{
    "NextMarker":"",
    "TagInfoForResource":{
        "ResourceARN":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/
testwebacl2/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "TagList":[

        ]
    }
}
```

如需詳細資訊，請參閱 AWS WAF、Firewall Manager 和 Shield 進階開發人員指南中的 WAF 入門。 AWS AWS AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

`list-web-acls`

以下程式碼範例顯示如何使用 `list-web-acls`。

AWS CLI

擷取範圍ACLs

以下內容會`list-web-acls`擷取為指定範圍的帳戶定義的所有 Web ACLs。

```
aws wafv2 list-web-acls \
    --scope REGIONAL
```

輸出：

```
{
    "NextMarker":"Testt",
    "WebACLs":[
        {
            "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/Testt/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Description":"sssss",
            "Name":"Testt",
            "LockToken":"7f36cb30-74ef-4cff-8cd4-a77e1aba1746",
            "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        }
    ]
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager AWS 和 Shield Advanced Developer Guide 中的管理和使用 Web 存取控制清單 (Web ACL)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListWebAcls。

`put-logging-configuration`

以下程式碼範例顯示如何使用 `put-logging-configuration`。

AWS CLI

將記錄組態新增至 Web ACL

以下內容會將 Amazon Kinesis Data Firehose 記錄組態put-logging-configuration新增至aws-waf-logs-custom-transformation指定的 Web ACL，而不會從日誌中修訂任何欄位。

```
aws wafv2 put-logging-configuration \
    --logging-configuration ResourceArn=arn:aws:wafv2:us-
west-2:123456789012:regional/webacl/test-cli/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111,LogDestinationConfigs=arn:aws:firehose:us-
west-2:123456789012:deliverystream/aws-waf-logs-custom-transformation \
            --region us-west-2
```

輸出：

```
{
    "LoggingConfiguration":{
        "ResourceArn":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test-
cli/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "LogDestinationConfigs":[
            "arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-logs-
custom-transformation"
        ]
    }
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide 中的記錄 Web ACL 流量資訊。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutLoggingConfiguration。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤新增至 AWS WAF 資源

下列tag-resource範例會將索引鍵為 Name且值設為 的標籤AWSWAF新增至指定的 Web ACL。

```
aws wafv2 tag-resource \
    --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/
apiGatewayWebAcl/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --tags Key=Name,Value=AWSWAF
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS WAF、Firewall Manager 和 Shield 進階開發人員指南中的 WAF 入門。 AWS AWS AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

從 AWS WAF 資源移除標籤

下列untag-resource範例KeyName會從指定的 Web ACL 移除具有 金鑰的標籤。

```
aws wafv2 untag-resource \
    --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/
apiGatewayWebAcl/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --tag-keys "KeyName"
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS WAF、Firewall Manager 和 Shield 進階開發人員指南中的 WAF 入門。 AWS AWS AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-ip-set

以下程式碼範例顯示如何使用 update-ip-set。

AWS CLI

修改現有 IP 集的設定

以下內容會update-ip-set更新指定 IP 集的設定。此呼叫需要一個 ID，您可以從呼叫取得，list-ip-sets以及您可以從呼叫取得的鎖定字符，list-ip-sets以及 get-ip-set。此呼叫也會傳回鎖定字符，供您用於後續更新。

```
aws wafv2 update-ip-set \
    --name testip \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --addresses 198.51.100.0/16 \
    --lock-token 447e55ac-2396-4c6d-b9f9-86b67c17f8b5
```

輸出：

```
{
    "NextLockToken": "0674c84b-0304-47fe-8728-c6bff46af8fc"
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager AWS 和 Shield Advanced Developer Guide 中的 IP 集和 Regex 模式集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateIpSet。

update-regex-pattern-set

以下程式碼範例顯示如何使用 update-regex-pattern-set。

AWS CLI

修改現有 regex 模式集的設定

以下update-regex-pattern-set更新指定 regex 模式集的設定。此呼叫需要一個 ID，您可以從呼叫取得，list-regex-pattern-sets以及您可以從呼叫取得的鎖定字符，list-regex-pattern-sets以及 get-regex-pattern-set。此呼叫也會傳回鎖定字符，供您用於後續更新。

```
aws wafv2 update-regex-pattern-set \
    --name ExampleRegex \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --regular-expression-list RegexString="^.+$"  \
    --lock-token ed207e9c-82e9-4a77-aadd-81e6173ab7eb
```

輸出：

```
{
    "NextLockToken": "12ebc73e-fa68-417d-a9b8-2bdd761a4fa5"
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager AWS 和 Shield Advanced Developer Guide 中的 IP 集和 Regex 模式集。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRegexPatternSet。

update-rule-group

以下程式碼範例顯示如何使用 update-rule-group。

AWS CLI

更新自訂規則群組

下列update-rule-group變更現有自訂規則群組的可見性組態。此呼叫需要一個 ID，您可以從呼叫取得，list-rule-groups以及您可以從呼叫取得的鎖定字符，list-rule-groups以及get-rule-group。此呼叫也會傳回鎖定字符，供您用於後續更新。

```
aws wafv2 update-rule-group \
    --name TestRuleGroup \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --lock-token 7b3bcec2-0000-0000-0000-563bf47249f0 \
    --visibility-
config SampledRequestsEnabled=false,CloudWatchMetricsEnabled=false,MetricName=TestMetricsFor
 \
    --region us-west-2
```

輸出：

```
{
    "NextLockToken": "1eb5ec48-0000-0000-0000-ee9b906c541e"
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager 和 AWS Shield Advanced Developer Guide 中的管理您自己的規則群組。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateRuleGroup。

update-web-acl

以下程式碼範例顯示如何使用 update-web-acl。

AWS CLI

更新 Web ACL

下列update-web-acl變更現有 Web ACL 的設定。此呼叫需要一個 ID，您可以從呼叫取得，list-web-acls以及一個鎖定字符和其他設定，您可以從呼叫取得get-web-acl。此呼叫也會傳回鎖定字符，供您用於後續更新。

```
aws wafv2 update-web-acl \
    --name TestWebAcl \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --lock-token 2294b3a1-0000-0000-0000-a3ae04329de9 \
    --default-action Block={} \
    --visibility-
config SampledRequestsEnabled=false,CloudWatchMetricsEnabled=false,MetricName=NewMetricTestW
 \
    --rules file://waf-rule.json \
    --region us-west-2
```

輸出：

```
{
    "NextLockToken": "714a0cfb-0000-0000-0000-2959c8b9a684"
}
```

如需詳細資訊，請參閱 AWS WAF、 AWS Firewall Manager AWS 和 Shield Advanced Developer Guide 中的管理和使用 Web 存取控制清單 (Web ACL)。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateWebAcl。

使用 的 WorkDocs 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 WorkDocs 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`abort-document-version-upload`

以下程式碼範例顯示如何使用 `abort-document-version-upload`。

AWS CLI

停止文件版本上傳

此範例會停止先前啟動的文件版本上傳。

命令：

```
aws workdocs abort-document-version-upload --document-
id feaba64d4efdf271c2521b60a2a44a8f057e84beaabbe22f012673132098 35f2 --version-
id 1536773972914-ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dcccd417da9313
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AbortDocumentVersionUpload。

`activate-user`

以下程式碼範例顯示如何使用 `activate-user`。

AWS CLI

啟用使用者

此範例會啟用非作用中的使用者。

命令：

```
aws workdocs activate-user --user-
id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

輸出：

```
{
  "User": {
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "Username": "exampleUser",
      "EmailAddress": "exampleUser@site.awsapps.com",
      "GivenName": "Example",
      "Surname": "User",
      "OrganizationId": "d-926726012c",
      "RootFolderId":
 "75f67c183aa1217409ac87576a45c03a5df5e6d8c51c35c01669970538e86cd0",
      "RecycleBinFolderId":
 "642b7dd3e60b14204534f3df7b1959e01b5d170f8c2707f410e40a8149120a57",
      "Status": "ACTIVE",
      "Type": "MINIMALUSER",
      "CreatedTimestamp": 1521226107.747,
      "ModifiedTimestamp": 1525297406.462,
      "Storage": {
          "StorageUtilizedInBytes": 0,
          "StorageRule": {
              "StorageAllocatedInBytes": 0,
              "StorageType": "QUOTA"
          }
      }
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ActivateUser。

add-resource-permissions

以下程式碼範例顯示如何使用 add-resource-permissions。

AWS CLI

新增資源的許可

此範例會將許可新增至指定主體的資源。

命令：

```
aws workdocs add-resource-permissions --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --
principals Id=anonymous,Type=ANONYMOUS,Role=VIEWER
```

輸出：

```
{
  "ShareResults": [
      {
          "PrincipalId": "anonymous",
          "Role": "VIEWER",
          "Status": "SUCCESS",
          "ShareId":
 "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
          "StatusMessage": ""
      }
  ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AddResourcePermissions。

create-comment

以下程式碼範例顯示如何使用 create-comment。

AWS CLI

新增註解

此範例會將新的註解新增至指定的文件版本。

命令：

```
aws workdocs create-comment --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
text "This is a comment."
```

輸出：

```
{
  "Comment": {
      "CommentId": "1534799058197-
c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5",
      "ThreadId": "1534799058197-
c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5",
      "Text": "This is a comment.",
      "Contributor": {
          "Id": "arn:aws:iam::123456789123:user/exampleUser",
          "Username": "exampleUser",
          "GivenName": "Example",
          "Surname": "User",
          "Status": "ACTIVE"
      },
      "CreatedTimestamp": 1534799058.197,
      "Status": "PUBLISHED",
      "Visibility": "PUBLIC"
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateComment。

create-custom-metadata

以下程式碼範例顯示如何使用 create-custom-metadata。

AWS CLI

建立自訂中繼資料

此範例會為指定的文件建立自訂中繼資料。

命令：

```
aws workdocs create-custom-metadata --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --custom-
metadata KeyName1=example,KeyName2=example2
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateCustomMetadata。

create-folder

以下程式碼範例顯示如何使用 create-folder。

AWS CLI

建立資料夾

此範例會建立資料夾。

命令：

```
aws workdocs create-folder --name documents --parent-folder-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

輸出：

```
{
  "Metadata": {
      "Id": "50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
      "Name": "documents",
      "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "ParentFolderId":
 "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
      "CreatedTimestamp": 1534450467.622,
      "ModifiedTimestamp": 1534450467.622,
      "ResourceState": "ACTIVE",
      "Signature": "",
      "Size": 0,
      "LatestVersionSize": 0
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateFolder。

create-labels

以下程式碼範例顯示如何使用 create-labels。

AWS CLI

建立標籤

此範例會為文件建立一系列標籤。

命令：

```
aws workdocs create-labels --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --
labels "documents" "examples" "my_documents"
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateLabels。

create-notification-subscription

以下程式碼範例顯示如何使用 create-notification-subscription。

AWS CLI

建立通知訂閱

下列create-notification-subscription範例會設定指定 Amazon WorkDocs 組織的通知訂閱。

```
aws workdocs create-notification-subscription \
    --organization-id d-123456789c \
    --protocol HTTPS \
    --subscription-type ALL \
    --notification-endpoint "https://example.com/example"
```

輸出：

```
{
```

```
    "Subscription": {
        "SubscriptionId": "123ab4c5-678d-901e-f23g-45h6789j0123",
        "EndPoint": "https://example.com/example",
        "Protocol": "HTTPS"
    }
}
```

如需詳細資訊，請參閱《Amazon WorkDocs 開發人員指南》中的訂閱通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateNotificationSubscription。

`create-user`

以下程式碼範例顯示如何使用 `create-user`。

AWS CLI

建立新使用者

此範例會在 Simple AD 或 Microsoft AD 目錄中建立新的使用者。

命令：

```
aws workdocs create-user --organization-id d-926726012c --username exampleUser2
 --email-address exampleUser2@site.awsapps.com --given-name example2Name --
surname example2Surname --password examplePa$$w0rd
```

輸出：

```
{
  "User": {
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "Username": "exampleUser2",
      "EmailAddress": "exampleUser2@site.awsapps.com",
      "GivenName": "example2Name",
      "Surname": "example2Surname",
      "OrganizationId": "d-926726012c",
      "RootFolderId":
 "35b886cb17198cbd547655e58b025dff0cf34aaed638be52009567e23dc67390",
      "RecycleBinFolderId":
 "9858c3e9ed4c2460dde9aadb4c69fde998070dd46e5e985bd08ec6169ea249ff",
      "Status": "ACTIVE",
      "Type": "MINIMALUSER",
```

```
        "CreatedTimestamp": 1535478836.584,
        "ModifiedTimestamp": 1535478836.584,
        "Storage": {
            "StorageUtilizedInBytes": 0,
            "StorageRule": {
                "StorageAllocatedInBytes": 0,
                "StorageType": "QUOTA"
            }
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateUser。

deactivate-user

以下程式碼範例顯示如何使用 deactivate-user。

AWS CLI

停用使用者

此範例會停用作用中的使用者。

命令：

```
aws workdocs deactivate-user --user-
id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeactivateUser。

delete-comment

以下程式碼範例顯示如何使用 delete-comment。

AWS CLI

從文件版本中刪除指定的註解

此範例會從指定的文件版本中刪除指定的註解。

命令：

```
aws workdocs delete-comment --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
comment-id 1534799058197-
c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteComment。

delete-custom-metadata

以下程式碼範例顯示如何使用 delete-custom-metadata。

AWS CLI

從資源刪除自訂中繼資料

此範例會從指定的資源刪除所有自訂中繼資料。

命令：

```
aws workdocs delete-custom-metadata --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --delete-all
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteCustomMetadata。

delete-document

以下程式碼範例顯示如何使用 delete-document。

AWS CLI

刪除文件

此範例會刪除指定的文件。

命令：

```
aws workdocs delete-document --document-
id b83ed5e5b167b65ef69de9d597627ff1a0d4f07a45e67f1fab7d26b54427de0a
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteDocument。

delete-folder-contents

以下程式碼範例顯示如何使用 delete-folder-contents。

AWS CLI

刪除資料夾的內容

此範例會刪除指定資料夾的內容。

命令：

```
aws workdocs delete-folder-contents --folder-
id 26fa8aa4ba2071447c194f7b150b07149dbdb9e1c8a301872dcd93a4735ce65d
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFolderContents。

delete-folder

以下程式碼範例顯示如何使用 delete-folder。

AWS CLI

刪除資料夾

此範例會刪除指定的資料夾。

命令：

```
aws workdocs delete-folder --folder-
id 26fa8aa4ba2071447c194f7b150b07149dbdb9e1c8a301872dcd93a4735ce65d
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteFolder。

delete-labels

以下程式碼範例顯示如何使用 delete-labels。

AWS CLI

刪除標籤

此範例會從文件中刪除指定的標籤。

命令：

```
aws workdocs delete-labels --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --
labels "documents" "examples"
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteLabels。

delete-notification-subscription

以下程式碼範例顯示如何使用 delete-notification-subscription。

AWS CLI

刪除通知訂閱

下列delete-notification-subscription範例會刪除指定的通知訂閱。

```
aws workdocs delete-notification-subscription \
    --subscription-id 123ab4c5-678d-901e-f23g-45h6789j0123 \
    --organization-id d-123456789c
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon WorkDocs 開發人員指南》中的訂閱通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteNotificationSubscription。

delete-user

以下程式碼範例顯示如何使用 delete-user。

AWS CLI

若要刪除使用者

此範例會刪除使用者。

命令：

```
aws workdocs delete-user --user-
id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteUser。

describe-activities

以下程式碼範例顯示如何使用 describe-activities。

AWS CLI

取得使用者活動的清單

此範例會傳回指定組織的最新使用者活動清單，並針對最近兩個活動設定限制。

命令：

```
aws workdocs describe-activities --organization-id d-926726012c --limit 2
```

輸出：

```
{
  "UserActivities": [
      {
          "Type": "DOCUMENT_VERSION_DOWNLOADED",
          "TimeStamp": 1534800122.17,
          "Initiator": {
              "Id": "arn:aws:iam::123456789123:user/exampleUser"
          },
          "ResourceMetadata": {
              "Type": "document",
              "Name": "updatedDoc",
              "Id":
 "15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3",
              "Owner": {
                  "Id":
 "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
                  "GivenName": "exampleName",
                  "Surname": "exampleSurname"
              }
          }
      },
      {
          "Type": "DOCUMENT_VERSION_VIEWED",
          "TimeStamp": 1534799079.207,
          "Initiator": {
              "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
              "GivenName": "exampleName",
              "Surname": "exampleSurname"
          },
          "ResourceMetadata": {
              "Type": "document",
```

```
            "Name": "updatedDoc",
            "Id":
 "15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3",
            "Owner": {
                "Id":
 "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
                "GivenName": "exampleName",
                "Surname": "exampleSurname"
            }
        }
    }
  ],
  "Marker":
 "DnF1ZXJ5VGhlbkZldGNoAgAAAAAAAAS7FmlTaU1OdlFTU1h1UU00VVFIbDlRWHcAAAAAAAAJTRY3bWh5eUgzaVF1ZX
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeActivities。

describe-comments

以下程式碼範例顯示如何使用 describe-comments。

AWS CLI

列出指定文件版本的所有註解

此範例會列出指定文件版本的所有註解。

命令：

```
aws workdocs describe-comments --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920
```

輸出：

```
{
  "Comments": [
      {
          "CommentId": "1534799058197-
c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5",
```

```
            "ThreadId": "1534799058197-
c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5",
            "Text": "This is a comment.",
            "Contributor": {
                "Username": "arn:aws:iam::123456789123:user/exampleUser",
                "Type": "USER"
            },
            "CreatedTimestamp": 1534799058.197,
            "Status": "PUBLISHED",
            "Visibility": "PUBLIC"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeComments。

describe-document-versions

以下程式碼範例顯示如何使用 describe-document-versions。

AWS CLI

擷取文件的版本

此範例會擷取指定文件的文件版本，包括初始化版本和來源文件的 URL。

命令：

```
aws workdocs describe-document-versions --document-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --fields SOURCE
```

輸出：

```
{
    "DocumentVersions": [
        {
            "Id":
 "1534452029587-15e129dfc187505c407588df255be83de2920d733859f1d2762411d22a83e3ef",
            "Name": "exampleDoc.docx",
            "ContentType": "application/vnd.openxmlformats-
officedocument.wordprocessingml.document",
            "Size": 13922,
            "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
```

```
            "Status": "ACTIVE",
            "CreatedTimestamp": 1534452029.587,
            "ModifiedTimestamp": 1534452029.849,
            "CreatorId":
 "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
            "Source": {
                "ORIGINAL": "https://gb-us-west-2-prod-doc-source.s3.us-
west-2.amazonaws.com/
d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65/1534452029587-15e129dfc1875
response-content-disposition=attachment%3B%20filename%2A
%3DUTF-8%27%27exampleDoc29.docx&X-Amz-Algorithm=AWS1-ABCD-EFG234&X-Amz-
Date=20180816T204149Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20180816%2Fus-west-2%2Fs3%2Faws1_request&X-Amz-
Signature=01Ab2c34d567e8f90123g456hi78j901k2345678l901234mno56pqr78EXAMPLE"
            }
        },
        {
            "Id": "1529005196082-
bb75fa19abc287699cb07147f75816dce43a53a10f28dc001bf61ef2fab01c59",
            "Name": "exampleDoc.pdf",
            "ContentType": "application/pdf",
            "Size": 425916,
            "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
            "Status": "ACTIVE",
            "CreatedTimestamp": 1529005196.082,
            "ModifiedTimestamp": 1529005196.796,
            "CreatorId":
 "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
            "Source": {
                "ORIGINAL": "https://gb-us-west-2-prod-doc-source.s3.us-
west-2.amazonaws.com/
d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65/1529005196082-
bb75fa19abc287699cb07147f75816dce43a53a10f28dc001bf61ef2fab01c59?
response-content-disposition=attachment%3B%20filename%2A
%3DUTF-8%27%27exampleDoc29.pdf&X-Amz-Algorithm=AWS1-ABCD-EFG234&X-Amz-
Date=20180816T204149Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20180816%2Fus-west-2%2Fs3%2Faws1_request&X-Amz-
Signature=01Ab2c34d567e8f90123g456hi78j901k2345678l901234mno56pqr78EXAMPLE"
            }
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeDocumentVersions。

describe-folder-contents

以下程式碼範例顯示如何使用 describe-folder-contents。

AWS CLI

描述資料夾的內容

此範例說明指定資料夾的所有作用中內容，包括其文件和子資料夾，依日期遞增排序。

命令：

```
aws workdocs describe-folder-contents --folder-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678 --sort DATE --
order ASCENDING --type ALL
```

輸出：

```
{
  "Folders": [
      {
          "Id": "50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
          "Name": "testing",
          "CreatorId":
 "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
          "ParentFolderId":
 "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
          "CreatedTimestamp": 1534450467.622,
          "ModifiedTimestamp": 1534451113.504,
          "ResourceState": "ACTIVE",
          "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
          "Size": 23019,
          "LatestVersionSize": 11537
      }
  ],
  "Documents": [
      {
          "Id": "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
          "CreatorId":
 "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
          "ParentFolderId":
 "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
          "CreatedTimestamp": 1529005196.082,
```

```
            "ModifiedTimestamp": 1534452483.01,
            "LatestVersionMetadata": {
                "Id":
 "1534452029587-15e129dfc187505c407588df255be83de2920d733859f1d2762411d22a83e3ef",
                "Name": "exampleDoc.docx",
                "ContentType": "application/vnd.openxmlformats-
officedocument.wordprocessingml.document",
                "Size": 13922,
                "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
                "Status": "ACTIVE",
                "CreatedTimestamp": 1534452029.587,
                "ModifiedTimestamp": 1534452029.587,
                "CreatorId":
 "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
            },
            "ResourceState": "ACTIVE"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeFolderContents。

describe-groups

以下程式碼範例顯示如何使用 describe-groups。

AWS CLI

擷取群組清單

下列describe-groups範例列出與指定 Amazon WorkDocs 組織相關聯的群組。

```
aws workdocs describe-groups \
    --search-query "e" \
    --organization-id d-123456789c
```

輸出：

```
{
    "Groups": [
        {
            "Id": "S-1-1-11-1122222222-2222233333-3333334444-4444&d-123456789c",
```

```
            "Name": "Example Group 1"
        },
        {
            "Id": "S-1-1-11-1122222222-2222233333-3333334444-5555&d-123456789c",
            "Name": "Example Group 2"
        }
    ]
}
```

如需詳細資訊，請參閱 Amazon WorkDocs 管理指南中的 Amazon WorkDocs 入門。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeGroups。

describe-notification-subscriptions

以下程式碼範例顯示如何使用 describe-notification-subscriptions。

AWS CLI

擷取通知訂閱清單

下列describe-notification-subscriptions範例會擷取指定 Amazon WorkDocs 組織的通知訂閱。

```
aws workdocs describe-notification-subscriptions \
    --organization-id d-123456789c
```

輸出：

```
{
    "Subscriptions": [
        {
            "SubscriptionId": "123ab4c5-678d-901e-f23g-45h6789j0123",
            "EndPoint": "https://example.com/example",
            "Protocol": "HTTPS"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon WorkDocs 開發人員指南》中的訂閱通知。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeNotificationSubscriptions。

describe-resource-permissions

以下程式碼範例顯示如何使用 describe-resource-permissions。

AWS CLI

取得資源的許可清單

此範例會傳回指定資源 （ 文件或資料夾 ） 的許可清單。

命令：

```
aws workdocs describe-resource-permissions --resource-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3
```

輸出：

```
{
  "Principals": [
      {
          "Id": "anonymous",
          "Type": "ANONYMOUS",
          "Roles": [
              {
                  "Role": "VIEWER",
                  "Type": "DIRECT"
              }
          ]
      },
      {
          "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
          "Type": "USER",
          "Roles": [
              {
                  "Role": "OWNER",
                  "Type": "DIRECT"
              }
          ]
      },
      {
          "Id": "d-926726012c",
          "Type": "ORGANIZATION",
          "Roles": [
```

```
                {
                    "Role": "VIEWER",
                    "Type": "INHERITED"
                }
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeResourcePermissions。

describe-users

以下程式碼範例顯示如何使用 describe-users。

AWS CLI

擷取指定使用者的詳細資訊

此範例會擷取指定組織中所有使用者的詳細資訊。

命令：

```
aws workdocs describe-users --organization-id d-926726012c
```

輸出：

```
{
  "Users": [
      {
          "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
          "Username": "example1User",
          "OrganizationId": "d-926726012c",
          "RootFolderId":
 "3c0e3f849dd20a9771d937b9bbcc97e18796150ae56c26d64a4fa0320a2dedc9",
          "RecycleBinFolderId":
 "c277f4c4d647be1f5147b3184ffa96e1e2bf708278b696cacba68ba13b91f4fe",
          "Status": "INACTIVE",
          "Type": "USER",
          "CreatedTimestamp": 1535478999.452,
          "ModifiedTimestamp": 1535478999.452
```

```
        },
        {
            "Id": "S-1-1-11-1111111111-2222222222-3333333333-4444&d-926726012c",
            "Username": "example2User",
            "EmailAddress": "example2User@site.awsapps.com",
            "GivenName": "example2Name",
            "Surname": "example2Surname",
            "OrganizationId": "d-926726012c",
            "RootFolderId":
 "35b886cb17198cbd547655e58b025dff0cf34aaed638be52009567e23dc67390",
            "RecycleBinFolderId":
 "9858c3e9ed4c2460dde9aadb4c69fde998070dd46e5e985bd08ec6169ea249ff",
            "Status": "ACTIVE",
            "Type": "MINIMALUSER",
            "CreatedTimestamp": 1535478836.584,
            "ModifiedTimestamp": 1535478836.584
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeUsers。

get-document-path

以下程式碼範例顯示如何使用 get-document-path。

AWS CLI

擷取文件的路徑資訊

此範例會擷取指定文件的路徑資訊（從根資料夾的階層），並包含父資料夾的名稱。

命令：

```
aws workdocs get-document-path --document-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --fields NAME
```

輸出：

```
{
    "Path": {
        "Components": [
```

```
            {
                "Id":
 "a43d29cbb8e7c4d25cfee8b803a504b0dc63e760b55ad0c611c6b87691eb6ff3",
                "Name": "/"
            },
            {
                "Id":
 "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
                "Name": "Top Level Folder"
            },
            {
                "Id":
 "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
                "Name": "exampleDoc.docx"
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDocumentPath。

get-document-version

以下程式碼範例顯示如何使用 get-document-version。

AWS CLI

擷取指定文件的版本中繼資料

此範例會擷取指定文件的版本中繼資料，包括來源 URL 和自訂中繼資料。

命令：

```
aws workdocs get-document-version --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
fields SOURCE --include-custom-metadata
```

輸出：

```
{
  "Metadata": {
```

```
      "Id":
 "1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920",
      "Name": "exampleDoc",
      "ContentType": "application/vnd.openxmlformats-
officedocument.wordprocessingml.document",
      "Size": 11537,
      "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
      "Status": "ACTIVE",
      "CreatedTimestamp": 1521672507.741,
      "ModifiedTimestamp": 1534451113.504,
      "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "Source": {
          "ORIGINAL": "https://gb-us-west-2-prod-doc-source.s3.us-
west-2.amazonaws.com/15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3/152167
response-content-disposition=attachment%3B%20filename%2A
%3DUTF-8%27%27exampleDoc&X-Amz-Algorithm=AWS1-ABCD-EFG234&X-Amz-
Date=20180820T212202Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20180820%2Fus-west-2%2Fs3%2Faws1_request&X-Amz-
Signature=01Ab2c34d567e8f90123g456hi78j901k2345678l901234mno56pqr78EXAMPLE"
      }
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetDocumentVersion。

get-document

以下程式碼範例顯示如何使用 get-document。

AWS CLI

擷取文件詳細資訊

此範例會擷取指定文件的詳細資訊。

命令：

```
aws workdocs get-document --document-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65
```

輸出：

```
{
```

```
  "Metadata": {
      "Id": "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
      "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "ParentFolderId":
 "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
      "CreatedTimestamp": 1529005196.082,
      "ModifiedTimestamp": 1534452483.01,
      "LatestVersionMetadata": {
          "Id":
 "1534452029587-15e129dfc187505c407588df255be83de2920d733859f1d2762411d22a83e3ef",
          "Name": "exampleDoc.docx",
          "ContentType": "application/vnd.openxmlformats-
officedocument.wordprocessingml.document",
          "Size": 13922,
          "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
          "Status": "ACTIVE",
          "CreatedTimestamp": 1534452029.587,
          "ModifiedTimestamp": 1534452029.587,
          "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
      },
      "ResourceState": "ACTIVE"
  }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetDocument。

get-folder-path

以下程式碼範例顯示如何使用 get-folder-path。

AWS CLI

擷取資料夾的路徑資訊

此範例會擷取指定資料夾的路徑資訊（從根資料夾的階層），並包含父資料夾的名稱。

命令：

```
aws workdocs get-folder-path --folder-
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08 --fields NAME
```

輸出：

```
{
  "Path": {
      "Components": [
          {
              "Id":
 "a43d29cbb8e7c4d25cfee8b803a504b0dc63e760b55ad0c611c6b87691eb6ff3",
              "Name": "/"
          },
          {
              "Id":
 "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
              "Name": "Top Level Folder"
          },
          {
              "Id":
 "50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
              "Name": "Sublevel Folder"
          }
      ]
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFolderPath。

get-folder

以下程式碼範例顯示如何使用 get-folder。

AWS CLI

擷取資料夾的中繼資料

此範例會擷取指定資料夾的中繼資料。

命令：

```
aws workdocs get-folder --folder-
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08
```

輸出：

```
{
```

```
  "Metadata": {
      "Id": "50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
      "Name": "exampleFolder",
      "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "ParentFolderId":
 "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
      "CreatedTimestamp": 1534450467.622,
      "ModifiedTimestamp": 1534451113.504,
      "ResourceState": "ACTIVE",
      "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
      "Size": 23019,
      "LatestVersionSize": 11537
  }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetFolder。

get-resources

以下程式碼範例顯示如何使用 get-resources。

AWS CLI

擷取共用資源

下列get-resources範例會擷取與指定 Amazon WorkDocs 使用者共用的資源。

```
aws workdocs get-resources \
    --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333" \
    --collection-type SHARED_WITH_ME
```

輸出：

```
{
"Folders": [],
"Documents": []
}
```

如需詳細資訊，請參閱《Amazon WorkDocs 使用者指南》中的共用檔案和資料夾。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetResources。

initiate-document-version-upload

以下程式碼範例顯示如何使用 initiate-document-version-upload。

AWS CLI

啟動文件版本上傳

下列initiate-document-upload範例會建立新的文件物件和版本物件。

```
aws workdocs initiate-document-version-upload \
    --name exampledocname \
    --parent-folder-
id eacd546d952531c633452ed67cac23161aa0d5df2e8061223a59e8f67e7b6189
```

輸出：

```
{
    "Metadata": {
        "Id": "feaba64d4efdf271c2521b60a2a44a8f057e84beaabbe22f01267313209835f2",
        "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
        "ParentFolderId":
 "eacd546d952531c633452ed67cac23161aa0d5df2e8061223a59e8f67e7b6189",
        "CreatedTimestamp": 1536773972.914,
        "ModifiedTimestamp": 1536773972.914,
        "LatestVersionMetadata": {
            "Id": "1536773972914-
ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dcccd417da9313",
            "Name": "exampledocname",
            "ContentType": "application/octet-stream",
            "Size": 0,
            "Status": "INITIALIZED",
            "CreatedTimestamp": 1536773972.914,
            "ModifiedTimestamp": 1536773972.914,
            "CreatorId": "arn:aws:iam::123456789123:user/EXAMPLE"
        },
        "ResourceState": "ACTIVE"
    },
    "UploadMetadata": {
        "UploadUrl": "https://gb-us-west-2-prod-doc-source.s3.us-
west-2.amazonaws.com/
feaba64d4efdf271c2521b60a2a44a8f057e84beaabbe22f01267313209835f2/1536773972914-
ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dcccd417da9313?X-Amz-
```

```
Algorithm=AWS1-ABCD-EFG234&X-Amz-Date=20180912T173932Z&X-Amz-SignedHeaders=content-
type%3Bhost%3Bx-amz-server-side-encryption&X-Amz-Expires=899&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20180912%2Fus-west-2%2Fs3%2Faws1_request&X-Amz-
Signature=01Ab2c34d567e8f90123g456hi78j901k2345678l901234mno56pqr78EXAMPLE",
        "SignedHeaders": {
            "Content-Type": "application/octet-stream",
            "x-amz-server-side-encryption": "ABC123"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 InitiateDocumentVersionUpload。

`remove-all-resource-permissions`

以下程式碼範例顯示如何使用 `remove-all-resource-permissions`。

AWS CLI

從指定的資源移除所有許可

此範例會從指定的資源移除所有許可。

命令：

```
aws workdocs remove-all-resource-permissions --resource-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveAllResourcePermissions。

`remove-resource-permission`

以下程式碼範例顯示如何使用 `remove-resource-permission`。

AWS CLI

從資源移除許可

此範例會從指定主體的資源中移除許可。

命令：

```
aws workdocs remove-resource-permission --resource-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678 --principal-
id anonymous
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RemoveResourcePermission。

update-document-version

以下程式碼範例顯示如何使用 update-document-version。

AWS CLI

將文件版本狀態變更為作用中

此範例會將文件版本的狀態變更為作用中。

命令：

```
aws workdocs update-document-version --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
version-status ACTIVE
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateDocumentVersion。

update-document

以下程式碼範例顯示如何使用 update-document。

AWS CLI

更新文件

此範例會更新文件的名稱和父資料夾。

命令：

```
aws workdocs update-document --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --
name updatedDoc --parent-folder-
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateDocument。

update-folder

以下程式碼範例顯示如何使用 update-folder。

AWS CLI

更新資料夾

此範例會更新資料夾的名稱和父資料夾。

命令：

```
aws workdocs update-folder --folder-
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08 --
name exampleFolder1 --parent-folder-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

輸出：

```
None
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateFolder。

update-user

以下程式碼範例顯示如何使用 update-user。

AWS CLI

更新使用者

此範例會更新指定使用者的時區。

命令：

```
aws workdocs update-user --user-
id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c" --time-zone-
id "America/Los_Angeles"
```

輸出：

```
{
  "User": {
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "Username": "exampleUser",
      "EmailAddress": "exampleUser@site.awsapps.com",
      "GivenName": "Example",
      "Surname": "User",
      "OrganizationId": "d-926726012c",
      "RootFolderId":
 "c5eceb5e1a2d1d460c9d1af8330ae117fc8d39bb1d3ed6acd0992d5ff192d986",
      "RecycleBinFolderId":
 "6ca20102926ad15f04b1d248d6d6e44f2449944eda5c758f9a1e9df6a6b7fa66",
      "Status": "ACTIVE",
      "Type": "USER",
      "TimeZoneId": "America/Los_Angeles",
      "Storage": {
          "StorageUtilizedInBytes": 0,
          "StorageRule": {
              "StorageAllocatedInBytes": 53687091200,
              "StorageType": "QUOTA"
          }
      }
  }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateUser。

使用 的 Amazon WorkMail 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon WorkMail 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`associate-delegate-to-resource`

以下程式碼範例顯示如何使用 `associate-delegate-to-resource`。

AWS CLI

將委派新增至資源

下列`associate-delegate-to-resource`命令會將委派新增至資源。

```
aws workmail associate-delegate-to-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-68bf2d3b1c0244aab7264c24b9217443 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateDelegateToResource。

`associate-member-to-group`

以下程式碼範例顯示如何使用 `associate-member-to-group`。

AWS CLI

將成員新增至群組

下列associate-member-to-group命令會將指定的成員新增至群組。

```
aws workmail associate-member-to-group \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --group-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --member-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 AssociateMemberToGroup。

create-alias

以下程式碼範例顯示如何使用 create-alias。

AWS CLI

建立別名

下列create-alias命令會為指定的實體（使用者或群組）建立別名。

```
aws workmail create-alias \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --alias exampleAlias@site.awsapps.com
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateAlias。

create-group

以下程式碼範例顯示如何使用 create-group。

AWS CLI

建立新群組

下列create-group命令會為指定的組織建立新的群組。

```
aws workmail create-group \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --name exampleGroup1
```

輸出：

```
{
    "GroupId": "S-1-1-11-1122222222-2222233333-3333334444-4444"
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateGroup。

create-resource

以下程式碼範例顯示如何使用 create-resource。

AWS CLI

建立新的資源

下列create-resource命令會為指定的組織建立新的資源（會議室）。

```
aws workmail create-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --name exampleRoom1 \
    --type ROOM
```

輸出：

```
{
    "ResourceId": "r-7afe0efbade843a58cdc10251fce992c"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateResource。

create-user

以下程式碼範例顯示如何使用 create-user。

AWS CLI

建立新使用者

下列create-user命令會建立新的使用者。

```
aws workmail create-user \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --name exampleName \
    --display-name exampleDisplayName \
    --password examplePa$$w0rd
```

輸出：

```
{
    "UserId": "S-1-1-11-1111111111-2222222222-3333333333-3333"
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateUser。

delete-access-control-rule

以下程式碼範例顯示如何使用 delete-access-control-rule。

AWS CLI

刪除存取控制規則

下列delete-access-control-rule範例會從指定的 Amazon WorkMail 組織刪除指定的存取控制規則。

```
aws workmail delete-access-control-rule \
    --organization-id m-n1pq2345678r901st2u3vx45x6789yza \
    --name "myRule"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon WorkMail 管理員指南》中的使用存取控制規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteAccessControlRule。

delete-alias

以下程式碼範例顯示如何使用 delete-alias。

AWS CLI

刪除別名

下列delete-alias命令會刪除指定實體（使用者或群組）的別名。

```
aws workmail delete-alias \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --alias exampleAlias@site.awsapps.com
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteAlias。

delete-group

以下程式碼範例顯示如何使用 delete-group。

AWS CLI

刪除現有群組

下列delete-group命令會從 Amazon WorkMail 刪除現有群組。

```
aws workmail delete-group \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteGroup。

delete-mailbox-permissions

以下程式碼範例顯示如何使用 delete-mailbox-permissions。

AWS CLI

刪除信箱許可

下列delete-mailbox-permissions命令會刪除先前授予使用者或群組的信箱許可。實體代表擁有信箱的使用者，承授者代表要為其刪除許可的使用者或群組。

```
aws workmail delete-mailbox-permissions \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --grantee-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteMailboxPermissions。

delete-resource

以下程式碼範例顯示如何使用 delete-resource。

AWS CLI

刪除現有資源

下列delete-resource命令會從 Amazon WorkMail 刪除現有資源。

```
aws workmail delete-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-7afe0efbade843a58cdc10251fce992c
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteResource。

delete-user

以下程式碼範例顯示如何使用 delete-user。

AWS CLI

若要刪除使用者

下列delete-user命令會從 Amazon WorkMail 和所有後續系統刪除指定的使用者。

```
aws workmail delete-user \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteUser。

deregister-from-work-mail

以下程式碼範例顯示如何使用 deregister-from-work-mail。

AWS CLI

停用現有的實體

下列deregister-from-work-mail命令會停用現有的實體（使用者、群組或資源），而無法使用 Amazon WorkMail。

```
aws workmail deregister-from-work-mail \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterFromWorkMail。

describe-group

以下程式碼範例顯示如何使用 describe-group。

AWS CLI

擷取群組的資訊

下列describe-group命令會擷取指定群組的相關資訊。

```
aws workmail describe-group \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

輸出：

```
{
    "GroupId": "S-1-1-11-1122222222-2222233333-3333334444-4444",
    "Name": "exampleGroup1",
    "State": "ENABLED"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeGroup。

describe-organization

以下程式碼範例顯示如何使用 describe-organization。

AWS CLI

擷取組織的資訊

下列describe-organization命令會擷取指定 Amazon WorkMail 組織的資訊。

```
aws workmail describe-organization \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

輸出：

```
{
    "OrganizationId": "m-d281d0a2fd824be5b6cd3d3ce909fd27",
    "Alias": "alias",
    "State": "Active",
    "DirectoryId": "d-926726012c",
    "DirectoryType": "VpcDirectory",
    "DefaultMailDomain": "site.awsapps.com",
    "CompletedDate": 1522693605.468,
    "ARN": "arn:aws:workmail:us-west-2:111122223333:organization/m-
n1pq2345678r901st2u3vx45x6789yza"
}
```

如需詳細資訊，請參閱《Amazon WorkMail 管理員指南》中的使用 Organizations。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeOrganization。

describe-resource

以下程式碼範例顯示如何使用 describe-resource。

AWS CLI

擷取資源的資訊

下列describe-resource命令會擷取指定資源的相關資訊。

```
aws workmail describe-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-7afe0efbade843a58cdc10251fce992c
```

輸出：

```
{
    "ResourceId": "r-7afe0efbade843a58cdc10251fce992c",
    "Name": "exampleRoom1",
    "Type": "ROOM",
    "BookingOptions": {
        "AutoAcceptRequests": true,
        "AutoDeclineRecurringRequests": false,
        "AutoDeclineConflictingRequests": true
    },
    "State": "ENABLED"
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeResource。

describe-user

以下程式碼範例顯示如何使用 describe-user。

AWS CLI

擷取使用者資訊

下列describe-user命令會擷取指定使用者的相關資訊。

```
aws workmail describe-user \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

輸出：

```
{
    "UserId": "S-1-1-11-1111111111-2222222222-3333333333-3333",
    "Name": "exampleUser1",
    "Email": "exampleUser1@site.awsapps.com",
    "DisplayName": "",
    "State": "ENABLED",
    "UserRole": "USER",
    "EnabledDate": 1532459261.827
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeUser。

disassociate-delegate-from-resource

以下程式碼範例顯示如何使用 disassociate-delegate-from-resource。

AWS CLI

從資源中移除成員

下列disassociate-delegate-from-resource命令會從資源中移除指定的成員。

```
ws workmail disassociate-delegate-from-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-68bf2d3b1c0244aab7264c24b9217443 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateDelegateFromResource。

disassociate-member-from-group

以下程式碼範例顯示如何使用 disassociate-member-from-group。

AWS CLI

從群組中移除成員

下列disassociate-member-from-group命令會從群組中移除指定的成員。

```
aws workmail disassociate-member-from-group \
```

```
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --group-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --member-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DisassociateMemberFromGroup。

get-access-control-effect

以下程式碼範例顯示如何使用 get-access-control-effect。

AWS CLI

取得存取控制規則的效果

下列get-access-control-effect範例會擷取指定 IP 地址、存取通訊協定動作和使用者 ID 之指定 Amazon WorkMail 組織的存取控制規則的效果。

```
aws workmail get-access-control-effect \
    --organization-id m-n1pq2345678r901st2u3vx45x6789yza \
    --ip-address "192.0.2.0" \
    --action "WindowsOutlook" \
    --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333"
```

輸出：

```
{
    "Effect": "DENY",
    "MatchedRules": [
        "myRule"
    ]
}
```

如需詳細資訊，請參閱《Amazon WorkMail 管理員指南》中的使用存取控制規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetAccessControlEffect。

get-mailbox-details

以下程式碼範例顯示如何使用 get-mailbox-details。

AWS CLI

取得使用者的信箱詳細資訊

下列get-mailbox-details命令會擷取指定使用者信箱的詳細資訊。

```
aws workmail get-mailbox-details \
    --organization-id m-n1pq2345678r901st2u3vx45x6789yza \
    --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

輸出：

```
{
    "MailboxQuota": 51200,
    "MailboxSize": 0.03890800476074219
}
```

如需詳細資訊，請參閱《Amazon WorkMail 管理員指南》中的管理使用者帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetMailboxDetails。

list-access-control-rules

以下程式碼範例顯示如何使用 list-access-control-rules。

AWS CLI

列出存取控制規則

下列list-access-control-rules範例列出指定 Amazon WorkMail 組織的存取控制規則。

```
aws workmail list-access-control-rules \
    --organization-id m-n1pq2345678r901st2u3vx45x6789yza
```

輸出：

```
{
    "Rules": [
        {
            "Name": "default",
            "Effect": "ALLOW",
            "Description": "Default WorkMail Rule",
```

```
            "DateCreated": 0.0,
            "DateModified": 0.0
        },
        {
            "Name": "myRule",
            "Effect": "DENY",
            "Description": "my rule",
            "UserIds": [
            "S-1-1-11-1111111111-2222222222-3333333333-3333"
            ],
            "DateCreated": 1581635628.0,
            "DateModified": 1581635628.0
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon WorkMail 管理員指南》中的使用存取控制規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAccessControlRules。

list-aliases

以下程式碼範例顯示如何使用 list-aliases。

AWS CLI

列出成員的別名

下列list-aliases命令會列出指定成員（使用者或群組）的別名。

```
aws workmail list-aliases \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

輸出：

```
{
    "Aliases": [
        "exampleAlias@site.awsapps.com",
        "exampleAlias1@site.awsapps.com"
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListAliases。

`list-group-members`

以下程式碼範例顯示如何使用 `list-group-members`。

AWS CLI

列出群組成員

下列`list-group-members`命令會列出指定群組的成員。

```
aws workmail list-group-members \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

輸出：

```
{
    "Members": [
        {
            "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",
            "Name": "exampleUser1",
            "Type": "USER",
            "State": "ENABLED",
            "EnabledDate": 1532459261.827
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListGroupMembers。

`list-groups`

以下程式碼範例顯示如何使用 `list-groups`。

AWS CLI

擷取群組清單

下列`list-groups`命令會擷取指定組織中群組的摘要。

```
aws workmail list-groups \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

輸出：

```
{
    "Groups": [
        {
            "Id": "S-1-1-11-1122222222-2222233333-3333334444-4444",
            "Name": "exampleGroup1",
            "State": "DISABLED"
        },
        {
            "Id": "S-4-4-44-1122222222-2222233333-3333334444-4444",
            "Name": "exampleGroup2",
            "State": "ENABLED"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListGroups。

list-mailbox-permissions

以下程式碼範例顯示如何使用 list-mailbox-permissions。

AWS CLI

擷取信箱許可

下列list-mailbox-permissions命令會擷取與指定實體信箱相關聯的信箱許可。

```
aws workmail list-mailbox-permissions \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

輸出：

```
{
    "Permissions": [
```

```
        {
            "GranteeId": "S-1-1-11-1122222222-2222233333-3333334444-4444",
            "GranteeType": "USER",
            "PermissionValues": [
                "FULL_ACCESS"
            ]
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListMailboxPermissions。

list-organizations

以下程式碼範例顯示如何使用 list-organizations。

AWS CLI

擷取組織清單

下列list-organizations命令會擷取客戶組織的摘要。

```
aws workmail list-organizations
```

輸出：

```
{
    "OrganizationSummaries": [
        {
            "OrganizationId": "m-d281d0a2fd824be5b6cd3d3ce909fd27",
            "Alias": "exampleAlias",
            "State": "Active"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListOrganizations。

list-resource-delegates

以下程式碼範例顯示如何使用 list-resource-delegates。

AWS CLI

列出資源的委派代表

下列list-resource-delegates命令會擷取與指定資源相關聯的委派代表。

```
aws workmail list-resource-delegates \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-68bf2d3b1c0244aab7264c24b9217443
```

輸出：

```
{
    "Delegates": [
        {
            "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",
            "Type": "USER"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResourceDelegates。

list-resources

以下程式碼範例顯示如何使用 list-resources。

AWS CLI

擷取資源清單

下列list-resources命令會擷取指定組織的 資源摘要。

```
aws workmail list-resources \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

輸出：

```
{
    "Resources": [
        {
```

```
            "Id": "r-7afe0efbade843a58cdc10251fce992c",
            "Name": "exampleRoom1",
            "Type": "ROOM",
            "State": "ENABLED"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ListResources。

list-tags-for-resource

以下程式碼範例顯示如何使用 list-tags-for-resource。

AWS CLI

列出資源的標籤

下列list-tags-for-resource範例列出指定 Amazon WorkMail 組織的標籤。

```
aws workmail list-tags-for-resource \
    --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-n1pq2345678r901st2u3vx45x6789yza
```

輸出：

```
{
    "Tags": [
        {
            "Key": "priority",
            "Value": "1"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon WorkMail 管理員指南》中的標記組織。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTagsForResource。

list-users

以下程式碼範例顯示如何使用 list-users。

AWS CLI

擷取使用者清單

下列list-users命令會擷取指定組織中使用者的摘要。

```
aws workmail list-users \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

輸出：

```
{
    "Users": [
        {
            "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",
            "Email": "exampleUser1@site.awsapps.com",
            "Name": "exampleUser1",
            "State": "ENABLED",
            "UserRole": "USER",
            "EnabledDate": 1532459261.827
        },
        {
            "Id": "S-1-1-11-1122222222-2222233333-3333334444-4444",
            "Name": "exampleGuestUser",
            "State": "DISABLED",
            "UserRole": "SYSTEM_USER"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListUsers。

put-access-control-rule

以下程式碼範例顯示如何使用 put-access-control-rule。

AWS CLI

放置新的存取控制規則

下列put-access-control-rule範例會拒絕指定使用者存取指定的 Amazon WorkMail 組織。

```
aws workmail put-access-control-rule \
```

```
    --name "myRule" \
    --effect "DENY" \
    --description "my rule" \
    --user-ids "S-1-1-11-1111111111-2222222222-3333333333-3333" \
    --organization-id m-n1pq2345678r901st2u3vx45x6789yza
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon WorkMail 管理員指南》中的使用存取控制規則。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutAccessControlRule。

put-mailbox-permissions

以下程式碼範例顯示如何使用 put-mailbox-permissions。

AWS CLI

設定信箱許可

下列put-mailbox-permissions命令會設定指定承授者（使用者或群組）的完整存取許可。實體代表信箱的擁有者。

```
aws workmail put-mailbox-permissions \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333 \
    --grantee-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --permission-values FULL_ACCESS
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutMailboxPermissions。

register-to-work-mail

以下程式碼範例顯示如何使用 register-to-work-mail。

AWS CLI

註冊現有或已停用的實體

下列register-to-work-mail命令可讓指定的現有實體（使用者、群組或資源）使用 Amazon WorkMail。

```
aws workmail register-to-work-mail \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --email exampleGroup1@site.awsapps.com
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterToWorkMail。

reset-password

以下程式碼範例顯示如何使用 reset-password。

AWS CLI

重設使用者的密碼

下列reset-password命令會重設指定使用者的密碼。

```
aws workmail reset-password \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --user-id S-1-1-11-1111111111-2222222222-3333333333-3333 \
    --password examplePa$$w0rd
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ResetPassword。

tag-resource

以下程式碼範例顯示如何使用 tag-resource。

AWS CLI

將標籤套用至資源

下列tag-resource範例會將索引鍵為 "priority" 且值為 "1" 的標籤套用至指定的 Amazon WorkMail 組織。

```
aws workmail tag-resource \
    --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-
n1pq2345678r901st2u3vx45x6789yza \
```

```
    --tags "Key=priority,Value=1"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon WorkMail 管理員指南》中的標記組織。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TagResource。

untag-resource

以下程式碼範例顯示如何使用 untag-resource。

AWS CLI

取消標記資源

下列untag-resource範例會從指定的 Amazon WorkMail 組織移除指定的標籤。

```
aws workmail untag-resource \
    --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-
n1pq2345678r901st2u3vx45x6789yza \
    --tag-keys "priority"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon WorkMail 管理員指南》中的標記組織。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UntagResource。

update-mailbox-quota

以下程式碼範例顯示如何使用 update-mailbox-quota。

AWS CLI

更新使用者的信箱配額

下列update-mailbox-quota命令會變更指定使用者的信箱配額。

```
aws workmail update-mailbox-quota \
    --organization-id m-n1pq2345678r901st2u3vx45x6789yza \
    --user-id S-1-1-11-1111111111-2222222222-3333333333-3333 \
    --mailbox-quota 40000
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon WorkMail 管理員指南》中的管理使用者帳戶。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateMailboxQuota。

update-primary-email-address

以下程式碼範例顯示如何使用 update-primary-email-address。

AWS CLI

更新主要電子郵件地址

下列update-primary-email-address命令會更新指定實體（使用者、群組或資源）的主要電子郵件地址。

```
aws workmail update-primary-email-address \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333 \
    --email exampleUser2@site.awsapps.com
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdatePrimaryEmailAddress。

update-resource

以下程式碼範例顯示如何使用 update-resource。

AWS CLI

更新資源

下列update-resource命令會更新指定資源的名稱。

```
aws workmail update-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-7afe0efbade843a58cdc10251fce992c \
    --name exampleRoom2
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateResource。

使用 的 Amazon WorkMail 訊息流程範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Amazon WorkMail 訊息流程來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`get-raw-message-content`

以下程式碼範例顯示如何使用 `get-raw-message-content`。

AWS CLI

取得電子郵件訊息的原始內容

下列`get-raw-message-content`範例會取得傳輸中電子郵件訊息的原始內容，並將其傳送至名為 的文字檔案`test`。

```
aws workmailmessageflow get-raw-message-content \
    --message-id a1b2cd34-ef5g-6h7j-kl8m-npq9012345rs \
    test
```

命令執行`test`後的檔案內容：

```
Subject: Hello World
From: =?UTF-8?Q?marymajor_marymajor?= <marymajor@example.com>
To: =?UTF-8?Q?mateojackson=40example=2Enet?= <mateojackson@example.net>
Date: Thu, 7 Nov 2019 19:22:46 +0000
Mime-Version: 1.0
Content-Type: multipart/alternative;
```

```
 boundary="=_EXAMPLE+"
References: <mail.1ab23c45.5de6.7f890g123hj45678@storage.wm.amazon.com>
X-Priority: 3 (Normal)
X-Mailer: Amazon WorkMail
Thread-Index: EXAMPLE
Thread-Topic: Hello World
Message-Id: <mail.1ab23c45.5de6.7f890g123hj45678@storage.wm.amazon.com>

This is a multi-part message in MIME format. Your mail reader does not
understand MIME message format.
--=_EXAMPLE+
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 7bit

hello world


--=_EXAMPLE+
Content-Type: text/html; charset=utf-8
Content-Transfer-Encoding: quoted-printable

<!DOCTYPE HTML><html>
<head>
<meta name=3D"Generator" content=3D"Amazon WorkMail v3.0-4510">
<meta http-equiv=3D"Content-Type" content=3D"text/html; charset=3Dutf-8">=

<title>testing</title>
</head>
<body>
<p style=3D"margin: 0px; font-family: Arial, Tahoma, Helvetica, sans-seri=
f; font-size: small;">hello world</p>
</body>
</html>
--=_EXAMPLE+--
```

如需詳細資訊，請參閱《Amazon WorkMail 管理員指南》中的使用 AWS Lambda 擷取訊息內容。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetRawMessageContent。

使用 的 WorkSpaces 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 WorkSpaces 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`create-tags`

以下程式碼範例顯示如何使用 `create-tags`。

AWS CLI

將標籤新增至 WorkSpace

下列`create-tags`範例會將指定的標籤新增至指定的 WorkSpace。

```
aws workspaces create-tags \
    --resource-id ws-dk1xzr417 \
    --tags Key=Department,Value=Finance
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的標記 WorkSpaces 資源。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateTags。

`create-workspaces`

以下程式碼範例顯示如何使用 `create-workspaces`。

AWS CLI

範例 1：建立 AlwaysOn WorkSpace

下列`create-workspaces`範例會使用指定的目錄和套件，為指定的使用者建立 AlwaysOn WorkSpace。

```
aws workspaces create-workspaces \
    --workspaces DirectoryId=d-926722edaf,UserName=Mateo,BundleId=wsb-0zsvgp8fc
```

輸出：

```
{
    "FailedRequests": [],
    "PendingRequests": [
        {
            "WorkspaceId": "ws-kcqms853t",
            "DirectoryId": "d-926722edaf",
            "UserName": "Mateo",
            "State": "PENDING",
            "BundleId": "wsb-0zsvgp8fc"
        }
    ]
}
```

範例 2：建立 AutoStop WorkSpace

下列`create-workspaces`範例會使用指定的目錄和套件，為指定的使用者建立 AutoStop WorkSpace。

```
aws workspaces create-workspaces \
    --
workspaces DirectoryId=d-926722edaf,UserName=Mary,BundleId=wsb-0zsvgp8fc,WorkspaceProperties
```

輸出：

```
{
    "FailedRequests": [],
    "PendingRequests": [
        {
            "WorkspaceId": "ws-dk1xzr417",
            "DirectoryId": "d-926722edaf",
            "UserName": "Mary",
            "State": "PENDING",
            "BundleId": "wsb-0zsvgp8fc"
        }
    ]
}
```

範例 3：建立使用者解耦的 WorkSpace

下列create-workspaces範例會建立使用者解耦的 WorkSpace，方法是將使用者名稱設定為 [UNDEFINED]，並指定 WorkSpace 名稱、目錄 ID 和套件 ID。

```
aws workspaces create-workspaces \
    --workspaces
 DirectoryId=d-926722edaf,UserName='"[UNDEFINED]"',WorkspaceName=MaryWorkspace1,BundleId=wsb
```

輸出：

```
{
    "FailedRequests": [],
    "PendingRequests": [
        {
            "WorkspaceId": "ws-abcd1234",
            "DirectoryId": "d-926722edaf",
            "UserName": "[UNDEFINED]",
            "State": "PENDING",
            "BundleId": "wsb-0zsvgp8fc",
            "WorkspaceName": "MaryWorkspace1"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的啟動虛擬桌面。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateWorkspaces。

delete-tags

以下程式碼範例顯示如何使用 delete-tags。

AWS CLI

從 WorkSpace 刪除標籤

下列delete-tags範例會從指定的 WorkSpace 刪除指定的標籤。

```
aws workspaces delete-tags \
    --resource-id ws-dk1xzr417 \
```

```
    --tag-keys Department
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的標記 WorkSpaces 資源。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteTags。

deregister-workspace-directory

以下程式碼範例顯示如何使用 deregister-workspace-directory。

AWS CLI

取消註冊目錄

下列deregister-workspace-directory範例會取消註冊指定的目錄。

```
aws workspaces deregister-workspace-directory \
    --directory-id d-926722edaf
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的向 WorkSpaces 註冊目錄。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeregisterWorkspaceDirectory。

describe-tags

以下程式碼範例顯示如何使用 describe-tags。

AWS CLI

描述 WorkSpace 的標籤

下列describe-tags範例說明指定 WorkSpace 的標籤。

```
aws workspaces describe-tags \
    --resource-id ws-dk1xzr417
```

輸出：

```
{
    "TagList": [
        {
            "Key": "Department",
            "Value": "Finance"
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的標記 WorkSpaces 資源。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeTags。

`describe-workspace-bundles`

以下程式碼範例顯示如何使用 `describe-workspace-bundles`。

AWS CLI

列出 Amazon 提供的套件

下列`describe-workspace-bundles`範例以資料表格式列出 Amazon 提供的套件名稱和 IDs，並依名稱排序。

```
aws workspaces describe-workspace-bundles \
    --owner AMAZON \
    --query "Bundles[*].[Name, BundleId]"
```

輸出：

```
[
    [
        "Standard with Amazon Linux 2",
        "wsb-clj85qzj1"
    ],
    [
        "Performance with Windows 10 (Server 2016 based)",
        "wsb-gm4d5tx2v"
    ],
```

```
    [
        "PowerPro with Windows 7",
        "wsb-1pzkp0bx4"
    ],
    [
        "Power with Amazon Linux 2",
        "wsb-2bs6k5lgn"
    ],
    [
        "Graphics with Windows 10 (Server 2019 based)",
        "wsb-03gyjnfyy"
    ],
    ...
]
```

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的 WorkSpaces 套件和映像。Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeWorkspaceBundles。

describe-workspace-directories

以下程式碼範例顯示如何使用 describe-workspace-directories。

AWS CLI

描述已註冊的目錄

下列describe-workspace-directories範例說明指定的已註冊目錄。

```
aws workspaces describe-workspace-directories \
    --directory-ids d-926722edaf
```

輸出：

```
{
    "Directories": [
        {
            "DirectoryId": "d-926722edaf",
            "Alias": "d-926722edaf",
            "DirectoryName": "example.com",
            "RegistrationCode": "WSpdx+9RJ8JT",
```

```
            "SubnetIds": [
                "subnet-9d19c4c6",
                "subnet-500d5819"
            ],
            "DnsIpAddresses": [
                "172.16.1.140",
                "172.16.0.30"
            ],
            "CustomerUserName": "Administrator",
            "IamRoleId": "arn:aws:iam::123456789012:role/workspaces_DefaultRole",
            "DirectoryType": "SIMPLE_AD",
            "WorkspaceSecurityGroupId": "sg-0d89e927e5645d7c5",
            "State": "REGISTERED",
            "WorkspaceCreationProperties": {
                "EnableInternetAccess": false,
                "UserEnabledAsLocalAdministrator": true,
                "EnableMaintenanceMode": true
            },
            "WorkspaceAccessProperties": {
                "DeviceTypeWindows": "ALLOW",
                "DeviceTypeOsx": "ALLOW",
                "DeviceTypeWeb": "DENY",
                "DeviceTypeIos": "ALLOW",
                "DeviceTypeAndroid": "ALLOW",
                "DeviceTypeChromeOs": "ALLOW",
                "DeviceTypeZeroClient": "ALLOW",
                "DeviceTypeLinux": "DENY"
            },
            "Tenancy": "SHARED",
            "SelfservicePermissions": {
                "RestartWorkspace": "ENABLED",
                "IncreaseVolumeSize": "DISABLED",
                "ChangeComputeType": "DISABLED",
                "SwitchRunningMode": "DISABLED",
                "RebuildWorkspace": "DISABLED"
            }
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的管理 WorkSpaces Personal 的目錄。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeWorkspaceDirectories。

describe-workspaces-connection-status

以下程式碼範例顯示如何使用 describe-workspaces-connection-status。

AWS CLI

描述 WorkSpace 的連線狀態

下列describe-workspaces-connection-status範例說明指定 WorkSpace 的連線狀態。

```
aws workspaces describe-workspaces-connection-status \
    --workspace-ids ws-dk1xzr417
```

輸出：

```
{
    "WorkspacesConnectionStatus": [
        {
            "WorkspaceId": "ws-dk1xzr417",
            "ConnectionState": "CONNECTED",
            "ConnectionStateCheckTimestamp": 1662526214.744
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的管理 WorkSpaces。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeWorkspacesConnectionStatus。

describe-workspaces

以下程式碼範例顯示如何使用 describe-workspaces。

AWS CLI

描述 WorkSpace

下列describe-workspaces範例說明指定的 WorkSpace。

```
aws workspaces describe-workspaces \
```

```
    --workspace-ids ws-dk1xzr417
```

輸出：

```
{
    "Workspaces": [
        {
            "WorkspaceId": "ws-dk1xzr417",
            "DirectoryId": "d-926722edaf",
            "UserName": "Mary",
            "IpAddress": "172.16.0.175",
            "State": "STOPPED",
            "BundleId": "wsb-0zsvgp8fc",
            "SubnetId": "subnet-500d5819",
            "ComputerName": "WSAMZN-RBSLTTD9",
            "WorkspaceProperties": {
                "RunningMode": "AUTO_STOP",
                "RunningModeAutoStopTimeoutInMinutes": 60,
                "RootVolumeSizeGib": 80,
                "UserVolumeSizeGib": 10,
                "ComputeTypeName": "VALUE"
            },
            "ModificationStates": []
        }
    ]
}
```

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的管理 WorkSpaces。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DescribeWorkspaces。

migrate-workspace

以下程式碼範例顯示如何使用 migrate-workspace。

AWS CLI

遷移 WorkSpace

下列migrate-workspace範例會將指定的 WorkSpace 遷移至指定的套件。

```
aws workspaces migrate-workspace \
```

```
    --source-workspace-id ws-dk1xzr417 \
    --bundle-id wsb-j4dky1gs4
```

輸出：

```
{
    "SourceWorkspaceId": "ws-dk1xzr417",
    "TargetWorkspaceId": "ws-x5h1lbkp5"
}
```

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的遷移 WorkSpace。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 MigrateWorkspace。

modify-workspace-creation-properties

以下程式碼範例顯示如何使用 modify-workspace-creation-properties。

AWS CLI

修改目錄的 WorkSpace 建立屬性

下列modify-workspace-creation-properties範例會啟用指定目錄的 EnableInternetAccess 屬性。這可針對為目錄建立的 WorkSpaces 自動指派公有 IP 地址。

```
aws workspaces modify-workspace-creation-properties \
    --resource-id d-926722edaf \
    --workspace-creation-properties EnableInternetAccess=true
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的更新 WorkSpaces 的目錄詳細資訊。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyWorkspaceCreationProperties。

modify-workspace-properties

以下程式碼範例顯示如何使用 modify-workspace-properties。

AWS CLI

修改 WorkSpace 的執行模式

下列modify-workspace-properties範例會將指定 WorkSpace 的執行模式設定為 AUTO_STOP。

```
aws workspaces modify-workspace-properties \
    --workspace-id ws-dk1xzr417 \
    --workspace-properties RunningMode=AUTO_STOP
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的修改 WorkSpace。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyWorkspaceProperties。

modify-workspace-state

以下程式碼範例顯示如何使用 modify-workspace-state。

AWS CLI

修改 WorkSpace 的狀態

下列modify-workspace-state範例會將指定 WorkSpace 的狀態設定為 ADMIN_MAINTENANCE。

```
aws workspaces modify-workspace-state \
    --workspace-id ws-dk1xzr417 \
    --workspace-state ADMIN_MAINTENANCE
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon WorkSpace s管理指南》中的 WorkSpace 維護。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 ModifyWorkspaceState。

reboot-workspaces

以下程式碼範例顯示如何使用 reboot-workspaces。

AWS CLI

重新啟動 WorkSpace

下列reboot-workspaces範例會重新啟動指定的 WorkSpace。

```
aws workspaces reboot-workspaces \
    --reboot-workspace-requests ws-dk1xzr417
```

輸出：

```
{
    "FailedRequests": []
}
```

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的重新啟動 WorkSpace。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RebootWorkspaces。

rebuild-workspaces

以下程式碼範例顯示如何使用 rebuild-workspaces。

AWS CLI

重建 WorkSpace

下列rebuild-workspaces範例會重建指定的 WorkSpace。

```
aws workspaces rebuild-workspaces \
    --rebuild-workspace-requests ws-dk1xzr417
```

輸出：

```
{
    "FailedRequests": []
}
```

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的重建 WorkSpace。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RebuildWorkspaces。

`register-workspace-directory`

以下程式碼範例顯示如何使用 `register-workspace-directory`。

AWS CLI

註冊目錄

下列`register-workspace-directory`範例會註冊指定目錄，以便與 Amazon WorkSpaces 搭配使用。

```
aws workspaces register-workspace-directory \
    --directory-id d-926722edaf
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的向 WorkSpaces Personal 註冊現有的 AWS Directory Service 目錄。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RegisterWorkspaceDirectory。

`restore-workspace`

以下程式碼範例顯示如何使用 `restore-workspace`。

AWS CLI

還原 WorkSpace

下列`restore-workspace`範例會還原指定的 WorkSpace。

```
aws workspaces restore-workspace \
    --workspace-id ws-dk1xzr417
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的還原 WorkSpace。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 RestoreWorkspace。

`start-workspaces`

以下程式碼範例顯示如何使用 `start-workspaces`。

AWS CLI

啟動 AutoStop WorkSpace

下列`start-workspaces`範例會啟動指定的 WorkSpace。WorkSpace 的執行模式必須為 `AutoStop`。

```
aws workspaces start-workspaces \
    --start-workspace-requests WorkspaceId=ws-dk1xzr417
```

輸出：

```
{
    "FailedRequests": []
}
```

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的停止和啟動 AutoStop WorkSpace。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StartWorkspaces。

`stop-workspaces`

以下程式碼範例顯示如何使用 `stop-workspaces`。

AWS CLI

停止 AutoStop WorkSpace

下列`stop-workspaces`範例會停止指定的 WorkSpace。WorkSpace 的執行模式必須為 `AutoStop`。

```
aws workspaces stop-workspaces \
```

```
    --stop-workspace-requests WorkspaceId=ws-dk1xzr417
```

輸出：

```
{
    "FailedRequests": []
}
```

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的停止和啟動 AutoStop WorkSpace。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 StopWorkspaces。

`terminate-workspaces`

以下程式碼範例顯示如何使用 `terminate-workspaces`。

AWS CLI

終止 WorkSpace

下列`terminate-workspaces`範例會終止指定的工作區。

```
aws workspaces terminate-workspaces \
    --terminate-workspace-requests ws-dk1xzr417
```

輸出：

```
{
    "FailedRequests": []
}
```

如需詳細資訊，請參閱《Amazon WorkSpaces 管理指南》中的刪除 WorkSpace。 Amazon WorkSpaces

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 TerminateWorkspaces。

使用 的 X-Ray 範例 AWS CLI

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 X-Ray 執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

batch-traces-get

以下程式碼範例顯示如何使用 batch-traces-get。

AWS CLI

取得追蹤清單

下列batch-get-traces範例會擷取 ID 指定的追蹤清單。完整追蹤包含每個區段的文件，該文件是從所有使用相同追蹤 ID 接收到的區段文件編譯而成。

```
aws xray batch-get-traces \
    --trace-ids 1-5d82881a-0a9126e92a73e971eed891b9
```

輸出：

```
{
    "Traces": [
        {
            "Id": "1-5d82881a-0a9126e92a73e971eed891b9",
            "Duration": 0.232,
            "Segments": [
                {
                    "Id": "54aff5735b12dd28",
                    "Document": "{\"id\":\"54aff5735b12dd28\",\"name\":
\"Scorekeep\",\"start_time\":1.568835610432E9,\"end_time\":1.568835610664E9,
\"http\":{\"request\":{\"url\":\"http://scorekeep-env-1.m4fg2pfzpv.us-
east-2.elasticbeanstalk.com/api/user\",\"method\":\"POST\",\"user_agent\":
\"curl/7.59.0\",\"client_ip\":\"52.95.4.28\",\"x_forwarded_for\":true},
\"response\":{\"status\":200}},\"aws\":{\"elastic_beanstalk\":{\"version_label
```

```
\":\"Sample Application-1\",\"deployment_id\":3,\"environment_name\":\"Scorekeep-
env-1\"},\"ec2\":{\"availability_zone\":\"us-east-2b\",\"instance_id\":
\"i-0e3cf4d2de0f3f37a\"},\"xray\":{\"sdk_version\":\"1.1.0\",\"sdk\":\"X-Ray for
 Java\"}},\"service\":{\"runtime\":\"OpenJDK 64-Bit Server VM\",\"runtime_version
\":\"1.8.0_222\"},\"trace_id\":\"1-5d82881a-0a9126e92a73e971eed891b9\",
\"origin\":\"AWS::ElasticBeanstalk::Environment\",\"subsegments\":[{\"id\":
\"2d6900034ccfe558\",\"name\":\"DynamoDB\",\"start_time\":1.568835610658E9,
\"end_time\":1.568835610664E9,\"http\":{\"response\":{\"status\":200,
\"content_length\":61}},\"aws\":{\"table_name\":\"scorekeep-user\",\"operation\":
\"UpdateItem\",\"request_id\":\"TPEIDNDUROMLPOV17U4A79555NVV4KQNSO5AEMVJF66Q9ASUAAJG
\",\"resource_names\":[\"scorekeep-user\"]},\"namespace\":\"aws\"}]}"
                },
                {
                    "Id": "0f278b6334c34e6b",
                    "Document": "{\"id\":\"0f278b6334c34e6b\",\"name\":
\"DynamoDB\",\"start_time\":1.568835610658E9,\"end_time\":1.568835610664E9,
\"parent_id\":\"2d6900034ccfe558\",\"inferred\":true,\"http\":{\"response
\":{\"status\":200,\"content_length\":61}},\"aws\":{\"table_name
\":\"scorekeep-user\",\"operation\":\"UpdateItem\",\"request_id\":
\"TPEIDNDUROMLPOV17U4A79555NVV4KQNSO5AEMVJF66Q9ASUAAJG\",\"resource_names\":
[\"scorekeep-user\"]},\"trace_id\":\"1-5d82881a-0a9126e92a73e971eed891b9\",\"origin
\":\"AWS::DynamoDB::Table\"}"
                }
            ]
        }
    ],
    "UnprocessedTraceIds": []
}
```

如需詳細資訊，請參閱《AWS X-Ray 開發人員指南》中的搭配 AWS CLI 使用 X-Ray API。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 BatchTracesGet。

create-group

以下程式碼範例顯示如何使用 create-group。

AWS CLI

建立群組

下列create-group範例會建立名為 的群組資源AdminGroup。群組會取得篩選條件表達式，將群組的條件定義為與導致錯誤或錯誤的特定服務相關的區段。

```
aws xray create-group \
   --group-name "AdminGroup" \
   --filter-expression "service(\"mydomain.com\") {fault OR error}"
```

輸出：

```
{
    "GroupName": "AdminGroup",
    "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/AdminGroup/123456789",
    "FilterExpression": "service(\"mydomain.com\") {fault OR error}"
}
```

如需詳細資訊，請參閱《AWS X-Ray 開發人員指南》中的使用 X-Ray API 設定取樣、群組和加密設定。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateGroup。

create-sampling-rule

以下程式碼範例顯示如何使用 create-sampling-rule。

AWS CLI

建立取樣規則

下列create-sampling-rule範例會建立規則來控制受檢測應用程式的取樣行為。規則由 JSON 檔案提供。建立規則需要大部分的取樣規則欄位。

```
aws xray create-sampling-rule \
    --cli-input-json file://9000-base-scorekeep.json
```

9000-base-scorekeep.json 的內容：

```
{
    "SamplingRule": {
        "RuleName": "base-scorekeep",
        "ResourceARN": "*",
        "Priority": 9000,
        "FixedRate": 0.1,
        "ReservoirSize": 5,
```

```
        "ServiceName": "Scorekeep",
        "ServiceType": "*",
        "Host": "*",
        "HTTPMethod": "*",
        "URLPath": "*",
        "Version": 1
    }
}
```

輸出：

```
{
    "SamplingRuleRecord": {
        "SamplingRule": {
            "RuleName": "base-scorekeep",
            "RuleARN": "arn:aws:xray:us-west-2:123456789012:sampling-rule/base-
scorekeep",
            "ResourceARN": "*",
            "Priority": 9000,
            "FixedRate": 0.1,
            "ReservoirSize": 5,
            "ServiceName": "Scorekeep",
            "ServiceType": "*",
            "Host": "*",
            "HTTPMethod": "*",
            "URLPath": "*",
            "Version": 1,
            "Attributes": {}
        },
        "CreatedAt": 1530574410.0,
        "ModifiedAt": 1530574410.0
    }
}
```

如需詳細資訊，請參閱《AWS X-Ray 開發人員指南》中的使用 X-Ray API 設定取樣、群組和加密設定。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 CreateSamplingRule。

delete-group

以下程式碼範例顯示如何使用 delete-group。

AWS CLI

刪除群組

下列delete-group範例會刪除指定的群組資源。

```
aws xray delete-group \
    --group-name "AdminGroup" \
    --group-arn "arn:aws:xray:us-east-2:123456789012:group/AdminGroup/123456789"
```

此命令不會產生輸出。

如需詳細資訊，請參閱《AWS X-Ray 開發人員指南》中的使用 X-Ray API 設定取樣、群組和加密設定。 AWS

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteGroup。

delete-sampling-rule

以下程式碼範例顯示如何使用 delete-sampling-rule。

AWS CLI

刪除取樣規則

下列delete-sampling-rule範例會刪除指定的取樣規則。您可以使用群組名稱或群組 ARN 來指定群組。

```
aws xray delete-sampling-rule \
    --rule-name polling-scorekeep
```

輸出：

```
{
    "SamplingRuleRecord": {
        "SamplingRule": {
            "RuleName": "polling-scorekeep",
            "RuleARN": "arn:aws:xray:us-west-2:123456789012:sampling-rule/polling-
scorekeep",
            "ResourceARN": "*",
            "Priority": 5000,
```

```
            "FixedRate": 0.003,
            "ReservoirSize": 0,
            "ServiceName": "Scorekeep",
            "ServiceType": "*",
            "Host": "*",
            "HTTPMethod": "GET",
            "URLPath": "/api/state/*",
            "Version": 1,
            "Attributes": {}
        },
        "CreatedAt": 1530574399.0,
        "ModifiedAt": 1530574399.0
    }
}
```

如需詳細資訊，請參閱《AWS X-Ray 開發人員指南》中的使用 X-Ray API 設定取樣、群組和加密設定。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 DeleteSamplingRule。

get-encryption-config

以下程式碼範例顯示如何使用 get-encryption-config。

AWS CLI

擷取加密組態

下列get-encryption-config範例會擷取 AWS X-Ray 資料的目前加密組態。

```
aws xray get-encryption-config
```

輸出：

```
{
    "EncryptionConfig": {
        "KeyId": "ae4aa6d49-a4d8-9df9-a475-4ff6d7898456",
        "Status": "ACTIVE",
        "Type": "NONE"
    }
}
```

如需詳細資訊，請參閱《AWS X-Ray 開發人員指南》中的使用 X-Ray API 設定取樣、群組和加密設定。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetEncryptionConfig。

get-group

以下程式碼範例顯示如何使用 get-group。

AWS CLI

擷取群組

下列get-group範例顯示指定群組資源的詳細資訊。詳細資訊包括群組名稱、群組 ARN，以及定義該群組條件的篩選條件表達式。ARN 也可以擷取群組。

```
aws xray get-group \
    --group-name "AdminGroup"
```

輸出：

```
{
    "Group": [
        {
            "GroupName": "AdminGroup",
            "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/
AdminGroup/123456789",
            "FilterExpression": "service(\"mydomain.com\") {fault OR error}"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS X-Ray 開發人員指南》中的使用 X-Ray API 設定取樣、群組和加密設定。 AWS

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 GetGroup。

get-groups

以下程式碼範例顯示如何使用 get-groups。

AWS CLI

擷取所有群組

下列範例顯示所有作用中群組的詳細資訊。

```
aws xray get-groups
```

輸出：

```
{
    "Groups": [
        {
            "GroupName": "AdminGroup",
            "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/
AdminGroup/123456789",
            "FilterExpression": "service(\"example.com\") {fault OR error}"
        },
        {
            "GroupName": "SDETGroup",
            "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/
SDETGroup/987654321",
            "FilterExpression": "responsetime > 2"
        }
    ]
}
```

如需詳細資訊，請參閱《AWS X-Ray 開發人員指南》中的使用 X-Ray API 設定取樣、群組和加密設定。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetGroups。

get-sampling-rules

以下程式碼範例顯示如何使用 get-sampling-rules。

AWS CLI

擷取所有取樣規則

下列get-sampling-rules範例顯示所有可用抽樣規則的詳細資訊：

```
aws xray get-sampling-rules
```

輸出：

```
{
    "SamplingRuleRecords": [
        {
            "SamplingRule": {
                "RuleName": "Default",
                "RuleARN": "arn:aws:xray:us-east-1::sampling-rule/Default",
                "ResourceARN": "*",
                "Priority": 10000,
                "FixedRate": 0.01,
                "ReservoirSize": 0,
                "ServiceName": "*",
                "ServiceType": "*",
                "Host": "*",
                "HTTPMethod": "*",
                "URLPath": "*",
                "Version": 1,
                "Attributes": {}
            },
            "CreatedAt": 0.0,
            "ModifiedAt": 1530558121.0
        },
        {
            "SamplingRule": {
                "RuleName": "base-scorekeep",
                "RuleARN": "arn:aws:xray:us-east-1::sampling-rule/base-scorekeep",
                "ResourceARN": "*",
                "Priority": 9000,
                "FixedRate": 0.1,
                "ReservoirSize": 2,
                "ServiceName": "Scorekeep",
                "ServiceType": "*",
                "Host": "*",
                "HTTPMethod": "*",
                "URLPath": "*",
                "Version": 1,
                "Attributes": {}
            },
            "CreatedAt": 1530573954.0,
            "ModifiedAt": 1530920505.0
```

```
        },
        {
            "SamplingRule": {
                "RuleName": "polling-scorekeep",
                "RuleARN": "arn:aws:xray:us-east-1::sampling-rule/polling-
scorekeep",
                "ResourceARN": "*",
                "Priority": 5000,
                "FixedRate": 0.003,
                "ReservoirSize": 0,
                "ServiceName": "Scorekeep",
                "ServiceType": "*",
                "Host": "*",
                "HTTPMethod": "GET",
                "URLPath": "/api/state/*",
                "Version": 1,
                "Attributes": {}
            },
            "CreatedAt": 1530918163.0,
            "ModifiedAt": 1530918163.0
        }
    ]
}
```

如需詳細資訊，請參閱《X-Ray 開發人員指南》中的搭配 X-Ray API 使用取樣規則。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSamplingRules。

`get-sampling-targets`

以下程式碼範例顯示如何使用 `get-sampling-targets`。

AWS CLI

請求取樣配額

下列`get-sampling-targets`範例會針對服務用來取樣請求的規則請求取樣配額。 AWS X-Ray 的回應包含配額，可用來取代從儲槽借用。

```
aws xray get-sampling-targets \
    --sampling-statistics-documents '[ { "RuleName": "base-scorekeep", "ClientID":
 "ABCDEF1234567890ABCDEF10", "Timestamp": "2018-07-07T00:20:06, "RequestCount": 110,
```

```
 "SampledCount": 20, "BorrowCount": 10 }, { "RuleName": "polling-scorekeep", 31,
 "BorrowCount": 0 } ]'
```

輸出：

```
{
    "SamplingTargetDocuments": [
        {
            "RuleName": "base-scorekeep",
            "FixedRate": 0.1,
            "ReservoirQuota": 2,
            "ReservoirQuotaTTL": 1530923107.0,
            "Interval": 10
        },
        {
            "RuleName": "polling-scorekeep",
            "FixedRate": 0.003,
            "ReservoirQuota": 0,
            "ReservoirQuotaTTL": 1530923107.0,
            "Interval": 10
        }
    ],
    "LastRuleModification": 1530920505.0,
    "UnprocessedStatistics": []
}
```

如需詳細資訊，請參閱《X-Ray 開發人員指南》中的搭配 X-Ray API 使用取樣規則。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetSamplingTargets。

get-service-graph

以下程式碼範例顯示如何使用 get-service-graph。

AWS CLI

取得服務圖表

下列範例顯示指定期間內的文件，描述處理傳入請求的服務，以及他們因此呼叫的下游服務：

```
aws xray get-service-graph \
    --start-time 1568835392.0
    --end-time 1568835446.0
```

輸出：


```
{
    "Services": [
        {
            "ReferenceId": 0,
            "Name": "Scorekeep",
            "Names": [
                "Scorekeep"
            ],
            "Root": true,
            "Type": "AWS::ElasticBeanstalk::Environment",
            "State": "active",
            "StartTime": 1568835392.0,
            "EndTime": 1568835446.0,
            "Edges": [
                {
                    "ReferenceId": 1,
                    "StartTime": 1568835392.0,
                    "EndTime": 1568835446.0,
                    "SummaryStatistics": {
                        "OkCount": 14,
                        "ErrorStatistics": {
                            "ThrottleCount": 0,
                            "OtherCount": 0,
                            "TotalCount": 0
                        },
                        "FaultStatistics": {
                            "OtherCount": 0,
                            "TotalCount": 0
                        },
                        "TotalCount": 14,
                        "TotalResponseTime": 0.13
                    },
                    "ResponseTimeHistogram": [
                        {
                            "Value": 0.008,
                            "Count": 1
                        },
                        {
                            "Value": 0.005,
                            "Count": 7
                        },
                        {
```

```
                            "Value": 0.009,
                            "Count": 1
                        },
                        {
                            "Value": 0.021,
                            "Count": 1
                        },
                        {
                            "Value": 0.038,
                            "Count": 1
                        },
                        {
                            "Value": 0.007,
                            "Count": 1
                        },
                        {
                            "Value": 0.006,
                            "Count": 2
                        }
                    ],
                    "Aliases": []
                },

                ... TRUNCATED FOR BREVITY ...

            ]
        }
    ],
    "StartTime": 1568835392.0,
    "EndTime": 1568835446.0,
    "ContainsOldGroupVersions": false
}
```

如需詳細資訊，請參閱《AWS X-Ray 開發人員指南》中的搭配 AWS CLI 使用 X-Ray API。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetServiceGraph。

`get-trace-summaries`

以下程式碼範例顯示如何使用 `get-trace-summaries`。

AWS CLI

取得追蹤摘要

下列get-trace-summaries範例會擷取指定時間範圍內可用追蹤的 IDs 和中繼資料。

```
aws xray get-trace-summaries \
    --start-time 1568835392.0 \
    --end-time 1568835446.0
```

輸出：

```
[
    "http://scorekeep-env-1.123456789.us-east-2.elasticbeanstalk.com/api/move/
VSAE93HF/GSSD2NTB/DP0PCC09",
    "http://scorekeep-env-1.123456789.us-east-2.elasticbeanstalk.com/api/move/
GCQ2B35P/FREELDFT/4LRE643M",
    "http://scorekeep-env-1.123456789.us-east-2.elasticbeanstalk.com/api/game/
VSAE93HF/GSSD2NTB/starttime/1568835513",
    "http://scorekeep-env-1.123456789.us-east-2.elasticbeanstalk.com/api/
move/4MQNA5NN/L99KK2RF/null"
]
```

如需詳細資訊，請參閱《AWS X-Ray 開發人員指南》中的搭配 AWS CLI 使用 X-Ray API。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 GetTraceSummaries。

put-encryption-config

以下程式碼範例顯示如何使用 put-encryption-config。

AWS CLI

更新加密組態

下列 put-encryption-config``example updates the encryption configuration for AWS X-Ray data to use the default AWS managed KMS key ``aws/xray。

```
aws xray put-encryption-config \
    --type KMS \
    --key-id alias/aws/xray
```

輸出：

```
{
    "EncryptionConfig": {
```

```
        "KeyId": "arn:aws:kms:us-west-2:123456789012:key/c234g4e8-39e9-4gb0-84e2-
b0ea215cbba5",
        "Status": "UPDATING",
        "Type": "KMS"
    }
}
```

如需詳細資訊，請參閱《AWS X-Ray 開發人員指南》中的使用 X-Ray API 設定取樣、群組和加密設定。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutEncryptionConfig。

put-trace-segments

以下程式碼範例顯示如何使用 put-trace-segments。

AWS CLI

上傳客群

下列put-trace-segments範例會將區段文件上傳至 AWS X-Ray。區段文件會做為 JSON 區段文件的清單使用。

```
aws xray put-trace-segments \
    --trace-segment-documents "{\"id\":\"20312a0e2b8809f4\",\"name
\":\"DynamoDB\",\"trace_id\":\"1-5832862d-a43aafded3334a971fe312db\",
\"start_time\":1.479706157195E9,\"end_time\":1.479706157202E9,\"parent_id\":
\"79736b962fe3239e\",\"http\":{\"response\":{\"content_length\":60,\"status
\":200}},\"inferred\":true,\"aws\":{\"consistent_read\":false,\"table_name
\":\"scorekeep-session-xray\",\"operation\":\"GetItem\",\"request_id\":
\"SCAU23OM6M8FO38UASGC7785ARVV4KQNSO5AEMVJF66Q9ASUAAJG\",\"resource_names\":
[\"scorekeep-session-xray\"]},\"origin\":\"AWS::DynamoDB::Table\"}"
```

輸出：

```
{
    "UnprocessedTraceSegments": []
}
```

如需詳細資訊，請參閱《AWS X-Ray 開發人員指南》中的將追蹤資料傳送至 AWS X-Ray。

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 PutTraceSegments。

update-group

以下程式碼範例顯示如何使用 update-group。

AWS CLI

更新群組

下列update-group範例會更新要接受追蹤到名為 群組的條件AdminGroup。您可以使用群組名稱或群組 ARN 來指定所需的群組。

```
aws xray update-group \
    --group-name "AdminGroup" \
    --group-arn "arn:aws:xray:us-west-2:123456789012:group/AdminGroup/123456789" \
    --filter-expression "service(\"mydomain.com\") {fault}"
```

輸出：

```
{
    "GroupName": "AdminGroup",
    "GroupARN": "arn:aws:xray:us-east-2:123456789012:group/AdminGroup/123456789",
    "FilterExpression": "service(\"mydomain.com\") {fault}"
}
```

如需詳細資訊，請參閱《AWS X-Ray 開發人員指南》中的使用 X-Ray API 設定取樣、群組和加密設定。 AWS

- 如需 API 詳細資訊，請參閱 AWS CLI Command Reference 中的 UpdateGroup。

update-sampling-rule

以下程式碼範例顯示如何使用 update-sampling-rule。

AWS CLI

更新取樣規則

下列update-sampling-rule範例會修改取樣規則的組態。規則會從 JSON 檔案取用。只需要更新的欄位。

```
aws xray update-sampling-rule \
```

```
    --cli-input-json file://1000-default.json
```

1000-default.json 的內容：

```
{
    "SamplingRuleUpdate": {
        "RuleName": "Default",
        "FixedRate": 0.01,
        "ReservoirSize": 0
    }
}
```

輸出：

```
{
    "SamplingRuleRecords": [
        {
            "SamplingRule": {
                "RuleName": "Default",
                "RuleARN": "arn:aws:xray:us-west-2:123456789012:sampling-rule/
Default",
                "ResourceARN": "*",
                "Priority": 10000,
                "FixedRate": 0.01,
                "ReservoirSize": 0,
                "ServiceName": "*",
                "ServiceType": "*",
                "Host": "*",
                "HTTPMethod": "*",
                "URLPath": "*",
                "Version": 1,
                "Attributes": {}
            },
            "CreatedAt": 0.0,
            "ModifiedAt": 1529959993.0
        }
    ]
}
```

如需詳細資訊，請參閱《AWS X-Ray 開發人員指南》中的使用 X-Ray API 設定取樣、群組和加密設定。 AWS

- 如需 API 詳細資訊，請參閱《 AWS CLI 命令參考》中的 UpdateSamplingRule。

AWS CLI 搭配 Bash 指令碼程式碼範例

本主題中的程式碼範例說明如何 AWS Command Line Interface 搭配 Bash 指令碼使用 AWS。

基本概念是程式碼範例，這些範例說明如何在服務內執行基本操作。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

案例是向您展示如何呼叫服務中的多個函數或與其他 AWS 服務組合來完成特定任務的程式碼範例。

有些服務包含其他範例類別，示範如何利用服務特定的程式庫或函數。

服務

- 搭配 Bash 指令碼使用 AWS CLI 的 DynamoDB 範例
- 搭配 Bash 指令碼使用 AWS CLI 的 Amazon EC2 範例
- HealthImaging 範例 AWS CLI 搭配 Bash 指令碼使用
- 搭配 Bash 指令碼使用 AWS CLI 的 IAM 範例
- AWS KMSAWS CLI 搭配 Bash 指令碼使用 的範例
- 搭配 Bash 指令碼使用 AWS CLI 的 Amazon S3 範例
- AWS STSAWS CLI 搭配 Bash 指令碼使用 的範例

搭配 Bash 指令碼使用 AWS CLI 的 DynamoDB 範例

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Bash 指令碼搭配 DynamoDB 來執行動作和實作常見案例。

基本概念是程式碼範例，這些範例說明如何在服務內執行基本操作。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

案例是向您展示如何呼叫服務中的多個函數或與其他 AWS 服務組合來完成特定任務的程式碼範例。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 基本概念
- 動作

- 案例

基本概念

了解基本概念

以下程式碼範例顯示做法：

- 建立可存放電影資料的資料表。
- 放入、取得和更新資料表中的單個電影。
- 將影片資料從範例 JSON 檔案寫入資料表。
- 查詢特定年份發表的電影。
- 掃描某個年份範圍內發表的電影。
- 從資料表刪除電影，然後刪除資料表。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

DynamoDB 入門案例。

```
###############################################################################
# function dynamodb_getting_started_movies
#
# Scenario to create an Amazon DynamoDB table and perform a series of operations on
 the table.
#
# Returns:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function dynamodb_getting_started_movies() {

  source ./dynamodb_operations.sh
```

```
key_schema_json_file="dynamodb_key_schema.json"
attribute_definitions_json_file="dynamodb_attr_def.json"
item_json_file="movie_item.json"
key_json_file="movie_key.json"
batch_json_file="batch.json"
attribute_names_json_file="attribute_names.json"
attributes_values_json_file="attribute_values.json"

echo_repeat "*" 88
echo
echo "Welcome to the Amazon DynamoDB getting started demo."
echo
echo_repeat "*" 88
echo

local table_name
echo -n "Enter a name for a new DynamoDB table: "
get_input
table_name=$get_input_result

echo '[
{"AttributeName": "year", "KeyType": "HASH"},
 {"AttributeName": "title", "KeyType": "RANGE"}
]' >"$key_schema_json_file"

echo '[
{"AttributeName": "year", "AttributeType": "N"},
 {"AttributeName": "title", "AttributeType": "S"}
]' >"$attribute_definitions_json_file"

if dynamodb_create_table -n "$table_name" -a "$attribute_definitions_json_file" \
  -k "$key_schema_json_file" 1>/dev/null; then
  echo "Created a DynamoDB table named $table_name"
else
  errecho "The table failed to create. This demo will exit."
  clean_up
  return 1
fi

echo "Waiting for the table to become active...."

if dynamodb_wait_table_active -n "$table_name"; then
  echo "The table is now active."
else
```

```
    errecho "The table failed to become active. This demo will exit."
    cleanup "$table_name"
    return 1
  fi

  echo
  echo_repeat "*" 88
  echo

  echo -n "Enter the title of a movie you want to add to the table: "
  get_input
  local added_title
  added_title=$get_input_result

  local added_year
  get_int_input "What year was it released? "
  added_year=$get_input_result

  local rating
  get_float_input "On a scale of 1 - 10, how do you rate it? " "1" "10"
  rating=$get_input_result

  local plot
  echo -n "Summarize the plot for me: "
  get_input
  plot=$get_input_result

  echo '{
    "year": {"N" :"'"$added_year"'"},
    "title": {"S" :  "'"$added_title"'"},
    "info": {"M" : {"plot": {"S" : "'"$plot"'"}, "rating": {"N" :"'"$rating"'"} } }
   }' >"$item_json_file"

  if dynamodb_put_item -n "$table_name" -i "$item_json_file"; then
    echo "The movie '$added_title' was successfully added to the table
'$table_name'."
  else
    errecho "Put item failed. This demo will exit."
    clean_up "$table_name"
    return 1
  fi

  echo
  echo_repeat "*" 88
```

```
  echo

  echo "Let's update your movie '$added_title'."
  get_float_input "You rated it $rating, what new rating would you give it? " "1"
"10"
  rating=$get_input_result

  echo -n "You summarized the plot as '$plot'."
  echo "What would you say now? "
  get_input
  plot=$get_input_result

  echo '{
    "year": {"N" :"'"$added_year"'"},
    "title": {"S" :  "'"$added_title"'"}
    }' >"$key_json_file"

  echo '{
    ":r": {"N" :"'"$rating"'"},
    ":p": {"S" : "'"$plot"'"}
   }' >"$item_json_file"

  local update_expression="SET info.rating = :r, info.plot = :p"

  if dynamodb_update_item -n "$table_name" -k "$key_json_file" -e
"$update_expression" -v "$item_json_file"; then
    echo "Updated '$added_title' with new attributes."
  else
    errecho "Update item failed. This demo will exit."
    clean_up "$table_name"
    return 1
  fi

  echo
  echo_repeat "*" 88
  echo

  echo "We will now use batch write to upload 150 movie entries into the table."

  local batch_json
  for batch_json in movie_files/movies_*.json; do
    echo "{ \"$table_name\" : $(<"$batch_json") }" >"$batch_json_file"
    if dynamodb_batch_write_item -i "$batch_json_file" 1>/dev/null; then
      echo "Entries in $batch_json added to table."
```

```
    else
      errecho "Batch write failed. This demo will exit."
      clean_up "$table_name"
      return 1
    fi
  done

  local title="The Lord of the Rings: The Fellowship of the Ring"
  local year="2001"

  if get_yes_no_input "Let's move on...do you want to get info about '$title'? (y/n)
"; then
    echo '{
  "year": {"N" :"'"$year"'"},
  "title": {"S" :  "'"$title"'"}
  }' >"$key_json_file"
    local info
    info=$(dynamodb_get_item -n "$table_name" -k "$key_json_file")

    # shellcheck disable=SC2181
    if [[ ${?} -ne 0 ]]; then
      errecho "Get item failed. This demo will exit."
      clean_up "$table_name"
      return 1
    fi

    echo "Here is what I found:"
    echo "$info"
  fi

  local ask_for_year=true
  while [[ "$ask_for_year" == true ]]; do
    echo "Let's get a list of movies released in a given year."
    get_int_input "Enter a year between 1972 and 2018: " "1972" "2018"
    year=$get_input_result
    echo '{
    "#n": "year"
    }' >"$attribute_names_json_file"

    echo '{
    ":v": {"N" :"'"$year"'"}
    }' >"$attributes_values_json_file"
```

```
    response=$(dynamodb_query -n "$table_name" -k "#n=:v" -a
"$attribute_names_json_file" -v "$attributes_values_json_file")

    # shellcheck disable=SC2181
    if [[ ${?} -ne 0 ]]; then
      errecho "Query table failed. This demo will exit."
      clean_up "$table_name"
      return 1
    fi

    echo "Here is what I found:"
    echo "$response"

    if ! get_yes_no_input "Try another year? (y/n) "; then
      ask_for_year=false
    fi
  done

  echo "Now let's scan for movies released in a range of years. Enter a year: "
  get_int_input "Enter a year between 1972 and 2018: " "1972" "2018"
  local start=$get_input_result

  get_int_input "Enter another year: " "1972" "2018"
  local end=$get_input_result

  echo '{
    "#n": "year"
    }' >"$attribute_names_json_file"

  echo '{
    ":v1": {"N" : "'"$start"'"},
    ":v2": {"N" : "'"$end"'"}
    }' >"$attributes_values_json_file"

  response=$(dynamodb_scan -n "$table_name" -f "#n BETWEEN :v1 AND :v2" -a
"$attribute_names_json_file" -v "$attributes_values_json_file")

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "Scan table failed. This demo will exit."
    clean_up "$table_name"
    return 1
  fi
```

```
  echo "Here is what I found:"
  echo "$response"

  echo
  echo_repeat "*" 88
  echo

  echo "Let's remove your movie '$added_title' from the table."

  if get_yes_no_input "Do you want to remove '$added_title'? (y/n) "; then
    echo '{
  "year": {"N" :"'"$added_year"'"},
  "title": {"S" :  "'"$added_title"'"}
  }' >"$key_json_file"

    if ! dynamodb_delete_item -n "$table_name" -k "$key_json_file"; then
      errecho "Delete item failed. This demo will exit."
      clean_up "$table_name"
      return 1
    fi
  fi

  if get_yes_no_input "Do you want to delete the table '$table_name'? (y/n) "; then
    if ! clean_up "$table_name"; then
      return 1
    fi
  else
    if ! clean_up; then
      return 1
    fi
  fi

  return 0
}
```

此案例中使用的 DynamoDB 函數。

```
###############################################################################
# function dynamodb_create_table
#
# This function creates an Amazon DynamoDB table.
#
```

```
# Parameters:
#       -n table_name  -- The name of the table to create.
#       -a attribute_definitions -- JSON file path of a list of attributes and their
 types.
#       -k key_schema -- JSON file path of a list of attributes and their key types.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_create_table() {
  local table_name attribute_definitions key_schema response
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_create_table"
    echo "Creates an Amazon DynamoDB table with on-demand billing."
    echo " -n table_name  -- The name of the table to create."
    echo " -a attribute_definitions -- JSON file path of a list of attributes and
 their types."
    echo " -k key_schema -- JSON file path of a list of attributes and their key
 types."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:a:k:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      a) attribute_definitions="${OPTARG}" ;;
      k) key_schema="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
```

```
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$attribute_definitions" ]]; then
    errecho "ERROR: You must provide an attribute definitions json file path the -a
 parameter."
    usage
    return 1
  fi

  if [[ -z "$key_schema" ]]; then
    errecho "ERROR: You must provide a key schema json file path the -k parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    attribute_definitions:   $attribute_definitions"
  iecho "    key_schema:   $key_schema"
  iecho ""

  response=$(aws dynamodb create-table \
    --table-name "$table_name" \
    --attribute-definitions file://"$attribute_definitions" \
    --billing-mode PAY_PER_REQUEST \
    --key-schema file://"$key_schema" )

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-table operation failed.$response"
    return 1
  fi

  return 0
}
```

```
###############################################################################
# function dynamodb_describe_table
#
# This function returns the status of a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#
#  Response:
#       - TableStatus:
#     And:
#       0 - Table is active.
#       1 - If it fails.
###############################################################################
function dynamodb_describe_table {
  local table_name
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_describe_table"
    echo "Describe the status of a DynamoDB table."
    echo "  -n table_name  -- The name of the table."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
```

```
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi

  local table_status
    table_status=$(
      aws dynamodb describe-table \
        --table-name "$table_name" \
        --output text \
        --query 'Table.TableStatus'
    )

   local error_code=${?}

    if [[ $error_code -ne 0 ]]; then
      aws_cli_error_log "$error_code"
      errecho "ERROR: AWS reports describe-table operation failed.$table_status"
      return 1
    fi

  echo "$table_status"

  return 0
}

###############################################################################
# function dynamodb_put_item
#
# This function puts an item into a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -i item  -- Path to json file containing the item values.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_put_item() {
  local table_name item response
```

```
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_put_item"
    echo "Put an item into a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -i item  -- Path to json file containing the item values."
    echo ""
  }

  while getopts "n:i:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      i) item="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$item" ]]; then
    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
```

```
  iecho "    item:   $item"
  iecho ""
  iecho ""

  response=$(aws dynamodb put-item \
    --table-name "$table_name" \
    --item file://"$item")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports put-item operation failed.$response"
    return 1
  fi

  return 0

}

###############################################################################
# function dynamodb_update_item
#
# This function updates an item in a DynamoDB table.
#
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -k keys  -- Path to json file containing the keys that identify the item to
 update.
#       -e update expression  -- An expression that defines one or more attributes
 to be updated.
#       -v values  -- Path to json file containing the update values.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_update_item() {
  local table_name keys update_expression values response
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
```

```
 ######################################
 function usage() {
   echo "function dynamodb_update_item"
   echo "Update an item in a DynamoDB table."
   echo " -n table_name  -- The name of the table."
   echo " -k keys  -- Path to json file containing the keys that identify the item
to update."
   echo " -e update expression  -- An expression that defines one or more
attributes to be updated."
   echo " -v values  -- Path to json file containing the update values."
   echo ""
 }

 while getopts "n:k:e:v:h" option; do
   case "${option}" in
     n) table_name="${OPTARG}" ;;
     k) keys="${OPTARG}" ;;
     e) update_expression="${OPTARG}" ;;
     v) values="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$table_name" ]]; then
   errecho "ERROR: You must provide a table name with the -n parameter."
   usage
   return 1
 fi

 if [[ -z "$keys" ]]; then
   errecho "ERROR: You must provide a keys json file path the -k parameter."
   usage
   return 1
 fi
 if [[ -z "$update_expression" ]]; then
```

```
    errecho "ERROR: You must provide an update expression with the -e parameter."
    usage
    return 1
  fi

  if [[ -z "$values" ]]; then
    errecho "ERROR: You must provide a values json file path the -v parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    keys:   $keys"
  iecho "    update_expression:   $update_expression"
  iecho "    values:   $values"

  response=$(aws dynamodb update-item \
    --table-name "$table_name" \
    --key file://"$keys" \
    --update-expression "$update_expression" \
    --expression-attribute-values file://"$values")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports update-item operation failed.$response"
    return 1
  fi

  return 0

}

###############################################################################
# function dynamodb_batch_write_item
#
# This function writes a batch of items into a DynamoDB table.
#
# Parameters:
#       -i item  -- Path to json file containing the items to write.
#
#  Returns:
```

```
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_batch_write_item() {
  local item response
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_batch_write_item"
    echo "Write a batch of items into a DynamoDB table."
    echo " -i item  -- Path to json file containing the items to write."
    echo ""
  }
  while getopts "i:h" option; do
    case "${option}" in
      i) item="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$item" ]]; then
    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    item:   $item"
  iecho ""

  response=$(aws dynamodb batch-write-item \
```

```
    --request-items file://"$item")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports batch-write-item operation failed.$response"
    return 1
  fi

  return 0
}

###############################################################################
# function dynamodb_get_item
#
# This function gets an item from a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -k keys  -- Path to json file containing the keys that identify the item to
 get.
#       [-q query]  -- Optional JMESPath query expression.
#
#  Returns:
#       The item as text output.
#  And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_get_item() {
  local table_name keys query response
  local option OPTARG # Required to use getopts command in a function.

  # ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_get_item"
    echo "Get an item from a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -k keys  -- Path to json file containing the keys that identify the item
 to get."
    echo " [-q query]  -- Optional JMESPath query expression."
```

```
  echo ""
}
query=""
while getopts "n:k:q:h" option; do
  case "${option}" in
    n) table_name="${OPTARG}" ;;
    k) keys="${OPTARG}" ;;
    q) query="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
  errecho "ERROR: You must provide a table name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$keys" ]]; then
  errecho "ERROR: You must provide a keys json file path the -k parameter."
  usage
  return 1
fi

if [[ -n "$query" ]]; then
  response=$(aws dynamodb get-item \
    --table-name "$table_name" \
    --key file://"$keys" \
    --output text \
    --query "$query")
else
  response=$(
    aws dynamodb get-item \
      --table-name "$table_name" \
      --key file://"$keys" \
```

```
        --output text
    )
  fi

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports get-item operation failed.$response"
    return 1
  fi

  if [[ -n "$query" ]]; then
    echo "$response" | sed "/^\t/s/\t//1" # Remove initial tab that the JMSEPath
 query inserts on some strings.
  else
    echo "$response"
  fi

  return 0
}

###############################################################################
# function dynamodb_query
#
# This function queries a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -k key_condition_expression -- The key condition expression.
#       -a attribute_names -- Path to JSON file containing the attribute names.
#       -v attribute_values -- Path to JSON file containing the attribute values.
#       [-p projection_expression]  -- Optional projection expression.
#
#  Returns:
#       The items as json output.
#  And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_query() {
  local table_name key_condition_expression attribute_names attribute_values
 projection_expression response
  local option OPTARG # Required to use getopts command in a function.
```

```
 # ######################################
 # Function usage explanation
 ######################################
 function usage() {
   echo "function dynamodb_query"
   echo "Query a DynamoDB table."
   echo " -n table_name  -- The name of the table."
   echo " -k key_condition_expression -- The key condition expression."
   echo " -a attribute_names -- Path to JSON file containing the attribute names."
   echo " -v attribute_values -- Path to JSON file containing the attribute
values."
   echo " [-p projection_expression]  -- Optional projection expression."
   echo ""
 }

 while getopts "n:k:a:v:p:h" option; do
   case "${option}" in
     n) table_name="${OPTARG}" ;;
     k) key_condition_expression="${OPTARG}" ;;
     a) attribute_names="${OPTARG}" ;;
     v) attribute_values="${OPTARG}" ;;
     p) projection_expression="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$table_name" ]]; then
   errecho "ERROR: You must provide a table name with the -n parameter."
   usage
   return 1
 fi

 if [[ -z "$key_condition_expression" ]]; then
```

```
    errecho "ERROR: You must provide a key condition expression with the -k
parameter."
    usage
    return 1
  fi

  if [[ -z "$attribute_names" ]]; then
    errecho "ERROR: You must provide a attribute names with the -a parameter."
    usage
    return 1
  fi

  if [[ -z "$attribute_values" ]]; then
    errecho "ERROR: You must provide a attribute values with the -v parameter."
    usage
    return 1
  fi

  if [[ -z "$projection_expression" ]]; then
    response=$(aws dynamodb query \
      --table-name "$table_name" \
      --key-condition-expression "$key_condition_expression" \
      --expression-attribute-names file://"$attribute_names" \
      --expression-attribute-values file://"$attribute_values")
  else
    response=$(aws dynamodb query \
      --table-name "$table_name" \
      --key-condition-expression "$key_condition_expression" \
      --expression-attribute-names file://"$attribute_names" \
      --expression-attribute-values file://"$attribute_values" \
      --projection-expression "$projection_expression")
  fi

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports query operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
```

```
}

###############################################################################
# function dynamodb_scan
#
# This function scans a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -f filter_expression  -- The filter expression.
#       -a expression_attribute_names -- Path to JSON file containing the expression
 attribute names.
#       -v expression_attribute_values -- Path to JSON file containing the
 expression attribute values.
#       [-p projection_expression]  -- Optional projection expression.
#
#  Returns:
#       The items as json output.
#  And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_scan() {
  local table_name filter_expression expression_attribute_names
 expression_attribute_values projection_expression response
  local option OPTARG # Required to use getopts command in a function.

  # ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_scan"
    echo "Scan a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -f filter_expression  -- The filter expression."
    echo " -a expression_attribute_names -- Path to JSON file containing the
 expression attribute names."
    echo " -v expression_attribute_values -- Path to JSON file containing the
 expression attribute values."
    echo " [-p projection_expression]  -- Optional projection expression."
    echo ""
  }

  while getopts "n:f:a:v:p:h" option; do
```

```
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      f) filter_expression="${OPTARG}" ;;
      a) expression_attribute_names="${OPTARG}" ;;
      v) expression_attribute_values="${OPTARG}" ;;
      p) projection_expression="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$filter_expression" ]]; then
    errecho "ERROR: You must provide a filter expression with the -f parameter."
    usage
    return 1
  fi

  if [[ -z "$expression_attribute_names" ]]; then
    errecho "ERROR: You must provide expression attribute names with the -a
parameter."
    usage
    return 1
  fi

  if [[ -z "$expression_attribute_values" ]]; then
    errecho "ERROR: You must provide expression attribute values with the -v
parameter."
    usage
    return 1
  fi
```

```
  if [[ -z "$projection_expression" ]]; then
    response=$(aws dynamodb scan \
      --table-name "$table_name" \
      --filter-expression "$filter_expression" \
      --expression-attribute-names file://"$expression_attribute_names" \
      --expression-attribute-values file://"$expression_attribute_values")
  else
    response=$(aws dynamodb scan \
      --table-name "$table_name" \
      --filter-expression "$filter_expression" \
      --expression-attribute-names file://"$expression_attribute_names" \
      --expression-attribute-values file://"$expression_attribute_values" \
      --projection-expression "$projection_expression")
  fi

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports scan operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function dynamodb_delete_item
#
# This function deletes an item from a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -k keys  -- Path to json file containing the keys that identify the item to
 delete.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_delete_item() {
```

```
  local table_name keys response
  local option OPTARG # Required to use getopts command in a function.

  # ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_delete_item"
    echo "Delete an item from a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -k keys  -- Path to json file containing the keys that identify the item
to delete."
    echo ""
  }
  while getopts "n:k:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      k) keys="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$keys" ]]; then
    errecho "ERROR: You must provide a keys json file path the -k parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
```

```
  iecho "    table_name:   $table_name"
  iecho "    keys:   $keys"
  iecho ""

  response=$(aws dynamodb delete-item \
    --table-name "$table_name" \
    --key file://"$keys")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-item operation failed.$response"
    return 1
  fi

  return 0

}

###############################################################################
# function dynamodb_delete_table
#
# This function deletes a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table to delete.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_delete_table() {
  local table_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function dynamodb_delete_table"
    echo "Deletes an Amazon DynamoDB table."
    echo " -n table_name  -- The name of the table to delete."
    echo ""
  }
```

```
  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho ""

  response=$(aws dynamodb delete-table \
    --table-name "$table_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-table operation failed.$response"
    return 1
  fi

  return 0
}
```

此案例中使用的公用程式函數。

```
#############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
#############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
```

```
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - BatchWriteItem
  - CreateTable
  - DeleteItem
  - DeleteTable
  - DescribeTable
  - GetItem
  - PutItem
  - 查詢
  - 掃描
  - UpdateItem

動作

`BatchGetItem`

以下程式碼範例顯示如何使用 `BatchGetItem`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function dynamodb_batch_get_item
#
# This function gets a batch of items from a DynamoDB table.
#
# Parameters:
#       -i item  -- Path to json file containing the keys of the items to get.
#
#  Returns:
#       The items as json output.
#  And:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function dynamodb_batch_get_item() {
  local item response
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_batch_get_item"
    echo "Get a batch of items from a DynamoDB table."
    echo " -i item  -- Path to json file containing the keys of the items to get."
    echo ""
  }

  while getopts "i:h" option; do
    case "${option}" in
      i) item="${OPTARG}" ;;
      h)
        usage
        return 0
```

```
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$item" ]]; then
    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
  fi

  response=$(aws dynamodb batch-get-item \
    --request-items file://"$item")
  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports batch-get-item operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

此範例中使用的公用程式函數。

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}
```

```
###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 BatchGetItem。

`BatchWriteItem`

以下程式碼範例顯示如何使用 `BatchWriteItem`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
#############################################################################
# function dynamodb_batch_write_item
#
# This function writes a batch of items into a DynamoDB table.
#
# Parameters:
#       -i item  -- Path to json file containing the items to write.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
#############################################################################
function dynamodb_batch_write_item() {
  local item response
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_batch_write_item"
    echo "Write a batch of items into a DynamoDB table."
    echo " -i item  -- Path to json file containing the items to write."
    echo ""
  }
  while getopts "i:h" option; do
    case "${option}" in
      i) item="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
```

```
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$item" ]]; then
    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    item:   $item"
  iecho ""

  response=$(aws dynamodb batch-write-item \
    --request-items file://"$item")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports batch-write-item operation failed.$response"
    return 1
  fi

  return 0
}
```

此範例中使用的公用程式函數。

```
###############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
###############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
```

```
    echo "$@"
  fi
}

############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
```

```
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 BatchWriteItem。

`CreateTable`

以下程式碼範例顯示如何使用 `CreateTable`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function dynamodb_create_table
#
# This function creates an Amazon DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table to create.
#       -a attribute_definitions -- JSON file path of a list of attributes and their
 types.
#       -k key_schema -- JSON file path of a list of attributes and their key types.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_create_table() {
  local table_name attribute_definitions key_schema response
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
```

```
 ######################################
 function usage() {
   echo "function dynamodb_create_table"
   echo "Creates an Amazon DynamoDB table with on-demand billing."
   echo " -n table_name  -- The name of the table to create."
   echo " -a attribute_definitions -- JSON file path of a list of attributes and
their types."
   echo " -k key_schema -- JSON file path of a list of attributes and their key
types."
   echo ""
 }

 # Retrieve the calling parameters.
 while getopts "n:a:k:h" option; do
   case "${option}" in
     n) table_name="${OPTARG}" ;;
     a) attribute_definitions="${OPTARG}" ;;
     k) key_schema="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$table_name" ]]; then
   errecho "ERROR: You must provide a table name with the -n parameter."
   usage
   return 1
 fi

 if [[ -z "$attribute_definitions" ]]; then
   errecho "ERROR: You must provide an attribute definitions json file path the -a
parameter."
   usage
   return 1
 fi
```

```
  if [[ -z "$key_schema" ]]; then
    errecho "ERROR: You must provide a key schema json file path the -k parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    attribute_definitions:   $attribute_definitions"
  iecho "    key_schema:   $key_schema"
  iecho ""

  response=$(aws dynamodb create-table \
    --table-name "$table_name" \
    --attribute-definitions file://"$attribute_definitions" \
    --billing-mode PAY_PER_REQUEST \
    --key-schema file://"$key_schema" )

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-table operation failed.$response"
    return 1
  fi

  return 0
}
```

此範例中使用的公用程式函數。

```
###############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
###############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}
```

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi
```

```
  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateTable。

`DeleteItem`

以下程式碼範例顯示如何使用 `DeleteItem`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function dynamodb_delete_item
#
# This function deletes an item from a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -k keys  -- Path to json file containing the keys that identify the item to
 delete.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_delete_item() {
  local table_name keys response
  local option OPTARG # Required to use getopts command in a function.

  # ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_delete_item"
    echo "Delete an item from a DynamoDB table."
```

```
    echo " -n table_name  -- The name of the table."
    echo " -k keys  -- Path to json file containing the keys that identify the item
to delete."
    echo ""
 }
 while getopts "n:k:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      k) keys="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
 done
 export OPTIND=1

 if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
 fi

 if [[ -z "$keys" ]]; then
    errecho "ERROR: You must provide a keys json file path the -k parameter."
    usage
    return 1
 fi

 iecho "Parameters:\n"
 iecho "    table_name:   $table_name"
 iecho "    keys:   $keys"
 iecho ""

 response=$(aws dynamodb delete-item \
    --table-name "$table_name" \
    --key file://"$keys")

 local error_code=${?}
```

```
  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-item operation failed.$response"
    return 1
  fi

  return 0

}
```

此範例中使用的公用程式函數。

```
#############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
```

```
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteItem。

`DeleteTable`

以下程式碼範例顯示如何使用 `DeleteTable`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function dynamodb_delete_table
#
# This function deletes a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table to delete.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_delete_table() {
  local table_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function dynamodb_delete_table"
    echo "Deletes an Amazon DynamoDB table."
    echo " -n table_name  -- The name of the table to delete."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
```

```
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho ""

  response=$(aws dynamodb delete-table \
    --table-name "$table_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-table operation failed.$response"
    return 1
  fi

  return 0
}
```

此範例中使用的公用程式函數。

```
###############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
###############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
```

```
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteTable。

`DescribeTable`

以下程式碼範例顯示如何使用 `DescribeTable`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function dynamodb_describe_table
#
# This function returns the status of a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#
#  Response:
#       - TableStatus:
#     And:
#       0 - Table is active.
#       1 - If it fails.
###############################################################################
function dynamodb_describe_table {
  local table_name
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_describe_table"
    echo "Describe the status of a DynamoDB table."
    echo "  -n table_name  -- The name of the table."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
```

```
      n) table_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi

  local table_status
    table_status=$(
      aws dynamodb describe-table \
        --table-name "$table_name" \
        --output text \
        --query 'Table.TableStatus'
    )

   local error_code=${?}

    if [[ $error_code -ne 0 ]]; then
      aws_cli_error_log "$error_code"
      errecho "ERROR: AWS reports describe-table operation failed.$table_status"
      return 1
    fi

  echo "$table_status"

  return 0
}
```

此範例中使用的公用程式函數。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
##############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi
```

```
  return 0
}
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeTable。

`GetItem`

以下程式碼範例顯示如何使用 `GetItem`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function dynamodb_get_item
#
# This function gets an item from a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -k keys  -- Path to json file containing the keys that identify the item to
 get.
#       [-q query]  -- Optional JMESPath query expression.
#
#  Returns:
#       The item as text output.
#  And:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function dynamodb_get_item() {
  local table_name keys query response
  local option OPTARG # Required to use getopts command in a function.

  # ######################################
  # Function usage explanation
  ######################################
```

```
 function usage() {
   echo "function dynamodb_get_item"
   echo "Get an item from a DynamoDB table."
   echo " -n table_name  -- The name of the table."
   echo " -k keys  -- Path to json file containing the keys that identify the item
to get."
   echo " [-q query]  -- Optional JMESPath query expression."
   echo ""
 }
 query=""
 while getopts "n:k:q:h" option; do
   case "${option}" in
     n) table_name="${OPTARG}" ;;
     k) keys="${OPTARG}" ;;
     q) query="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$table_name" ]]; then
   errecho "ERROR: You must provide a table name with the -n parameter."
   usage
   return 1
 fi

 if [[ -z "$keys" ]]; then
   errecho "ERROR: You must provide a keys json file path the -k parameter."
   usage
   return 1
 fi

 if [[ -n "$query" ]]; then
   response=$(aws dynamodb get-item \
     --table-name "$table_name" \
     --key file://"$keys" \
```

```
      --output text \
      --query "$query")
  else
    response=$(
      aws dynamodb get-item \
        --table-name "$table_name" \
        --key file://"$keys" \
        --output text
    )
  fi

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports get-item operation failed.$response"
    return 1
  fi

  if [[ -n "$query" ]]; then
    echo "$response" | sed "/^\t/s/\t//1" # Remove initial tab that the JMSEPath
 query inserts on some strings.
  else
    echo "$response"
  fi

  return 0
}
```

此範例中使用的公用程式函數。

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
```

```
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
##############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetItem。

ListTables

以下程式碼範例顯示如何使用 ListTables。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function dynamodb_list_tables
#
# This function lists all the tables in a DynamoDB.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function dynamodb_list_tables() {
  response=$(aws dynamodb list-tables \
    --output text \
    --query "TableNames")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports batch-write-item operation failed.$response"
    return 1
  fi

  echo "$response" | tr -s "[:space:]" "\n"

  return 0
}
```

此範例中使用的公用程式函數。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
```

```
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListTables。

`PutItem`

以下程式碼範例顯示如何使用 `PutItem`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function dynamodb_put_item
#
# This function puts an item into a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -i item  -- Path to json file containing the item values.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function dynamodb_put_item() {
  local table_name item response
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_put_item"
    echo "Put an item into a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -i item  -- Path to json file containing the item values."
    echo ""
  }

  while getopts "n:i:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
```

```
    i) item="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
  errecho "ERROR: You must provide a table name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$item" ]]; then
  errecho "ERROR: You must provide an item with the -i parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "    table_name:   $table_name"
iecho "    item:   $item"
iecho ""
iecho ""

response=$(aws dynamodb put-item \
  --table-name "$table_name" \
  --item file://"$item")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports put-item operation failed.$response"
  return 1
fi
```

```
  return 0

}
```

此範例中使用的公用程式函數。

```
################################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
################################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

################################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
################################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

################################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
################################################################################
```

```
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutObject。

Query

以下程式碼範例顯示如何使用 Query。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
#############################################################################
# function dynamodb_query
#
# This function queries a DynamoDB table.
#
```

```
# Parameters:
#       -n table_name  -- The name of the table.
#       -k key_condition_expression -- The key condition expression.
#       -a attribute_names -- Path to JSON file containing the attribute names.
#       -v attribute_values -- Path to JSON file containing the attribute values.
#       [-p projection_expression]  -- Optional projection expression.
#
#  Returns:
#       The items as json output.
#  And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_query() {
  local table_name key_condition_expression attribute_names attribute_values
 projection_expression response
  local option OPTARG # Required to use getopts command in a function.

  # ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_query"
    echo "Query a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -k key_condition_expression -- The key condition expression."
    echo " -a attribute_names -- Path to JSON file containing the attribute names."
    echo " -v attribute_values -- Path to JSON file containing the attribute
 values."
    echo " [-p projection_expression]  -- Optional projection expression."
    echo ""
  }

  while getopts "n:k:a:v:p:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      k) key_condition_expression="${OPTARG}" ;;
      a) attribute_names="${OPTARG}" ;;
      v) attribute_values="${OPTARG}" ;;
      p) projection_expression="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
```

```
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$key_condition_expression" ]]; then
    errecho "ERROR: You must provide a key condition expression with the -k
parameter."
    usage
    return 1
  fi

  if [[ -z "$attribute_names" ]]; then
    errecho "ERROR: You must provide a attribute names with the -a parameter."
    usage
    return 1
  fi

  if [[ -z "$attribute_values" ]]; then
    errecho "ERROR: You must provide a attribute values with the -v parameter."
    usage
    return 1
  fi

  if [[ -z "$projection_expression" ]]; then
    response=$(aws dynamodb query \
      --table-name "$table_name" \
      --key-condition-expression "$key_condition_expression" \
      --expression-attribute-names file://"$attribute_names" \
      --expression-attribute-values file://"$attribute_values")
  else
    response=$(aws dynamodb query \
      --table-name "$table_name" \
      --key-condition-expression "$key_condition_expression" \
```

```
      --expression-attribute-names file://"$attribute_names" \
      --expression-attribute-values file://"$attribute_values" \
      --projection-expression "$projection_expression")
  fi

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports query operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

此範例中使用的公用程式函數。

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
```

```
#
################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 Query。

Scan

以下程式碼範例顯示如何使用 Scan。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
################################################################################
# function dynamodb_scan
#
```

```
# This function scans a DynamoDB table.
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -f filter_expression  -- The filter expression.
#       -a expression_attribute_names -- Path to JSON file containing the expression
 attribute names.
#       -v expression_attribute_values -- Path to JSON file containing the
 expression attribute values.
#       [-p projection_expression]  -- Optional projection expression.
#
#  Returns:
#       The items as json output.
#  And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_scan() {
  local table_name filter_expression expression_attribute_names
 expression_attribute_values projection_expression response
  local option OPTARG # Required to use getopts command in a function.

  # ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_scan"
    echo "Scan a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -f filter_expression  -- The filter expression."
    echo " -a expression_attribute_names -- Path to JSON file containing the
 expression attribute names."
    echo " -v expression_attribute_values -- Path to JSON file containing the
 expression attribute values."
    echo " [-p projection_expression]  -- Optional projection expression."
    echo ""
  }

  while getopts "n:f:a:v:p:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      f) filter_expression="${OPTARG}" ;;
      a) expression_attribute_names="${OPTARG}" ;;
      v) expression_attribute_values="${OPTARG}" ;;
```

```
      p) projection_expression="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$filter_expression" ]]; then
    errecho "ERROR: You must provide a filter expression with the -f parameter."
    usage
    return 1
  fi

  if [[ -z "$expression_attribute_names" ]]; then
    errecho "ERROR: You must provide expression attribute names with the -a
parameter."
    usage
    return 1
  fi

  if [[ -z "$expression_attribute_values" ]]; then
    errecho "ERROR: You must provide expression attribute values with the -v
parameter."
    usage
    return 1
  fi

  if [[ -z "$projection_expression" ]]; then
    response=$(aws dynamodb scan \
      --table-name "$table_name" \
      --filter-expression "$filter_expression" \
```

```
      --expression-attribute-names file://"$expression_attribute_names" \
      --expression-attribute-values file://"$expression_attribute_values")
  else
    response=$(aws dynamodb scan \
      --table-name "$table_name" \
      --filter-expression "$filter_expression" \
      --expression-attribute-names file://"$expression_attribute_names" \
      --expression-attribute-values file://"$expression_attribute_values" \
      --projection-expression "$projection_expression")
  fi

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports scan operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

此範例中使用的公用程式函數。

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
```

```
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 Scan。

`UpdateItem`

以下程式碼範例顯示如何使用 `UpdateItem`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function dynamodb_update_item
#
# This function updates an item in a DynamoDB table.
#
#
# Parameters:
#       -n table_name  -- The name of the table.
#       -k keys  -- Path to json file containing the keys that identify the item to
 update.
#       -e update expression  -- An expression that defines one or more attributes
 to be updated.
#       -v values  -- Path to json file containing the update values.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function dynamodb_update_item() {
  local table_name keys update_expression values response
  local option OPTARG # Required to use getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_update_item"
    echo "Update an item in a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -k keys  -- Path to json file containing the keys that identify the item
 to update."
    echo " -e update expression  -- An expression that defines one or more
 attributes to be updated."
    echo " -v values  -- Path to json file containing the update values."
```

```
  echo ""
}

while getopts "n:k:e:v:h" option; do
  case "${option}" in
    n) table_name="${OPTARG}" ;;
    k) keys="${OPTARG}" ;;
    e) update_expression="${OPTARG}" ;;
    v) values="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
  errecho "ERROR: You must provide a table name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$keys" ]]; then
  errecho "ERROR: You must provide a keys json file path the -k parameter."
  usage
  return 1
fi
if [[ -z "$update_expression" ]]; then
  errecho "ERROR: You must provide an update expression with the -e parameter."
  usage
  return 1
fi

if [[ -z "$values" ]]; then
  errecho "ERROR: You must provide a values json file path the -v parameter."
  usage
  return 1
fi
```

```
  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    keys:   $keys"
  iecho "    update_expression:   $update_expression"
  iecho "    values:   $values"

  response=$(aws dynamodb update-item \
    --table-name "$table_name" \
    --key file://"$keys" \
    --update-expression "$update_expression" \
    --expression-attribute-values file://"$values")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports update-item operation failed.$response"
    return 1
  fi

  return 0

}
```

此範例中使用的公用程式函數。

```
###############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
###############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

###############################################################################
# function errecho
#
```

```
# This function outputs everything sent to it to STDERR (standard error output).
#########################################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#########################################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
#########################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateItem。

案例

進階全域次要索引案例

下列程式碼範例示範如何使用進階全域次要索引組態。

- 建立具有多個 GSIs資料表。
- 建立具有隨需容量和 GSI 的資料表。
- 將項目放入具有多個 GSIs資料表中。
- 查詢具有不同條件GSIs。

AWS CLI 搭配 Bash 指令碼

建立具有多個 GSIs資料表。

```
# Create a table with multiple GSIs
aws dynamodb create-table \
    --table-name MusicLibrary \
    --attribute-definitions \
        AttributeName=Artist,AttributeType=S \
        AttributeName=SongTitle,AttributeType=S \
        AttributeName=AlbumTitle,AttributeType=S \
        AttributeName=Genre,AttributeType=S \
        AttributeName=Year,AttributeType=N \
    --key-schema \
        AttributeName=Artist,KeyType=HASH \
        AttributeName=SongTitle,KeyType=RANGE \
    --billing-mode PAY_PER_REQUEST \
    --global-secondary-indexes \
        "[
            {
                \"IndexName\": \"AlbumIndex\",
                \"KeySchema\": [{\"AttributeName\":\"AlbumTitle\",\"KeyType\":\"HASH
\"}],
                \"Projection\": {\"ProjectionType\":\"ALL\"}
            },
            {
                \"IndexName\": \"GenreYearIndex\",
                \"KeySchema\": [
```

```
                    {\"AttributeName\":\"Genre\",\"KeyType\":\"HASH\"},
                    {\"AttributeName\":\"Year\",\"KeyType\":\"RANGE\"}
                ],
                \"Projection\": {\"ProjectionType\":\"INCLUDE\",\"NonKeyAttributes
\":[\"Artist\",\"SongTitle\"]}
            }
        ]"
```

建立具有隨需容量和 GSI 的資料表。

```
# Create a table with on-demand capacity and GSI
aws dynamodb create-table \
    --table-name MusicOnDemand \
    --attribute-definitions \
        AttributeName=Artist,AttributeType=S \
        AttributeName=SongTitle,AttributeType=S \
        AttributeName=Genre,AttributeType=S \
    --key-schema \
        AttributeName=Artist,KeyType=HASH \
        AttributeName=SongTitle,KeyType=RANGE \
    --billing-mode PAY_PER_REQUEST \
    --global-secondary-indexes \
        "[
            {
                \"IndexName\": \"GenreIndex\",
                \"KeySchema\": [{\"AttributeName\":\"Genre\",\"KeyType\":\"HASH\"}],
                \"Projection\": {\"ProjectionType\":\"ALL\"}
            }
        ]"
```

將項目放入具有多個 GSIs資料表中。

```
# Add items to MusicLibrary table
aws dynamodb put-item \
    --table-name MusicLibrary \
    --item '{
        "Artist": {"S": "The Beatles"},
        "SongTitle": {"S": "Hey Jude"},
        "AlbumTitle": {"S": "Past Masters"},
        "Genre": {"S": "Rock"},
        "Year": {"N": "1968"}
```

```
    }'

aws dynamodb put-item \
    --table-name MusicLibrary \
    --item '{
        "Artist": {"S": "Miles Davis"},
        "SongTitle": {"S": "So What"},
        "AlbumTitle": {"S": "Kind of Blue"},
        "Genre": {"S": "Jazz"},
        "Year": {"N": "1959"}
    }'
```

從具有多個 GSIs資料表查詢項目。

```
# Query the AlbumIndex GSI
echo "Querying AlbumIndex GSI:"
aws dynamodb query \
    --table-name MusicLibrary \
    --index-name AlbumIndex \
    --key-condition-expression "AlbumTitle = :album" \
    --expression-attribute-values '{":album":{"S":"Kind of Blue"}}'

# Query the GenreYearIndex GSI with a range condition
echo "Querying GenreYearIndex GSI with range condition:"
aws dynamodb query \
    --table-name MusicLibrary \
    --index-name GenreYearIndex \
    --key-condition-expression "Genre = :genre AND #yr > :year" \
    --expression-attribute-names '{"#yr": "Year"}' \
    --expression-attribute-values '{":genre":{"S":"Rock"},":year":{"N":"1965"}}'
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - CreateTable
  - PutItem
  - 查詢

建立和管理 MRSC 全域資料表

下列程式碼範例示範如何使用多區域強一致性 (MRSC) 建立和管理 DynamoDB 全域資料表。

- 建立具有多區域強一致性的資料表。
- 驗證 MRSC 組態和複本狀態。
- 使用即時讀取測試跨區域的強大一致性。
- 使用 MRSC 保證執行條件式寫入。
- 清除 MRSC 全域資料表資源。

AWS CLI 搭配 Bash 指令碼

建立具有多區域強一致性的資料表。

```
# Step 1: Create a new table in us-east-2 (primary region for MRSC)
# Note: Table must be empty when enabling MRSC
aws dynamodb create-table \
    --table-name MusicTable \
    --attribute-definitions \
        AttributeName=Artist,AttributeType=S \
        AttributeName=SongTitle,AttributeType=S \
    --key-schema \
        AttributeName=Artist,KeyType=HASH \
        AttributeName=SongTitle,KeyType=RANGE \
    --billing-mode PAY_PER_REQUEST \
    --region us-east-2

# Wait for table to become active
aws dynamodb wait table-exists --table-name MusicTable --region us-east-2

# Step 2: Add replica and witness with Multi-Region Strong Consistency
# MRSC requires exactly three replicas in supported regions
aws dynamodb update-table \
    --table-name MusicTable \
    --replica-updates '[{"Create": {"RegionName": "us-east-1"}}]' \
    --global-table-witness-updates '[{"Create": {"RegionName": "us-west-2"}}]' \
    --multi-region-consistency STRONG \
    --region us-east-2
```

驗證 MRSC 組態和複本狀態。

```
# Verify the global table configuration and MRSC setting
aws dynamodb describe-table \
```

```
    --table-name MusicTable \
    --region us-east-2 \
    --query 'Table.
{TableName:TableName,TableStatus:TableStatus,MultiRegionConsistency:MultiRegionConsistency,R
{Region:RegionName,Status:ReplicaStatus}}'
```

測試跨 區域的即時讀取的強式一致性。

```
# Write an item to the primary region
aws dynamodb put-item \
    --table-name MusicTable \
    --item '{"Artist": {"S":"The Beatles"},"SongTitle": {"S":"Hey Jude"},"Album":
 {"S":"The Beatles 1967-1970"},"Year": {"N":"1968"}}' \
    --region us-east-2

# Read the item from replica region to verify strong consistency (cannot read or
 write to witness)
# No wait time needed - MRSC provides immediate consistency
echo "Reading from us-east-1 (immediate consistency):"
aws dynamodb get-item \
    --table-name MusicTable \
    --key '{"Artist": {"S":"The Beatles"},"SongTitle": {"S":"Hey Jude"}}' \
    --consistent-read \
    --region us-east-1
```

使用 MRSC 保證執行條件式寫入。

```
# Perform a conditional update from a different region
# This demonstrates that conditions work consistently across all regions
aws dynamodb update-item \
    --table-name MusicTable \
    --key '{"Artist": {"S":"The Beatles"},"SongTitle": {"S":"Hey Jude"}}' \
    --update-expression "SET #rating = :rating" \
    --condition-expression "attribute_exists(Artist)" \
    --expression-attribute-names '{"#rating": "Rating"}' \
    --expression-attribute-values '{":rating": {"N":"5"}}' \
    --region us-east-1
```

清除 MRSC 全域資料表資源。

```
# Remove replica tables (must be done before deleting the primary table)
aws dynamodb update-table \
    --table-name MusicTable \
    --replica-updates '[{"Delete": {"RegionName": "us-east-1"}}]' \
    --global-table-witness-updates '[{"Delete": {"RegionName": "us-west-2"}}]' \
    --region us-east-2

# Wait for replicas to be deleted
echo "Waiting for replicas to be deleted..."
sleep 30

# Delete the primary table
aws dynamodb delete-table \
    --table-name MusicTable \
    --region us-east-2
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - CreateTable
  - DeleteTable
  - DescribeTable
  - GetItem
  - PutItem
  - UpdateItem
  - UpdateTable

管理全域次要索引

下列程式碼範例示範如何管理全域次要索引的完整生命週期。

- 使用全域次要索引建立資料表。
- 將新的 GSI 新增至現有的資料表。
- 更新（增加）GSI 暖輸送量。
- 使用 GSIs查詢資料。
- 刪除 GSI。

AWS CLI 搭配 Bash 指令碼

使用全域次要索引建立資料表。

```
# Create a table with a GSI
aws dynamodb create-table \
    --table-name MusicCollection \
    --attribute-definitions \
        AttributeName=Artist,AttributeType=S \
        AttributeName=SongTitle,AttributeType=S \
        AttributeName=AlbumTitle,AttributeType=S \
    --key-schema \
        AttributeName=Artist,KeyType=HASH \
        AttributeName=SongTitle,KeyType=RANGE \
    --billing-mode PAY_PER_REQUEST \
    --global-secondary-indexes \
        "IndexName=AlbumIndex,\
        KeySchema=[{AttributeName=AlbumTitle,KeyType=HASH}],\
        Projection={ProjectionType=ALL}"
```

將新的（隨需）GSI 新增至現有的資料表。

```
# Add a new GSI to an existing table
aws dynamodb update-table \
    --table-name MusicCollection \
    --attribute-definitions \
        AttributeName=Genre,AttributeType=S \
    --global-secondary-index-updates \
        "[{\"Create\":{\"IndexName\":\"GenreIndex\",\
        \"KeySchema\":[{\"AttributeName\":\"Genre\",\"KeyType\":\"HASH\"}],\
        \"Projection\":{\"ProjectionType\":\"ALL\"}}}]"
```

更新（增加）GSI 暖輸送量。

```
# Increase the warm throughput of a GSI (default values are 12k reads, 4k writes)
aws dynamodb update-table \
    --table-name MusicCollection \
    --global-secondary-index-updates \
        "[{\"Update\":{\"IndexName\":\"AlbumIndex\",\
        \"WarmThroughput\":{\"ReadUnitsPerSecond\":15000,\"WriteUnitsPerSecond
\":6000}}}]"
```

使用 GSIs查詢資料。

```
# Query the AlbumIndex GSI
aws dynamodb query \
    --table-name MusicCollection \
    --index-name AlbumIndex \
    --key-condition-expression "AlbumTitle = :album" \
    --expression-attribute-values '{":album":{"S":"Let It Be"}}'

# Query the GenreIndex GSI
aws dynamodb query \
    --table-name MusicCollection \
    --index-name GenreIndex \
    --key-condition-expression "Genre = :genre" \
    --expression-attribute-values '{":genre":{"S":"Jazz"}}'
```

刪除 GSI。

```
# Delete a GSI from a table
aws dynamodb update-table \
    --table-name MusicCollection \
    --global-secondary-index-updates \
        "[{\"Delete\":{\"IndexName\":\"GenreIndex\"}}]"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - CreateTable
  - DeleteTable
  - 查詢
  - UpdateTable

管理資源型政策

下列程式碼範例示範如何管理 DynamoDB 資料表之資源型政策的完整生命週期。

- 建立具有資源政策的資料表。

- 取得資源政策。
- 更新資源政策。
- 刪除資源政策。

AWS CLI 搭配 Bash 指令碼

建立具有資源政策的資料表。

```
# Step 1: Create a DynamoDB table
aws dynamodb create-table \
    --table-name MusicCollection \
    --attribute-definitions \
        AttributeName=Artist,AttributeType=S \
        AttributeName=SongTitle,AttributeType=S \
    --key-schema \
        AttributeName=Artist,KeyType=HASH \
        AttributeName=SongTitle,KeyType=RANGE \
    --billing-mode PAY_PER_REQUEST

# Step 2: Create a resource-based policy document
cat > policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/DynamoDBReadOnly"
      },
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:BatchGetItem",
        "dynamodb:Query",
        "dynamodb:Scan"
      ],
      "Resource": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection"
    }
  ]
}
EOF

# Step 3: Attach the resource-based policy to the table
```

```
aws dynamodb put-resource-policy \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --policy file://policy.json
```

取得資源政策。

```
# Get the resource-based policy attached to a table
aws dynamodb get-resource-policy \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection
```

更新資源政策。

```
# Step 1: Create an updated policy document
cat > updated-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:role/DynamoDBReadOnly",
          "arn:aws:iam::123456789012:role/DynamoDBAnalytics"
        ]
      },
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:BatchGetItem",
        "dynamodb:Query",
        "dynamodb:Scan"
      ],
      "Resource": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection"
    }
  ]
}
EOF

# Step 2: Update the resource-based policy on the table
aws dynamodb put-resource-policy \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --policy file://updated-policy.json
```

刪除資源政策。

```
# Delete the resource-based policy from a table
aws dynamodb delete-resource-policy \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - CreateTable
  - DeleteResourcePolicy
  - GetResourcePolicy
  - PutResourcePolicy

設定屬性型存取控制

下列程式碼範例示範如何實作 DynamoDB 的屬性型存取控制 (ABAC)。

- 建立 ABAC 的 IAM 政策。
- 為不同部門建立具有標籤的資料表。
- 根據標籤列出和篩選資料表。

AWS CLI 搭配 Bash 指令碼

建立 ABAC 的 IAM 政策。

```
# Step 1: Create a policy document for ABAC
cat > abac-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:BatchGetItem",
        "dynamodb:Query",
        "dynamodb:Scan"
```

```
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Department": "${aws:PrincipalTag/Department}"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:DeleteItem",
        "dynamodb:BatchWriteItem"
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Department": "${aws:PrincipalTag/Department}",
          "aws:ResourceTag/Environment": "Development"
        }
      }
    }
  ]
}
EOF

# Step 2: Create the IAM policy
aws iam create-policy \
    --policy-name DynamoDBDepartmentBasedAccess \
    --policy-document file://abac-policy.json
```

為不同部門建立具有標籤的資料表。

```
# Create a DynamoDB table with tags for ABAC
aws dynamodb create-table \
    --table-name FinanceData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
```

```
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Department,Value=Finance \
        Key=Environment,Value=Development

# Create another table with different tags
aws dynamodb create-table \
    --table-name MarketingData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Department,Value=Marketing \
        Key=Environment,Value=Production
```

根據標籤列出和篩選資料表。

```
# List all DynamoDB tables
echo "Listing all tables:"
aws dynamodb list-tables

# Get ARNs for all tables
echo -e "\nGetting ARNs for all tables:"
TABLE_ARNS=$(aws dynamodb list-tables --query "TableNames[*]" --output text | xargs
 -I {} aws dynamodb describe-table --table-name {} --query "Table.TableArn" --output
 text)

# For each table ARN, list its tags
echo -e "\nListing tags for each table:"
for ARN in $TABLE_ARNS; do
    TABLE_NAME=$(echo $ARN | awk -F/ '{print $2}')
    echo -e "\nTags for table: $TABLE_NAME"
    aws dynamodb list-tags-of-resource --resource-arn $ARN
done

# Example: Find tables with a specific tag
echo -e "\nFinding tables with Environment=Production tag:"
for ARN in $TABLE_ARNS; do
    TABLE_NAME=$(echo $ARN | awk -F/ '{print $2}')
```

```
    TAGS=$(aws dynamodb list-tags-of-resource --resource-arn $ARN --query "Tags[?
Key=='Environment' && Value=='Production']" --output text)
    if [ ! -z "$TAGS" ]; then
        echo "Table with Production tag: $TABLE_NAME"
    fi
done
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - CreatePolicy
  - CreateTable
  - ListTables

使用本機次要索引

下列程式碼範例示範如何使用本機次要索引建立和查詢資料表。

- 使用本機次要索引 (LSI) 建立資料表。
- 使用具有不同投影類型的多個 LSIs建立資料表。
- 使用 LSIs查詢資料。

AWS CLI 搭配 Bash 指令碼

使用本機次要索引建立資料表。

```
# Create a table with a Local Secondary Index
aws dynamodb create-table \
    --table-name CustomerOrders \
    --attribute-definitions \
        AttributeName=CustomerID,AttributeType=S \
        AttributeName=OrderID,AttributeType=S \
        AttributeName=OrderDate,AttributeType=S \
    --key-schema \
        AttributeName=CustomerID,KeyType=HASH \
        AttributeName=OrderID,KeyType=RANGE \
    --local-secondary-indexes \
        "IndexName=OrderDateIndex,\
        KeySchema=[{AttributeName=CustomerID,KeyType=HASH},
{AttributeName=OrderDate,KeyType=RANGE}],\
        Projection={ProjectionType=ALL}" \
```

```
    --billing-mode PAY_PER_REQUEST
```

建立具有多個 LSIs資料表。

```
# Create a table with multiple Local Secondary Indexes
aws dynamodb create-table \
    --table-name CustomerDetails \
    --attribute-definitions \
        AttributeName=CustomerID,AttributeType=S \
        AttributeName=Name,AttributeType=S \
        AttributeName=Email,AttributeType=S \
        AttributeName=RegistrationDate,AttributeType=S \
    --key-schema \
        AttributeName=CustomerID,KeyType=HASH \
        AttributeName=Name,KeyType=RANGE \
    --local-secondary-indexes \
        "[
            {
                \"IndexName\": \"EmailIndex\",
                \"KeySchema\": [
                    {\"AttributeName\":\"CustomerID\",\"KeyType\":\"HASH\"},
                    {\"AttributeName\":\"Email\",\"KeyType\":\"RANGE\"}
                ],
                \"Projection\": {\"ProjectionType\":\"INCLUDE\",\"NonKeyAttributes
\":[\"Address\",\"Phone\"]}
            },
            {
                \"IndexName\": \"RegistrationIndex\",
                \"KeySchema\": [
                    {\"AttributeName\":\"CustomerID\",\"KeyType\":\"HASH\"},
                    {\"AttributeName\":\"RegistrationDate\",\"KeyType\":\"RANGE\"}
                ],
                \"Projection\": {\"ProjectionType\":\"KEYS_ONLY\"}
            }
        ]" \
    --billing-mode PAY_PER_REQUEST
```

使用 LSIs查詢資料。

```
# Query the OrderDateIndex LSI
aws dynamodb query \
```

```
    --table-name CustomerOrders \
    --index-name OrderDateIndex \
    --key-condition-expression "CustomerID = :custId AND OrderDate BETWEEN :date1
 AND :date2" \
    --expression-attribute-values '{
        ":custId": {"S": "C1"},
        ":date1": {"S": "2023-01-01"},
        ":date2": {"S": "2023-02-01"}
    }'

# Query with a filter expression
aws dynamodb query \
    --table-name CustomerOrders \
    --index-name OrderDateIndex \
    --key-condition-expression "CustomerID = :custId" \
    --filter-expression "Amount > :amount" \
    --expression-attribute-values '{
        ":custId": {"S": "C1"},
        ":amount": {"N": "150"}
    }'
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - CreateTable
  - 查詢

使用串流和Time-to-Live

下列程式碼範例示範如何管理 DynamoDB 串流和Time-to-Live功能。

- 建立已啟用串流的資料表。
- 描述串流。
- 建立 Lambda 函數以處理串流。
- 在資料表上啟用 TTL。
- 新增具有 TTL 屬性的項目。
- 描述 TTL 設定。

AWS CLI 搭配 Bash 指令碼

建立已啟用串流的資料表。

```
# Create a table with DynamoDB Streams enabled
aws dynamodb create-table \
    --table-name StreamsDemo \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --stream-specification StreamEnabled=true,StreamViewType=NEW_AND_OLD_IMAGES
```

描述串流。

```
# Get information about the stream
aws dynamodb describe-table \
    --table-name StreamsDemo \
    --query "Table.StreamSpecification"

# Get the stream ARN
STREAM_ARN=$(aws dynamodb describe-table \
    --table-name StreamsDemo \
    --query "Table.LatestStreamArn" \
    --output text)

echo "Stream ARN: $STREAM_ARN"

# Describe the stream
aws dynamodbstreams describe-stream \
    --stream-arn $STREAM_ARN
```

為 Streams 建立 Lambda 函數。

```
# Step 1: Create an IAM role for the Lambda function
cat > trust-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

aws iam create-role \
    --role-name DynamoDBStreamsLambdaRole \
    --assume-role-policy-document file://trust-policy.json

# Step 2: Attach permissions to the role
aws iam attach-role-policy \
    --role-name DynamoDBStreamsLambdaRole \
    --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaDynamoDBExecutionRole

# Step 3: Create a Lambda function (code would be in a separate file)
echo "Lambda function creation would be done separately with appropriate code"

# Step 4: Create an event source mapping
echo "Example command to create event source mapping:"
echo "aws lambda create-event-source-mapping \\"
echo "    --function-name ProcessDynamoDBRecords \\"
echo "    --event-source $STREAM_ARN \\"
echo "    --batch-size 100 \\"
echo "    --starting-position LATEST"
```

在資料表上啟用 TTL。

```
# Create a table for TTL demonstration
aws dynamodb create-table \
    --table-name TTLDemo \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST
```

```
# Wait for table to become active
aws dynamodb wait table-exists --table-name TTLDemo

# Enable TTL on the table
aws dynamodb update-time-to-live \
    --table-name TTLDemo \
    --time-to-live-specification "Enabled=true, AttributeName=ExpirationTime"
```

新增具有 TTL 屬性的項目。

```
# Calculate expiration time (current time + 1 day in seconds)
EXPIRATION_TIME=$(date -d "+1 day" +%s)

# Add an item with TTL attribute
aws dynamodb put-item \
    --table-name TTLDemo \
    --item '{
        "ID": {"S": "item1"},
        "Data": {"S": "This item will expire in 1 day"},
        "ExpirationTime": {"N": "'$EXPIRATION_TIME'"}
    }'

# Add an item that expires in 1 hour
EXPIRATION_TIME_HOUR=$(date -d "+1 hour" +%s)
aws dynamodb put-item \
    --table-name TTLDemo \
    --item '{
        "ID": {"S": "item2"},
        "Data": {"S": "This item will expire in 1 hour"},
        "ExpirationTime": {"N": "'$EXPIRATION_TIME_HOUR'"}
    }'
```

描述 TTL 設定。

```
# Describe TTL settings for a table
aws dynamodb describe-time-to-live \
    --table-name TTLDemo
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。

- AttachRolePolicy
- CreateRole
- CreateTable
- DescribeTable
- DescribeTimeToLive
- PutItem
- UpdateTimeToLive

使用全域資料表和多區域複寫最終一致性 (MREC)

下列程式碼範例示範如何使用具有最終一致性 (MREC) 的多區域複寫來管理 DynamoDB 全域資料表。

- 建立具有多區域複寫 (MREC) 的資料表。
- 放置並從複本資料表取得項目。
- one-by-one移除複本。
- 刪除資料表以清除 。

AWS CLI 搭配 Bash 指令碼

建立具有多區域複寫的資料表。

```
# Step 1: Create a new table (MusicTable) in US East (Ohio), with DynamoDB Streams
 enabled (NEW_AND_OLD_IMAGES)
aws dynamodb create-table \
    --table-name MusicTable \
    --attribute-definitions \
        AttributeName=Artist,AttributeType=S \
        AttributeName=SongTitle,AttributeType=S \
    --key-schema \
        AttributeName=Artist,KeyType=HASH \
        AttributeName=SongTitle,KeyType=RANGE \
    --billing-mode PAY_PER_REQUEST \
    --stream-specification StreamEnabled=true,StreamViewType=NEW_AND_OLD_IMAGES \
    --region us-east-2

# Step 2: Create an identical MusicTable table in US East (N. Virginia)
aws dynamodb update-table --table-name MusicTable --cli-input-json \
'{
```

```
  "ReplicaUpdates":
  [
    {
      "Create": {
        "RegionName": "us-east-1"
      }
    }
  ]
}' \
--region us-east-2

# Step 3: Create a table in Europe (Ireland)
aws dynamodb update-table --table-name MusicTable --cli-input-json \
'{
  "ReplicaUpdates":
  [
    {
      "Create": {
        "RegionName": "eu-west-1"
      }
    }
  ]
}' \
--region us-east-2
```

描述多區域資料表。

```
# Step 4: View the list of replicas created using describe-table
aws dynamodb describe-table \
    --table-name MusicTable \
    --region us-east-2 \
    --query 'Table.
{TableName:TableName,TableStatus:TableStatus,MultiRegionConsistency:MultiRegionConsistency,R
{Region:RegionName,Status:ReplicaStatus}}'
```

將項目放在複本資料表中。

```
# Step 5: To verify that replication is working, add a new item to the Music table
 in US East (Ohio)
aws dynamodb put-item \
    --table-name MusicTable \
```

```
    --item '{"Artist": {"S":"item_1"},"SongTitle": {"S":"Song Value 1"}}' \
    --region us-east-2
```

從複本資料表取得項目。

```
# Step 6: Wait for a few seconds, and then check to see whether the item has been
# successfully replicated to US East (N. Virginia) and Europe (Ireland)
aws dynamodb get-item \
    --table-name MusicTable \
    --key '{"Artist": {"S":"item_1"},"SongTitle": {"S":"Song Value 1"}}' \
    --region us-east-1

aws dynamodb get-item \
    --table-name MusicTable \
    --key '{"Artist": {"S":"item_1"},"SongTitle": {"S":"Song Value 1"}}' \
    --region eu-west-1
```

移除複本。

```
# Step 7: Delete the replica table in Europe (Ireland) Region
aws dynamodb update-table --table-name MusicTable --cli-input-json \
'{
  "ReplicaUpdates":
  [
    {
      "Delete": {
        "RegionName": "eu-west-1"
      }
    }
  ]
}' \
--region us-east-2

# Delete the replica table in US East (N. Virginia) Region
aws dynamodb update-table --table-name MusicTable --cli-input-json \
'{
  "ReplicaUpdates":
  [
    {
      "Delete": {
        "RegionName": "us-east-1"
```

```
      }
    }
  ]
}' \
--region us-east-2
```

刪除資料表以清除 。

```
# Clean up: Delete the primary table
aws dynamodb delete-table --table-name MusicTable --region us-east-2

echo "Global table demonstration complete."
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - CreateTable
  - DeleteTable
  - DescribeTable
  - GetItem
  - PutItem
  - UpdateTable

使用資源標記

下列程式碼範例示範如何管理 DynamoDB 資源的標籤。

- 建立具有標籤的資料表。
- 列出資源的標籤。
- 將標籤新增至資源。
- 從資源移除標籤。
- 依標籤篩選資料表。

AWS CLI 搭配 Bash 指令碼

建立具有標籤的資料表。

```
# Create a table with tags
aws dynamodb create-table \
    --table-name TaggedTable \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Environment,Value=Production \
        Key=Project,Value=Analytics \
        Key=Owner,Value=DataTeam
```

列出資源的標籤。

```
# Get the table ARN
TABLE_ARN=$(aws dynamodb describe-table \
    --table-name TaggedTable \
    --query "Table.TableArn" \
    --output text)

# List tags for the table
aws dynamodb list-tags-of-resource \
    --resource-arn $TABLE_ARN
```

將標籤新增至資源。

```
# Add tags to an existing table
aws dynamodb tag-resource \
    --resource-arn $TABLE_ARN \
    --tags \
        Key=CostCenter,Value=12345 \
        Key=BackupSchedule,Value=Daily
```

從資源移除標籤。

```
# Remove tags from a table
aws dynamodb untag-resource \
    --resource-arn $TABLE_ARN \
```

```
    --tag-keys Owner BackupSchedule
```

依標籤篩選資料表。

```
# Create another table with different tags
aws dynamodb create-table \
    --table-name AnotherTaggedTable \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Environment,Value=Development \
        Key=Project,Value=Testing

# Wait for table to become active
aws dynamodb wait table-exists --table-name AnotherTaggedTable

# List all tables
echo "All tables:"
aws dynamodb list-tables

# Get ARNs for all tables
echo -e "\nFiltering tables by Environment=Production tag:"
TABLE_ARNS=$(aws dynamodb list-tables --query "TableNames[*]" --output text | xargs
 -I {} aws dynamodb describe-table --table-name {} --query "Table.TableArn" --output
 text)

# Find tables with specific tag
for ARN in $TABLE_ARNS; do
    TABLE_NAME=$(echo $ARN | awk -F/ '{print $2}')
    TAGS=$(aws dynamodb list-tags-of-resource --resource-arn $ARN --query "Tags[?
Key=='Environment' && Value=='Production']" --output text)
    if [ ! -z "$TAGS" ]; then
        echo "Table with Production tag: $TABLE_NAME"
    fi
done
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - CreateTable

- ListTagsOfResource
- TagResource
- UntagResource

使用資料表加密

下列程式碼範例示範如何管理 DynamoDB 資料表的加密選項。

- 建立具有預設加密的資料表。
- 使用客戶受管 CMK 建立資料表。
- 更新資料表加密設定。
- 描述資料表加密。

AWS CLI 搭配 Bash 指令碼

建立具有預設加密的資料表。

```
# Create a table with default encryption (AWS owned key)
aws dynamodb create-table \
    --table-name CustomerData \
    --attribute-definitions \
        AttributeName=CustomerID,AttributeType=S \
    --key-schema \
        AttributeName=CustomerID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --sse-specification Enabled=true,SSEType=KMS
```

使用客戶受管 CMK 建立資料表。

```
# Step 1: Create a customer managed key in KMS
aws kms create-key \
    --description "Key for DynamoDB table encryption" \
    --key-usage ENCRYPT_DECRYPT \
    --customer-master-key-spec SYMMETRIC_DEFAULT

# Store the key ID for later use
KEY_ID=$(aws kms list-keys --query "Keys[?contains(KeyArn, 'Key for
 DynamoDB')].KeyId" --output text)
```

```
# Step 2: Create a table with the customer managed key
aws dynamodb create-table \
    --table-name SensitiveData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

更新資料表加密。

```
# Update a table to use a different KMS key
aws dynamodb update-table \
    --table-name CustomerData \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

描述資料表加密。

```
# Describe the table to see encryption settings
aws dynamodb describe-table \
    --table-name CustomerData \
    --query "Table.SSEDescription"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - CreateKey
  - CreateTable
  - DescribeTable
  - UpdateTable

搭配 Bash 指令碼使用 AWS CLI 的 Amazon EC2 範例

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Bash 指令碼搭配 Amazon EC2 來執行動作和實作常見案例。

基本概念是程式碼範例，這些範例說明如何在服務內執行基本操作。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 基本概念
- 動作

基本概念

了解基本概念

以下程式碼範例顯示做法：

- 建立金鑰對和安全群組。
- 選取 Amazon Machine Image (AMI) 和相容的執行個體類型，然後建立執行個體。
- 停止並重新啟動執行個體。
- 將彈性 IP 地址與您的執行個體建立關聯.
- 使用 SSH 連線至執行個體，然後清理資源。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

在命令提示中執行互動式案例。

```
###############################################################################
# function get_started_with_ec2_instances
#
# Runs an interactive scenario that shows how to get started using EC2 instances.
#
#     "EC2 access" permissions are needed to run this code.
#
# Returns:
```

```
#       0 - If successful.
#       1 - If an error occurred.
################################################################################
function get_started_with_ec2_instances() {
  # Requires version 4 for mapfile.
  local required_version=4.0

  # Get the current Bash version
  # Check if BASH_VERSION is set
  local current_version
  if [[ -n "$BASH_VERSION" ]]; then
    # Convert BASH_VERSION to a number for comparison
    current_version=$BASH_VERSION
  else
    # Get the current Bash version using the bash command
    current_version=$(bash --version | head -n 1 | awk '{ print $4 }')
  fi

  # Convert version strings to numbers for comparison
  local required_version_num current_version_num
  required_version_num=$(echo "$required_version" | awk -F. '{ print ($1 * 10000) +
 ($2 * 100) + $3 }')
  current_version_num=$(echo "$current_version" | awk -F. '{ print ($1 * 10000) +
 ($2 * 100) + $3 }')

  # Compare versions
  if ((current_version_num < required_version_num)); then
    echo "Error: This script requires Bash version $required_version or higher."
    echo "Your current Bash version is number is $current_version."
    exit 1
  fi

  {
    if [ "$EC2_OPERATIONS_SOURCED" != "True" ]; then

      source ./ec2_operations.sh
    fi
  }

  echo_repeat "*" 88
  echo "Welcome to the Amazon Elastic Compute Cloud (Amazon EC2) get started with
 instances demo."
  echo_repeat "*" 88
  echo
```

```
echo "Let's create an RSA key pair that you can be use to securely connect to "
echo "your EC2 instance."

echo -n "Enter a unique name for your key: "
get_input
local key_name
key_name=$get_input_result

local temp_dir
temp_dir=$(mktemp -d)
local key_file_name="$temp_dir/${key_name}.pem"

if ec2_create_keypair -n "${key_name}" -f "${key_file_name}"; then
  echo "Created a key pair $key_name and saved the private key to $key_file_name"
  echo
else
  errecho "The key pair failed to create. This demo will exit."
  return 1
fi

chmod 400 "${key_file_name}"

if yes_no_input "Do you want to list some of your key pairs? (y/n) "; then
  local keys_and_fingerprints
  keys_and_fingerprints="$(ec2_describe_key_pairs)" && {
    local image_name_and_id
    while IFS=$'\n' read -r image_name_and_id; do
      local entries
      IFS=$'\t' read -ra entries <<<"$image_name_and_id"
      echo "Found rsa key ${entries[0]} with fingerprint:"
      echo "     ${entries[1]}"
    done <<<"$keys_and_fingerprints"

  }
fi

echo_repeat "*" 88
echo_repeat "*" 88

echo "Let's create a security group to manage access to your instance."
echo -n "Enter a unique name for your security group: "
get_input
local security_group_name
```

```
  security_group_name=$get_input_result
  local security_group_id
  security_group_id=$(ec2_create_security_group -n "$security_group_name" \
    -d "Security group for EC2 instance") || {
    errecho "The security failed to create. This demo will exit."
    clean_up "$key_name" "$key_file_name"
    return 1
  }

  echo "Security group created with ID $security_group_id"
  echo

  local public_ip
  public_ip=$(curl -s http://checkip.amazonaws.com)

  echo "Let's add a rule to allow SSH only from your current IP address."
  echo "Your public IP address is $public_ip"
  echo -n "press return to add this rule to your security group."
  get_input

  if ! ec2_authorize_security_group_ingress -g "$security_group_id" -i "$public_ip"
-p tcp -f 22 -t 22; then
    errecho "The security group rules failed to update. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
  fi

  echo "Security group rules updated"

  local security_group_description
  security_group_description="$(ec2_describe_security_groups -g
"${security_group_id}")" || {
    errecho "Failed to describe security groups. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
  }

  mapfile -t parameters <<<"$security_group_description"
  IFS=$'\t' read -ra entries <<<"${parameters[0]}"
  echo "Security group: ${entries[0]}"
  echo "    ID: ${entries[1]}"
  echo "    VPC: ${entries[2]}"
  echo "Inbound permissions:"
  IFS=$'\t' read -ra entries <<<"${parameters[1]}"
```

```
  echo "    IpProtocol: ${entries[0]}"
  echo "    FromPort: ${entries[1]}"
  echo "    ToPort: ${entries[2]}"
  echo "    CidrIp: ${parameters[2]}"

  local parameters
  parameters="$(ssm_get_parameters_by_path -p "/aws/service/ami-amazon-linux-
latest")" || {
    errecho "Failed to get parameters. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1

  }

  local image_ids=""
  mapfile -t parameters <<<"$parameters"
  for image_name_and_id in "${parameters[@]}"; do
    IFS=$'\t' read -ra values <<<"$image_name_and_id"
    if [[ "${values[0]}" == *"amzn2"* ]]; then
      image_ids+="${values[1]} "
    fi
  done

  local images
  images="$(ec2_describe_images -i "$image_ids")" || {
    errecho "Failed to describe images. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1

  }

  new_line_and_tab_to_list "$images"
  local images=("${list_result[@]}")

  # Get the size of the array
  local images_count=${#images[@]}

  if ((images_count == 0)); then
    errecho "No images found. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
  fi

  echo_repeat "*" 88
```

```
  echo_repeat "*" 88

  echo "Let's create an instance from an Amazon Linux 2 AMI. Here are some options:"
  for ((i = 0; i < images_count; i += 3)); do
    echo "$(((i / 3) + 1)) - ${images[$i]}"
  done

  integer_input "Please enter the number of the AMI you want to use: " 1
 "$((images_count / 3))"
  local choice=$get_input_result
  choice=$(((choice - 1) * 3))

  echo "Great choice."
  echo

  local architecture=${images[$((choice + 1))]}
  local image_id=${images[$((choice + 2))]}
  echo "Here are some instance types that support the ${architecture} architecture
 of the image:"
  response="$(ec2_describe_instance_types -a "${architecture}" -t
 "*.micro,*.small")" || {
    errecho "Failed to describe instance types. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
  }

  local instance_types
  mapfile -t instance_types <<<"$response"

  # Get the size of the array
  local instance_types_count=${#instance_types[@]}

  echo "Here are some options:"
  for ((i = 0; i < instance_types_count; i++)); do
    echo "$((i + 1)) - ${instance_types[$i]}"
  done

  integer_input "Which one do you want to use? " 1 "${#instance_types[@]}
"
  choice=$get_input_result
  local instance_type=${instance_types[$((choice - 1))]}
  echo "Another great choice."
  echo
```

```
 echo "Creating your instance and waiting for it to start..."
 local instance_id
 instance_id=$(ec2_run_instances -i "$image_id" -t "$instance_type" -k "$key_name"
-s "$security_group_id") || {
   errecho "Failed to run instance. This demo will exit."
   clean_up "$key_name" "$key_file_name" "$security_group_id"
   return 1
 }

 ec2_wait_for_instance_running -i "$instance_id"
 echo "Your instance is ready:"
 echo

 local instance_details
 instance_details="$(ec2_describe_instances -i "${instance_id}")"

 echo
 print_instance_details "${instance_details}"

 local public_ip
 public_ip=$(echo "${instance_details}" | awk '{print $6}')
 echo
 echo "You can use SSH to connect to your instance"
 echo "If the connection attempt times out, you might have to manually update the
SSH ingress rule"
 echo "for your IP address in the AWS Management Console."
 connect_to_instance "$key_file_name" "$public_ip"

 echo -n "Press Enter when you're ready to continue the demo: "
 get_input

 echo_repeat "*" 88
 echo_repeat "*" 88

 echo "Let's stop and start your instance to see what changes."
 echo "Stopping your instance and waiting until it's stopped..."
 ec2_stop_instances -i "$instance_id"
 ec2_wait_for_instance_stopped -i "$instance_id"

 echo "Your instance is stopped. Restarting..."

 ec2_start_instances -i "$instance_id"
 ec2_wait_for_instance_running -i "$instance_id"
```

```
  echo "Your instance is running again."
  local instance_details
  instance_details="$(ec2_describe_instances -i "${instance_id}")"

  print_instance_details "${instance_details}"

  public_ip=$(echo "${instance_details}" | awk '{print $6}')

  echo "Every time your instance is restarted, its public IP address changes"
  connect_to_instance "$key_file_name" "$public_ip"

  echo -n "Press Enter when you're ready to continue the demo: "
  get_input

  echo_repeat "*" 88
  echo_repeat "*" 88

  echo "You can allocate an Elastic IP address and associate it with your instance"
  echo "to keep a consistent IP address even when your instance restarts."

  local result
  result=$(ec2_allocate_address -d vpc) || {
    errecho "Failed to allocate an address. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id"
    return 1
  }

  local elastic_ip allocation_id
  elastic_ip=$(echo "$result" | awk '{print $1}')
  allocation_id=$(echo "$result" | awk '{print $2}')

  echo "Allocated static Elastic IP address: $elastic_ip"

  local association_id
  association_id=$(ec2_associate_address -i "$instance_id" -a "$allocation_id") || {
    errecho "Failed to associate an address. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id"
"$allocation_id"
    return 1
  }

  echo "Associated your Elastic IP with your instance."
  echo "You can now use SSH to connect to your instance by using the Elastic IP."
  connect_to_instance "$key_file_name" "$elastic_ip"
```

```
  echo -n "Press Enter when you're ready to continue the demo: "
  get_input

  echo_repeat "*" 88
  echo_repeat "*" 88

  echo "Let's stop and start your instance to see what changes."
  echo "Stopping your instance and waiting until it's stopped..."
  ec2_stop_instances -i "$instance_id"
  ec2_wait_for_instance_stopped -i "$instance_id"

  echo "Your instance is stopped. Restarting..."

  ec2_start_instances -i "$instance_id"
  ec2_wait_for_instance_running -i "$instance_id"

  echo "Your instance is running again."
  local instance_details
  instance_details="$(ec2_describe_instances -i "${instance_id}")"

  print_instance_details "${instance_details}"

  echo "Because you have associated an Elastic IP with your instance, you can"
  echo "connect by using a consistent IP address after the instance restarts."
  connect_to_instance "$key_file_name" "$elastic_ip"

  echo -n "Press Enter when you're ready to continue the demo: "
  get_input

  echo_repeat "*" 88
  echo_repeat "*" 88

  if yes_no_input "Do you want to delete the resources created in this demo: (y/n)
"; then
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id" \
      "$allocation_id" "$association_id"
  else
    echo "The following resources were not deleted."
    echo "Key pair: $key_name"
    echo "Key file: $key_file_name"
    echo "Security group: $security_group_id"
    echo "Instance: $instance_id"
    echo "Elastic IP address: $elastic_ip"
```

```
  fi
}

###############################################################################
# function clean_up
#
# This function cleans up the created resources.
#     $1 - The name of the ec2 key pair to delete.
#     $2 - The name of the key file to delete.
#     $3 - The ID of the security group to delete.
#     $4 - The ID of the instance to terminate.
#     $5 - The ID of the elastic IP address to release.
#     $6 - The ID of the elastic IP address to disassociate.
#
# Returns:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function clean_up() {
  local result=0
  local key_pair_name=$1
  local key_file_name=$2
  local security_group_id=$3
  local instance_id=$4
  local allocation_id=$5
  local association_id=$6

  if [ -n "$association_id" ]; then
    # bashsupport disable=BP2002
    if (ec2_disassociate_address -a "$association_id"); then
      echo "Disassociated elastic IP address with ID $association_id"
    else
      errecho "The elastic IP address disassociation failed."
      result=1
    fi
  fi

  if [ -n "$allocation_id" ]; then
    # bashsupport disable=BP2002
    if (ec2_release_address -a "$allocation_id"); then
      echo "Released elastic IP address with ID $allocation_id"
    else
      errecho "The elastic IP address release failed."
      result=1
```

```
    fi
  fi

  if [ -n "$instance_id" ]; then
    # bashsupport disable=BP2002
    if (ec2_terminate_instances -i "$instance_id"); then
      echo "Started terminating instance with ID $instance_id"

      ec2_wait_for_instance_terminated -i "$instance_id"
    else
      errecho "The instance terminate failed."
      result=1
    fi
  fi

  if [ -n "$security_group_id" ]; then
    # bashsupport disable=BP2002
    if (ec2_delete_security_group -i "$security_group_id"); then
      echo "Deleted security group with ID $security_group_id"
    else
      errecho "The security group delete failed."
      result=1
    fi
  fi

  if [ -n "$key_pair_name" ]; then
    # bashsupport disable=BP2002
    if (ec2_delete_keypair -n "$key_pair_name"); then
      echo "Deleted key pair named $key_pair_name"
    else
      errecho "The key pair delete failed."
      result=1
    fi
  fi

  if [ -n "$key_file_name" ]; then
    rm -f "$key_file_name"
  fi

  return $result
}

################################################################################
# function ssm_get_parameters_by_path
```

```
#
# This function retrieves one or more parameters from the AWS Systems Manager
 Parameter Store
# by specifying a parameter path.
#
# Parameters:
#       -p parameter_path - The path of the parameter(s) to retrieve.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ssm_get_parameters_by_path() {
  local parameter_path response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ssm_get_parameters_by_path"
    echo "Retrieves one or more parameters from the AWS Systems Manager Parameter
 Store by specifying a parameter path."
    echo "  -p parameter_path - The path of the parameter(s) to retrieve."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "p:h" option; do
    case "${option}" in
      p) parameter_path="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$parameter_path" ]]; then
    errecho "ERROR: You must provide a parameter path with the -p parameter."
```

```
    usage
    return 1
  fi

  response=$(aws ssm get-parameters-by-path \
    --path "$parameter_path" \
    --query "Parameters[*].[Name, Value]" \
    --output text) || {
    aws_cli_error_log $?
    errecho "ERROR: AWS reports get-parameters-by-path operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}

###############################################################################
# function print_instance_details
#
# This function prints the details of an Amazon Elastic Compute Cloud (Amazon EC2)
 instance.
#
# Parameters:
#       instance_details - The instance details in the format "InstanceId ImageId
 InstanceType KeyName VpcId PublicIpAddress State.Name".
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function print_instance_details() {
  local instance_details="$1"

  if [[ -z "${instance_details}" ]]; then
    echo "Error: Missing required instance details argument."
    return 1
  fi

  local instance_id image_id instance_type key_name vpc_id public_ip state
  instance_id=$(echo "${instance_details}" | awk '{print $1}')
  image_id=$(echo "${instance_details}" | awk '{print $2}')
  instance_type=$(echo "${instance_details}" | awk '{print $3}')
```

```
  key_name=$(echo "${instance_details}" | awk '{print $4}')
  vpc_id=$(echo "${instance_details}" | awk '{print $5}')
  public_ip=$(echo "${instance_details}" | awk '{print $6}')
  state=$(echo "${instance_details}" | awk '{print $7}')

  echo "    ID: ${instance_id}"
  echo "    Image ID: ${image_id}"
  echo "    Instance type: ${instance_type}"
  echo "    Key name: ${key_name}"
  echo "    VPC ID: ${vpc_id}"
  echo "    Public IP: ${public_ip}"
  echo "    State: ${state}"

  return 0
}

###############################################################################
# function connect_to_instance
#
# This function displays the public IP address of an Amazon Elastic Compute Cloud
 (Amazon EC2) instance and prompts the user to connect to the instance via SSH.
#
# Parameters:
#       $1 - The name of the key file used to connect to the instance.
#       $2 - The public IP address of the instance.
#
# Returns:
#       None
###############################################################################
function connect_to_instance() {
  local key_file_name="$1"
  local public_ip="$2"

  # Validate the input parameters
  if [[ -z "$key_file_name" ]]; then
    echo "ERROR: You must provide a key file name as the first argument." >&2
    return 1
  fi

  if [[ -z "$public_ip" ]]; then
    echo "ERROR: You must provide a public IP address as the second argument." >&2
    return 1
  fi
```

```
  # Display the public IP address and connection command
  echo "To connect, run the following command:"
  echo "    ssh -i ${key_file_name} ec2-user@${public_ip}"

  # Prompt the user to connect to the instance
  if yes_no_input "Do you want to connect now? (y/n) "; then
    echo "After you have connected, you can return to this example by typing 'exit'"
    ssh -i "${key_file_name}" ec2-user@"${public_ip}"
  fi
}

###############################################################################
# function get_input
#
# This function gets user input from the command line.
#
# Outputs:
#   User input to stdout.
#
# Returns:
#       0
###############################################################################
function get_input() {

  if [ -z "${mock_input+x}" ]; then
    read -r get_input_result
  else

    if [ "$mock_input_array_index" -lt ${#mock_input_array[@]} ]; then
      get_input_result="${mock_input_array[$mock_input_array_index]}"
      # bashsupport disable=BP2001
      # shellcheck disable=SC2206
      ((mock_input_array_index++))
      echo -n "$get_input_result"
    else
      echo "MOCK_INPUT_ARRAY has no more elements" 1>&2
      return 1
    fi
  fi

  return 0
}

###############################################################################
```

```
# function yes_no_input
#
# This function requests a yes/no answer from the user, following to a prompt.
#
# Parameters:
#       $1 - The prompt.
#
# Returns:
#       0 - If yes.
#       1 - If no.
###############################################################################
function yes_no_input() {
  if [ -z "$1" ]; then
    echo "Internal error yes_no_input"
    return 1
  fi

  local index=0
  local response="N"
  while [[ $index -lt 10 ]]; do
    index=$((index + 1))
    echo -n "$1"
    if ! get_input; then
      return 1
    fi
    response=$(echo "$get_input_result" | tr '[:upper:]' '[:lower:]')
    if [ "$response" = "y" ] || [ "$response" = "n" ]; then
      break
    else
      echo -e "\nPlease enter or 'y' or 'n'."
    fi
  done

  echo

  if [ "$response" = "y" ]; then
    return 0
  else
    return 1
  fi
}

###############################################################################
# function integer_input
```

```
#
# This function prompts the user to enter an integer within a specified range
# and validates the input.
#
# Parameters:
#       $1 - The prompt message to display to the user.
#       $2 - The minimum value of the accepted range.
#       $3 - The maximum value of the accepted range.
#
# Returns:
#       The valid integer input from the user.
#       If the input is invalid or out of range, the function will continue
#       prompting the user until a valid input is provided.
###############################################################################
function integer_input() {
  local prompt="$1"
  local min_value="$2"
  local max_value="$3"
  local input=""

  while true; do
    # Display the prompt message and wait for user input
    echo -n "$prompt"

    if ! get_input; then
      return 1
    fi

    input="$get_input_result"

    # Check if the input is a valid integer
    if [[ "$input" =~ ^-?[0-9]+$ ]]; then
      # Check if the input is within the specified range
      if ((input >= min_value && input <= max_value)); then
        return 0
      else
        echo "Error: Input, $input, must be between $min_value and $max_value."
      fi
    else
      echo "Error: Invalid input- $input. Please enter an integer."
    fi
  done
}
###############################################################################
```

```
# function new_line_and_tab_to_list
#
# This function takes a string input containing newlines and tabs, and
# converts it into a list (array) of elements.
#
# Parameters:
#       $1 - The input string containing newlines and tabs.
#
# Returns:
#       The resulting list (array) is stored in the global variable
#       'list_result'.
##############################################################################
function new_line_and_tab_to_list() {
  local input=$1
  export list_result

  list_result=()
  mapfile -t lines <<<"$input"
  local line
  for line in "${lines[@]}"; do
    IFS=$'\t' read -ra parameters <<<"$line"
    list_result+=("${parameters[@]}")
  done
}

##############################################################################
# function echo_repeat
#
# This function prints a string 'n' times to stdout.
#
# Parameters:
#       $1 - The string.
#       $2 - Number of times to print the string.
#
# Outputs:
#   String 'n' times to stdout.
#
# Returns:
#       0
##############################################################################
function echo_repeat() {
  local end=$2
  for ((i = 0; i < end; i++)); do
    echo -n "$1"
```

```
  done
  echo
}
```

此案例中使用的 DynamoDB 函數。

```
###############################################################################
# function ec2_create_keypair
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or
 2048-bit RSA key pair
# and writes it to a file.
#
# Parameters:
#       -n key_pair_name - A key pair name.
#       -f file_path - File to store the key pair.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_create_keypair() {
  local key_pair_name file_path response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_create_keypair"
    echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or 2048-bit
 RSA key pair"
    echo " and writes it to a file."
    echo "  -n key_pair_name - A key pair name."
    echo "  -f file_path - File to store the key pair."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:f:h" option; do
    case "${option}" in
      n) key_pair_name="${OPTARG}" ;;
      f) file_path="${OPTARG}" ;;
      h)
```

```
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$file_path" ]]; then
    errecho "ERROR: You must provide a file path with the -f parameter."
    usage
    return 1
  fi

  response=$(aws ec2 create-key-pair \
    --key-name "$key_pair_name" \
    --query 'KeyMaterial' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
  }

  if [[ -n "$file_path" ]]; then
    echo "$response" >"$file_path"
  fi

  return 0
}

################################################################################
# function ec2_describe_key_pairs
#
```

```
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
 pairs.
#
# Parameters:
#       -h - Display help.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_describe_key_pairs() {
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_key_pairs"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
 pairs."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "h" option; do
    case "${option}" in
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local response

  response=$(aws ec2 describe-key-pairs \
    --query 'KeyPairs[*].[KeyName, KeyFingerprint]' \
    --output text) || {
    aws_cli_error_log ${?}
```

```
    errecho "ERROR: AWS reports describe-key-pairs operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}

################################################################################
# function ec2_create_security_group
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Parameters:
#       -n security_group_name - The name of the security group.
#       -d security_group_description - The description of the security group.
#
# Returns:
#       The ID of the created security group, or an error message if the operation
 fails.
# And:
#       0 - If successful.
#       1 - If it fails.
#
################################################################################
function ec2_create_security_group() {
  local security_group_name security_group_description response

  # Function to display usage information
  function usage() {
    echo "function ec2_create_security_group"
    echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group."
    echo "  -n security_group_name - The name of the security group."
    echo "  -d security_group_description - The description of the security group."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "n:d:h" option; do
    case "${option}" in
      n) security_group_name="${OPTARG}" ;;
      d) security_group_description="${OPTARG}" ;;
      h)
```

```
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$security_group_name" ]]; then
    errecho "ERROR: You must provide a security group name with the -n parameter."
    return 1
  fi

  if [[ -z "$security_group_description" ]]; then
    errecho "ERROR: You must provide a security group description with the -d
 parameter."
    return 1
  fi

  # Create the security group
  response=$(aws ec2 create-security-group \
    --group-name "$security_group_name" \
    --description "$security_group_description" \
    --query "GroupId" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports create-security-group operation failed."
    errecho "$response"
    return 1
  }

  echo "$response"
  return 0
}

################################################################################
# function ec2_describe_security_groups
#
```

```
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
 security groups.
#
# Parameters:
#       -g security_group_id - The ID of the security group to describe (optional).
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_describe_security_groups() {
  local security_group_id response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_security_groups"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) security
 groups."
    echo "  -g security_group_id - The ID of the security group to describe
 (optional)."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "g:h" option; do
    case "${option}" in
      g) security_group_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local query="SecurityGroups[*].[GroupName, GroupId, VpcId, IpPermissions[*].
[IpProtocol, FromPort, ToPort, IpRanges[*].CidrIp]]"
```

```
  if [[ -n "$security_group_id" ]]; then
    response=$(aws ec2 describe-security-groups --group-ids "$security_group_id" --
query "${query}" --output text)
  else
    response=$(aws ec2 describe-security-groups --query "${query}" --output text)
  fi

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports describe-security-groups operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function ec2_authorize_security_group_ingress
#
# This function authorizes an ingress rule for an Amazon Elastic Compute Cloud
 (Amazon EC2) security group.
#
# Parameters:
#       -g security_group_id - The ID of the security group.
#       -i ip_address - The IP address or CIDR block to authorize.
#       -p protocol - The protocol to authorize (e.g., tcp, udp, icmp).
#       -f from_port - The start of the port range to authorize.
#       -t to_port - The end of the port range to authorize.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_authorize_security_group_ingress() {
  local security_group_id ip_address protocol from_port to_port response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_authorize_security_group_ingress"
```

```
   echo "Authorizes an ingress rule for an Amazon Elastic Compute Cloud (Amazon
EC2) security group."
   echo "  -g security_group_id - The ID of the security group."
   echo "  -i ip_address - The IP address or CIDR block to authorize."
   echo "  -p protocol - The protocol to authorize (e.g., tcp, udp, icmp)."
   echo "  -f from_port - The start of the port range to authorize."
   echo "  -t to_port - The end of the port range to authorize."
   echo ""
 }

 # Retrieve the calling parameters.
 while getopts "g:i:p:f:t:h" option; do
   case "${option}" in
     g) security_group_id="${OPTARG}" ;;
     i) ip_address="${OPTARG}" ;;
     p) protocol="${OPTARG}" ;;
     f) from_port="${OPTARG}" ;;
     t) to_port="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$security_group_id" ]]; then
   errecho "ERROR: You must provide a security group ID with the -g parameter."
   usage
   return 1
 fi

 if [[ -z "$ip_address" ]]; then
   errecho "ERROR: You must provide an IP address or CIDR block with the -i
parameter."
   usage
   return 1
 fi
```

```
  if [[ -z "$protocol" ]]; then
    errecho "ERROR: You must provide a protocol with the -p parameter."
    usage
    return 1
  fi

  if [[ -z "$from_port" ]]; then
    errecho "ERROR: You must provide a start port with the -f parameter."
    usage
    return 1
  fi

  if [[ -z "$to_port" ]]; then
    errecho "ERROR: You must provide an end port with the -t parameter."
    usage
    return 1
  fi

  response=$(aws ec2 authorize-security-group-ingress \
    --group-id "$security_group_id" \
    --cidr "${ip_address}/32" \
    --protocol "$protocol" \
    --port "$from_port-$to_port" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports authorize-security-group-ingress operation failed.
$response"
    return 1
  }

  return 0
}

################################################################################
# function ec2_describe_images
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
 images.
#
# Parameters:
#       -i image_ids - A space-separated  list of image IDs (optional).
#       -h - Display help.
#
# And:
```

```
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_describe_images() {
  local image_ids response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_images"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) images."
    echo "  -i image_ids - A space-separated list of image IDs (optional)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) image_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local aws_cli_args=()

  if [[ -n "$image_ids" ]]; then
    # shellcheck disable=SC2206
    aws_cli_args+=("--image-ids" $image_ids)
  fi

  response=$(aws ec2 describe-images \
    "${aws_cli_args[@]}" \
    --query 'Images[*].[Description,Architecture,ImageId]' \
    --output text) || {
```

```
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-images operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}

################################################################################
# ec2_describe_instance_types
#
# This function describes EC2 instance types filtered by processor architecture
# and optionally by instance type. It takes the following arguments:
#
# -a, --architecture ARCHITECTURE  Specify the processor architecture (e.g., x86_64)
# -t, --type INSTANCE_TYPE         Comma-separated list of instance types (e.g.,
 t2.micro)
# -h, --help                       Show the usage help
#
# The function prints the instance type and supported architecture for each
# matching instance type.
################################################################################
function ec2_describe_instance_types() {
  local architecture=""
  local instance_types=""

  # bashsupport disable=BP5008
  function usage() {
    echo "Usage: ec2_describe_instance_types [-a|--architecture ARCHITECTURE] [-t|--
type INSTANCE_TYPE] [-h|--help]"
    echo "  -a, --architecture ARCHITECTURE  Specify the processor architecture
 (e.g., x86_64)"
    echo "  -t, --type INSTANCE_TYPE         Comma-separated list of instance types
 (e.g., t2.micro)"
    echo "  -h, --help                       Show this help message"
  }

  while [[ $# -gt 0 ]]; do
    case "$1" in
      -a | --architecture)
        architecture="$2"
        shift 2
```

```
      ;;
    -t | --type)
      instance_types="$2"
      shift 2
      ;;
    -h | --help)
      usage
      return 0
      ;;
    *)
      echo "Unknown argument: $1"
      return 1
      ;;
  esac
done

if [[ -z "$architecture" ]]; then
  errecho "Error: Architecture not specified."
  usage
  return 1
fi

if [[ -z "$instance_types" ]]; then
  errecho "Error: Instance type not specified."
  usage
  return 1
fi

local tmp_json_file="temp_ec2.json"
echo -n '[
  {
    "Name": "processor-info.supported-architecture",
    "Values": [' >"$tmp_json_file"

local items
IFS=',' read -ra items <<<"$architecture"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
  echo -n '"'"${items[$i]}"'"' >>"$tmp_json_file"
  if [[ $i -lt $((array_size - 1)) ]]; then
    echo -n ',' >>"$tmp_json_file"
  fi
done
```

```
  echo -n ']},
    {
    "Name": "instance-type",
      "Values": [' >>"$tmp_json_file"
  IFS=',' read -ra items <<<"$instance_types"
  local array_size
  array_size=${#items[@]}
  for i in $(seq 0 $((array_size - 1))); do
    echo -n '"'"${items[$i]}"'"' >>"$tmp_json_file"
    if [[ $i -lt $((array_size - 1)) ]]; then
      echo -n ',' >>"$tmp_json_file"
    fi
  done

  echo -n ']}]' >>"$tmp_json_file"

  local response
  response=$(aws ec2 describe-instance-types --filters file://"$tmp_json_file" \
    --query 'InstanceTypes[*].[InstanceType]' --output text)

  local error_code=$?

  rm "$tmp_json_file"

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    echo "ERROR: AWS reports describe-instance-types operation failed."
    return 1
  fi

  echo "$response"
  return 0
}

###############################################################################
# function ec2_run_instances
#
# This function launches one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
#
# Parameters:
#       -i image_id - The ID of the Amazon Machine Image (AMI) to use.
#       -t instance_type - The instance type to use (e.g., t2.micro).
#       -k key_pair_name - The name of the key pair to use.
```

```
#       -s security_group_id - The ID of the security group to use.
#       -c count - The number of instances to launch (default: 1).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_run_instances() {
  local image_id instance_type key_pair_name security_group_id count response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_run_instances"
    echo "Launches one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo "  -i image_id - The ID of the Amazon Machine Image (AMI) to use."
    echo "  -t instance_type - The instance type to use (e.g., t2.micro)."
    echo "  -k key_pair_name - The name of the key pair to use."
    echo "  -s security_group_id - The ID of the security group to use."
    echo "  -c count - The number of instances to launch (default: 1)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:t:k:s:c:h" option; do
    case "${option}" in
      i) image_id="${OPTARG}" ;;
      t) instance_type="${OPTARG}" ;;
      k) key_pair_name="${OPTARG}" ;;
      s) security_group_id="${OPTARG}" ;;
      c) count="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
```

```
 export OPTIND=1

 if [[ -z "$image_id" ]]; then
   errecho "ERROR: You must provide an Amazon Machine Image (AMI) ID with the -i
parameter."
   usage
   return 1
 fi

 if [[ -z "$instance_type" ]]; then
   errecho "ERROR: You must provide an instance type with the -t parameter."
   usage
   return 1
 fi

 if [[ -z "$key_pair_name" ]]; then
   errecho "ERROR: You must provide a key pair name with the -k parameter."
   usage
   return 1
 fi

 if [[ -z "$security_group_id" ]]; then
   errecho "ERROR: You must provide a security group ID with the -s parameter."
   usage
   return 1
 fi

 if [[ -z "$count" ]]; then
   count=1
 fi

 response=$(aws ec2 run-instances \
   --image-id "$image_id" \
   --instance-type "$instance_type" \
   --key-name "$key_pair_name" \
   --security-group-ids "$security_group_id" \
   --count "$count" \
   --query 'Instances[*].[InstanceId]' \
   --output text) || {
   aws_cli_error_log ${?}
   errecho "ERROR: AWS reports run-instances operation failed.$response"
   return 1
 }
```

```
  echo "$response"

  return 0
}

###############################################################################
# function ec2_describe_instances
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
#
# Parameters:
#       -i instance_id - The ID of the instance to describe (optional).
#       -q query - The query to filter the response (optional).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_describe_instances() {
  local instance_id query response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_instances"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances."
    echo "  -i instance_id - The ID of the instance to describe (optional)."
    echo "  -q query - The query to filter the response (optional)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:q:h" option; do
    case "${option}" in
      i) instance_id="${OPTARG}" ;;
      q) query="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
```

```
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local aws_cli_args=()

  if [[ -n "$instance_id" ]]; then
    # shellcheck disable=SC2206
    aws_cli_args+=("--instance-ids" $instance_id)
  fi

  local query_arg=""
  if [[ -n "$query" ]]; then
    query_arg="--query '$query'"
  else
    query_arg="--query Reservations[*].Instances[*].
[InstanceId,ImageId,InstanceType,KeyName,VpcId,PublicIpAddress,State.Name]"
  fi

  # shellcheck disable=SC2086
  response=$(aws ec2 describe-instances \
    "${aws_cli_args[@]}" \
    $query_arg \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-instances operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}

###############################################################################
# function ec2_stop_instances
#
# This function stops one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
```

```
#
# Parameters:
#       -i instance_id - The ID(s) of the instance(s) to stop (comma-separated).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_stop_instances() {
  local instance_ids
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_stop_instances"
    echo "Stops one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo "  -i instance_id - The ID(s) of the instance(s) to stop (comma-
separated)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) instance_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$instance_ids" ]]; then
    errecho "ERROR: You must provide one or more instance IDs with the -i
 parameter."
    usage
```

```
    return 1
  fi

  response=$(aws ec2 stop-instances \
    --instance-ids "${instance_ids}") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports stop-instances operation failed with $response."
    return 1
  }

  return 0
}

###############################################################################
# function ec2_start_instances
#
# This function starts one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
#
# Parameters:
#       -i instance_id - The ID(s) of the instance(s) to start (comma-separated).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_start_instances() {
  local instance_ids
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_start_instances"
    echo "Starts one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo "  -i instance_id - The ID(s) of the instance(s) to start (comma-
separated)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
```

```
      i) instance_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$instance_ids" ]]; then
    errecho "ERROR: You must provide one or more instance IDs with the -i
 parameter."
    usage
    return 1
  fi

  response=$(aws ec2 start-instances \
    --instance-ids "${instance_ids}") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports start-instances operation failed with $response."
    return 1
  }

  return 0
}

###############################################################################
# function ec2_allocate_address
#
# This function allocates an Elastic IP address for use with Amazon Elastic Compute
 Cloud (Amazon EC2) instances in a specific AWS Region.
#
# Parameters:
#       -d domain - The domain for the Elastic IP address (either 'vpc' or
 'standard').
#
# Returns:
#       The allocated Elastic IP address, or an error message if the operation
 fails.
```

```
# And:
#       0 - If successful.
#       1 - If it fails.
#
################################################################################
function ec2_allocate_address() {
  local domain response

  # Function to display usage information
  function usage() {
    echo "function ec2_allocate_address"
    echo "Allocates an Elastic IP address for use with Amazon Elastic Compute Cloud
 (Amazon EC2) instances in a specific AWS Region."
    echo "  -d domain - The domain for the Elastic IP address (either 'vpc' or
 'standard')."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "d:h" option; do
    case "${option}" in
      d) domain="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$domain" ]]; then
    errecho "ERROR: You must provide a domain with the -d parameter (either 'vpc' or
 'standard')."
    return 1
  fi

  if [[ "$domain" != "vpc" && "$domain" != "standard" ]]; then
    errecho "ERROR: Invalid domain value. Must be either 'vpc' or 'standard'."
```

```
    return 1
  fi

  # Allocate the Elastic IP address
  response=$(aws ec2 allocate-address \
    --domain "$domain" \
    --query "[PublicIp,AllocationId]" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports allocate-address operation failed."
    errecho "$response"
    return 1
  }

  echo "$response"
  return 0
}

################################################################################
# function ec2_associate_address
#
# This function associates an Elastic IP address with an Amazon Elastic Compute
 Cloud (Amazon EC2) instance.
#
# Parameters:
#       -a allocation_id - The allocation ID of the Elastic IP address to associate.
#       -i instance_id - The ID of the EC2 instance to associate the Elastic IP
 address with.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
#
################################################################################
function ec2_associate_address() {
  local allocation_id instance_id response

  # Function to display usage information
  function usage() {
    echo "function ec2_associate_address"
    echo "Associates an Elastic IP address with an Amazon Elastic Compute Cloud
 (Amazon EC2) instance."
    echo "  -a allocation_id - The allocation ID of the Elastic IP address to
 associate."
```

```
    echo "  -i instance_id - The ID of the EC2 instance to associate the Elastic IP
address with."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "a:i:h" option; do
    case "${option}" in
      a) allocation_id="${OPTARG}" ;;
      i) instance_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$allocation_id" ]]; then
    errecho "ERROR: You must provide an allocation ID with the -a parameter."
    return 1
  fi

  if [[ -z "$instance_id" ]]; then
    errecho "ERROR: You must provide an instance ID with the -i parameter."
    return 1
  fi

  # Associate the Elastic IP address
  response=$(aws ec2 associate-address \
    --allocation-id "$allocation_id" \
    --instance-id "$instance_id" \
    --query "AssociationId" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports associate-address operation failed."
    errecho "$response"
    return 1
```

```
  }

  echo "$response"
  return 0
}

###############################################################################
# function ec2_disassociate_address
#
# This function disassociates an Elastic IP address from an Amazon Elastic Compute
 Cloud (Amazon EC2) instance.
#
# Parameters:
#       -a association_id - The association ID that represents the association of
 the Elastic IP address with an instance.
#
# And:
#       0 - If successful.
#       1 - If it fails.
#
###############################################################################
function ec2_disassociate_address() {
  local association_id response

  # Function to display usage information
  function usage() {
    echo "function ec2_disassociate_address"
    echo "Disassociates an Elastic IP address from an Amazon Elastic Compute Cloud
 (Amazon EC2) instance."
    echo "  -a association_id - The association ID that represents the association
 of the Elastic IP address with an instance."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "a:h" option; do
    case "${option}" in
      a) association_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
```

```
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$association_id" ]]; then
    errecho "ERROR: You must provide an association ID with the -a parameter."
    return 1
  fi

  response=$(aws ec2 disassociate-address \
    --association-id "$association_id") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports disassociate-address operation failed."
    errecho "$response"
    return 1
  }

  return 0
}

###############################################################################
# function ec2_release_address
#
# This function releases an Elastic IP address from an Amazon Elastic Compute Cloud
 (Amazon EC2) instance.
#
# Parameters:
#       -a allocation_id - The allocation ID of the Elastic IP address to release.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
#
###############################################################################
function ec2_release_address() {
  local allocation_id response

  # Function to display usage information
  function usage() {
    echo "function ec2_release_address"
```

```
    echo "Releases an Elastic IP address from an Amazon Elastic Compute Cloud
 (Amazon EC2) instance."
    echo "  -a allocation_id - The allocation ID of the Elastic IP address to
 release."
    echo ""
 }

 # Parse the command-line arguments
 while getopts "a:h" option; do
   case "${option}" in
     a) allocation_id="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 # Validate the input parameters
 if [[ -z "$allocation_id" ]]; then
   errecho "ERROR: You must provide an allocation ID with the -a parameter."
   return 1
 fi

 response=$(aws ec2 release-address \
   --allocation-id "$allocation_id") || {
   aws_cli_error_log ${?}
   errecho "ERROR: AWS reports release-address operation failed."
   errecho "$response"
   return 1
 }

 return 0
}

###############################################################################
# function ec2_terminate_instances
#
```

```
# This function terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances using the AWS CLI.
#
# Parameters:
#       -i instance_ids - A space-separated list of instance IDs.
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_terminate_instances() {
  local instance_ids response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_terminate_instances"
    echo "Terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances."
    echo "  -i instance_ids - A space-separated list of instance IDs."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) instance_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Check if instance ID is provided
  if [[ -z "${instance_ids}" ]]; then
```

```
    echo "Error: Missing required instance IDs parameter."
    usage
    return 1
  fi

  # shellcheck disable=SC2086
  response=$(aws ec2 terminate-instances \
    "--instance-ids" $instance_ids \
    --query 'TerminatingInstances[*].[InstanceId,CurrentState.Name]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports terminate-instances operation failed.$response"
    return 1
  }

  return 0
}

###############################################################################
# function ec2_delete_security_group
#
# This function deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Parameters:
#       -i security_group_id - The ID of the security group to delete.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_delete_security_group() {
  local security_group_id response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_delete_security_group"
    echo "Deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group."
    echo "  -i security_group_id - The ID of the security group to delete."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
```

```
    case "${option}" in
      i) security_group_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -i parameter."
    usage
    return 1
  fi

  response=$(aws ec2 delete-security-group --group-id "$security_group_id" --output
 text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-security-group operation failed.$response"
    return 1
  }

  return 0
}

###############################################################################
# function ec2_delete_keypair
#
# This function deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair.
#
# Parameters:
#       -n key_pair_name - A key pair name.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_delete_keypair() {
```

```
  local key_pair_name response

  local option OPTARG # Required to use getopts command in a function.
  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_delete_keypair"
    echo "Deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair."
    echo "  -n key_pair_name - A key pair name."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) key_pair_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key pair name with the -n parameter."
    usage
    return 1
  fi

  response=$(aws ec2 delete-key-pair \
    --key-name "$key_pair_name") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-key-pair operation failed.$response"
    return 1
  }

  return 0
}
```

此案例中使用的公用程式函數。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
##############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi
```

```
  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - AllocateAddress
  - AssociateAddress
  - AuthorizeSecurityGroupIngress
  - CreateKeyPair
  - CreateSecurityGroup
  - DeleteKeyPair
  - DeleteSecurityGroup
  - DescribeImages
  - DescribeInstanceTypes
  - DescribeInstances
  - DescribeKeyPairs
  - DescribeSecurityGroups
  - DisassociateAddress
  - ReleaseAddress
  - RunInstances
  - StartInstances
  - StopInstances
  - TerminateInstances
  - UnmonitorInstances

動作

`AllocateAddress`

以下程式碼範例顯示如何使用 `AllocateAddress`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
#############################################################################
# function ec2_allocate_address
#
# This function allocates an Elastic IP address for use with Amazon Elastic Compute
 Cloud (Amazon EC2) instances in a specific AWS Region.
#
# Parameters:
#       -d domain - The domain for the Elastic IP address (either 'vpc' or
 'standard').
#
# Returns:
#       The allocated Elastic IP address, or an error message if the operation
 fails.
# And:
#       0 - If successful.
#       1 - If it fails.
#
#############################################################################
function ec2_allocate_address() {
  local domain response

  # Function to display usage information
  function usage() {
    echo "function ec2_allocate_address"
    echo "Allocates an Elastic IP address for use with Amazon Elastic Compute Cloud
 (Amazon EC2) instances in a specific AWS Region."
    echo "  -d domain - The domain for the Elastic IP address (either 'vpc' or
 'standard')."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "d:h" option; do
    case "${option}" in
```

```
      d) domain="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$domain" ]]; then
    errecho "ERROR: You must provide a domain with the -d parameter (either 'vpc' or
 'standard')."
    return 1
  fi

  if [[ "$domain" != "vpc" && "$domain" != "standard" ]]; then
    errecho "ERROR: Invalid domain value. Must be either 'vpc' or 'standard'."
    return 1
  fi

  # Allocate the Elastic IP address
  response=$(aws ec2 allocate-address \
    --domain "$domain" \
    --query "[PublicIp,AllocationId]" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports allocate-address operation failed."
    errecho "$response"
    return 1
  }

  echo "$response"
  return 0
}
```

此範例中使用的公用程式函數。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
##############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AllocateAddress。

`AssociateAddress`

以下程式碼範例顯示如何使用 `AssociateAddress`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function ec2_associate_address
#
# This function associates an Elastic IP address with an Amazon Elastic Compute
 Cloud (Amazon EC2) instance.
#
# Parameters:
#       -a allocation_id - The allocation ID of the Elastic IP address to associate.
#       -i instance_id - The ID of the EC2 instance to associate the Elastic IP
 address with.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
#
###############################################################################
function ec2_associate_address() {
  local allocation_id instance_id response

  # Function to display usage information
  function usage() {
    echo "function ec2_associate_address"
    echo "Associates an Elastic IP address with an Amazon Elastic Compute Cloud
 (Amazon EC2) instance."
    echo "  -a allocation_id - The allocation ID of the Elastic IP address to
 associate."
    echo "  -i instance_id - The ID of the EC2 instance to associate the Elastic IP
 address with."
```

```
  echo ""
}

# Parse the command-line arguments
while getopts "a:i:h" option; do
  case "${option}" in
    a) allocation_id="${OPTARG}" ;;
    i) instance_id="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$allocation_id" ]]; then
  errecho "ERROR: You must provide an allocation ID with the -a parameter."
  return 1
fi

if [[ -z "$instance_id" ]]; then
  errecho "ERROR: You must provide an instance ID with the -i parameter."
  return 1
fi

# Associate the Elastic IP address
response=$(aws ec2 associate-address \
  --allocation-id "$allocation_id" \
  --instance-id "$instance_id" \
  --query "AssociationId" \
  --output text) || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports associate-address operation failed."
  errecho "$response"
  return 1
}
```

```
  echo "$response"
  return 0
}
```

此範例中使用的公用程式函數。

```
################################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
################################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

################################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
```

```
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AssociateAddress。

`AuthorizeSecurityGroupIngress`

以下程式碼範例顯示如何使用 `AuthorizeSecurityGroupIngress`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function ec2_authorize_security_group_ingress
#
# This function authorizes an ingress rule for an Amazon Elastic Compute Cloud
 (Amazon EC2) security group.
#
# Parameters:
#       -g security_group_id - The ID of the security group.
#       -i ip_address - The IP address or CIDR block to authorize.
#       -p protocol - The protocol to authorize (e.g., tcp, udp, icmp).
#       -f from_port - The start of the port range to authorize.
#       -t to_port - The end of the port range to authorize.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_authorize_security_group_ingress() {
  local security_group_id ip_address protocol from_port to_port response
  local option OPTARG # Required to use getopts command in a function.
```

```
 # bashsupport disable=BP5008
 function usage() {
   echo "function ec2_authorize_security_group_ingress"
   echo "Authorizes an ingress rule for an Amazon Elastic Compute Cloud (Amazon
EC2) security group."
   echo "  -g security_group_id - The ID of the security group."
   echo "  -i ip_address - The IP address or CIDR block to authorize."
   echo "  -p protocol - The protocol to authorize (e.g., tcp, udp, icmp)."
   echo "  -f from_port - The start of the port range to authorize."
   echo "  -t to_port - The end of the port range to authorize."
   echo ""
 }

 # Retrieve the calling parameters.
 while getopts "g:i:p:f:t:h" option; do
   case "${option}" in
     g) security_group_id="${OPTARG}" ;;
     i) ip_address="${OPTARG}" ;;
     p) protocol="${OPTARG}" ;;
     f) from_port="${OPTARG}" ;;
     t) to_port="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$security_group_id" ]]; then
   errecho "ERROR: You must provide a security group ID with the -g parameter."
   usage
   return 1
 fi

 if [[ -z "$ip_address" ]]; then
   errecho "ERROR: You must provide an IP address or CIDR block with the -i
parameter."
```

```
    usage
    return 1
  fi

  if [[ -z "$protocol" ]]; then
    errecho "ERROR: You must provide a protocol with the -p parameter."
    usage
    return 1
  fi

  if [[ -z "$from_port" ]]; then
    errecho "ERROR: You must provide a start port with the -f parameter."
    usage
    return 1
  fi

  if [[ -z "$to_port" ]]; then
    errecho "ERROR: You must provide an end port with the -t parameter."
    usage
    return 1
  fi

  response=$(aws ec2 authorize-security-group-ingress \
    --group-id "$security_group_id" \
    --cidr "${ip_address}/32" \
    --protocol "$protocol" \
    --port "$from_port-$to_port" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports authorize-security-group-ingress operation failed.
$response"
    return 1
  }

  return 0
}
```

此範例中使用的公用程式函數。

```
###############################################################################
# function errecho
#
```

```
# This function outputs everything sent to it to STDERR (standard error output).
#####################################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#####################################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
#####################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AuthorizeSecurityGroupIngress。

`CreateKeyPair`

以下程式碼範例顯示如何使用 `CreateKeyPair`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function ec2_create_keypair
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or
 2048-bit RSA key pair
# and writes it to a file.
#
# Parameters:
#       -n key_pair_name - A key pair name.
#       -f file_path - File to store the key pair.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_create_keypair() {
  local key_pair_name file_path response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_create_keypair"
    echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or 2048-bit
 RSA key pair"
    echo " and writes it to a file."
    echo "  -n key_pair_name - A key pair name."
    echo "  -f file_path - File to store the key pair."
    echo ""
  }

  # Retrieve the calling parameters.
```

```
  while getopts "n:f:h" option; do
    case "${option}" in
      n) key_pair_name="${OPTARG}" ;;
      f) file_path="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$file_path" ]]; then
    errecho "ERROR: You must provide a file path with the -f parameter."
    usage
    return 1
  fi

  response=$(aws ec2 create-key-pair \
    --key-name "$key_pair_name" \
    --query 'KeyMaterial' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
  }

  if [[ -n "$file_path" ]]; then
    echo "$response" >"$file_path"
  fi

  return 0
}
```

此範例中使用的公用程式函數。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
##############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi
```

```
  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateKeyPair。

`CreateSecurityGroup`

以下程式碼範例顯示如何使用 `CreateSecurityGroup`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function ec2_create_security_group
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Parameters:
#       -n security_group_name - The name of the security group.
#       -d security_group_description - The description of the security group.
#
# Returns:
#       The ID of the created security group, or an error message if the operation
 fails.
# And:
#       0 - If successful.
#       1 - If it fails.
#
###############################################################################
function ec2_create_security_group() {
  local security_group_name security_group_description response

  # Function to display usage information
  function usage() {
    echo "function ec2_create_security_group"
```

```
    echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group."
    echo "  -n security_group_name - The name of the security group."
    echo "  -d security_group_description - The description of the security group."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "n:d:h" option; do
    case "${option}" in
      n) security_group_name="${OPTARG}" ;;
      d) security_group_description="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$security_group_name" ]]; then
    errecho "ERROR: You must provide a security group name with the -n parameter."
    return 1
  fi

  if [[ -z "$security_group_description" ]]; then
    errecho "ERROR: You must provide a security group description with the -d
parameter."
    return 1
  fi

  # Create the security group
  response=$(aws ec2 create-security-group \
    --group-name "$security_group_name" \
    --description "$security_group_description" \
    --query "GroupId" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports create-security-group operation failed."
```

```
    errecho "$response"
    return 1
  }

  echo "$response"
  return 0
}
```

此範例中使用的公用程式函數。

```
################################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
################################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

################################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
```

```
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateSecurityGroup。

`DeleteKeyPair`

以下程式碼範例顯示如何使用 `DeleteKeyPair`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function ec2_delete_keypair
#
# This function deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair.
#
# Parameters:
#       -n key_pair_name - A key pair name.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_delete_keypair() {
  local key_pair_name response

  local option OPTARG # Required to use getopts command in a function.
```

```
  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_delete_keypair"
    echo "Deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair."
    echo "  -n key_pair_name - A key pair name."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) key_pair_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key pair name with the -n parameter."
    usage
    return 1
  fi

  response=$(aws ec2 delete-key-pair \
    --key-name "$key_pair_name") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-key-pair operation failed.$response"
    return 1
  }

  return 0
}
```

此範例中使用的公用程式函數。

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
#############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteKeyPair。

`DeleteSecurityGroup`

以下程式碼範例顯示如何使用 `DeleteSecurityGroup`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function ec2_delete_security_group
#
# This function deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Parameters:
#       -i security_group_id - The ID of the security group to delete.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_delete_security_group() {
  local security_group_id response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_delete_security_group"
    echo "Deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group."
    echo "  -i security_group_id - The ID of the security group to delete."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) security_group_id="${OPTARG}" ;;
```

```
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -i parameter."
    usage
    return 1
  fi

  response=$(aws ec2 delete-security-group --group-id "$security_group_id" --output
 text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-security-group operation failed.$response"
    return 1
  }

  return 0
}
```

此範例中使用的公用程式函數。

```
########################################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
########################################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

########################################################################################
# function aws_cli_error_log()
```

```
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteSecurityGroup。

DescribeImages

以下程式碼範例顯示如何使用 DescribeImages。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
#############################################################################
# function ec2_describe_images
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
 images.
#
# Parameters:
#       -i image_ids - A space-separated  list of image IDs (optional).
#       -h - Display help.
#
# And:
#       0 - If successful.
#       1 - If it fails.
#############################################################################
function ec2_describe_images() {
  local image_ids response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_images"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) images."
    echo "  -i image_ids - A space-separated list of image IDs (optional)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) image_ids="${OPTARG}" ;;
      h)
        usage
        return 0
```

```
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local aws_cli_args=()

  if [[ -n "$image_ids" ]]; then
    # shellcheck disable=SC2206
    aws_cli_args+=("--image-ids" $image_ids)
  fi

  response=$(aws ec2 describe-images \
    "${aws_cli_args[@]}" \
    --query 'Images[*].[Description,Architecture,ImageId]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-images operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}
```

此範例中使用的公用程式函數。

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}
```

```
###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeImages。

DescribeInstanceTypes

以下程式碼範例顯示如何使用 DescribeInstanceTypes。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# ec2_describe_instance_types
#
# This function describes EC2 instance types filtered by processor architecture
# and optionally by instance type. It takes the following arguments:
#
# -a, --architecture ARCHITECTURE  Specify the processor architecture (e.g., x86_64)
# -t, --type INSTANCE_TYPE         Comma-separated list of instance types (e.g.,
 t2.micro)
# -h, --help                       Show the usage help
#
# The function prints the instance type and supported architecture for each
# matching instance type.
##############################################################################
function ec2_describe_instance_types() {
  local architecture=""
  local instance_types=""

  # bashsupport disable=BP5008
  function usage() {
    echo "Usage: ec2_describe_instance_types [-a|--architecture ARCHITECTURE] [-t|--
type INSTANCE_TYPE] [-h|--help]"
    echo "  -a, --architecture ARCHITECTURE  Specify the processor architecture
 (e.g., x86_64)"
    echo "  -t, --type INSTANCE_TYPE         Comma-separated list of instance types
 (e.g., t2.micro)"
    echo "  -h, --help                       Show this help message"
  }

  while [[ $# -gt 0 ]]; do
    case "$1" in
      -a | --architecture)
        architecture="$2"
        shift 2
```

```
      ;;
    -t | --type)
      instance_types="$2"
      shift 2
      ;;
    -h | --help)
      usage
      return 0
      ;;
    *)
      echo "Unknown argument: $1"
      return 1
      ;;
  esac
done

if [[ -z "$architecture" ]]; then
  errecho "Error: Architecture not specified."
  usage
  return 1
fi

if [[ -z "$instance_types" ]]; then
  errecho "Error: Instance type not specified."
  usage
  return 1
fi

local tmp_json_file="temp_ec2.json"
echo -n '[
  {
    "Name": "processor-info.supported-architecture",
    "Values": [' >"$tmp_json_file"

local items
IFS=',' read -ra items <<<"$architecture"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
  echo -n '"'"${items[$i]}"'"' >>"$tmp_json_file"
  if [[ $i -lt $((array_size - 1)) ]]; then
    echo -n ',' >>"$tmp_json_file"
  fi
done
```

```
  echo -n ']},
    {
    "Name": "instance-type",
      "Values": [' >>"$tmp_json_file"
  IFS=',' read -ra items <<<"$instance_types"
  local array_size
  array_size=${#items[@]}
  for i in $(seq 0 $((array_size - 1))); do
    echo -n '"'"${items[$i]}"'"' >>"$tmp_json_file"
    if [[ $i -lt $((array_size - 1)) ]]; then
      echo -n ',' >>"$tmp_json_file"
    fi
  done

  echo -n ']}]' >>"$tmp_json_file"

  local response
  response=$(aws ec2 describe-instance-types --filters file://"$tmp_json_file" \
    --query 'InstanceTypes[*].[InstanceType]' --output text)

  local error_code=$?

  rm "$tmp_json_file"

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    echo "ERROR: AWS reports describe-instance-types operation failed."
    return 1
  fi

  echo "$response"
  return 0
}
```

此範例中使用的公用程式函數。

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
```

```
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeInstanceTypes。

`DescribeInstances`

以下程式碼範例顯示如何使用 `DescribeInstances`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function ec2_describe_instances
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
#
# Parameters:
#       -i instance_id - The ID of the instance to describe (optional).
#       -q query - The query to filter the response (optional).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function ec2_describe_instances() {
  local instance_id query response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_instances"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances."
    echo "  -i instance_id - The ID of the instance to describe (optional)."
    echo "  -q query - The query to filter the response (optional)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:q:h" option; do
    case "${option}" in
      i) instance_id="${OPTARG}" ;;
```

```
      q) query="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local aws_cli_args=()

  if [[ -n "$instance_id" ]]; then
    # shellcheck disable=SC2206
    aws_cli_args+=("--instance-ids" $instance_id)
  fi

  local query_arg=""
  if [[ -n "$query" ]]; then
    query_arg="--query '$query'"
  else
    query_arg="--query Reservations[*].Instances[*].
[InstanceId,ImageId,InstanceType,KeyName,VpcId,PublicIpAddress,State.Name]"
  fi

  # shellcheck disable=SC2086
  response=$(aws ec2 describe-instances \
    "${aws_cli_args[@]}" \
    $query_arg \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-instances operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}
```

此範例中使用的公用程式函數。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
##############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi
```

```
  return 0
}
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeInstances。

`DescribeKeyPairs`

以下程式碼範例顯示如何使用 `DescribeKeyPairs`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function ec2_describe_key_pairs
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
 pairs.
#
# Parameters:
#       -h - Display help.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_describe_key_pairs() {
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_key_pairs"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
 pairs."
    echo "  -h - Display help."
    echo ""
```

```
  }

  # Retrieve the calling parameters.
  while getopts "h" option; do
    case "${option}" in
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local response

  response=$(aws ec2 describe-key-pairs \
    --query 'KeyPairs[*].[KeyName, KeyFingerprint]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-key-pairs operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}
```

此範例中使用的公用程式函數。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
```

```
}

##############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
##############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeKeyPairs。

DescribeSecurityGroups

以下程式碼範例顯示如何使用 DescribeSecurityGroups。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
#############################################################################
# function ec2_describe_security_groups
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
 security groups.
#
# Parameters:
#       -g security_group_id - The ID of the security group to describe (optional).
#
# And:
#       0 - If successful.
#       1 - If it fails.
#############################################################################
function ec2_describe_security_groups() {
  local security_group_id response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_security_groups"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) security
 groups."
    echo "  -g security_group_id - The ID of the security group to describe
 (optional)."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "g:h" option; do
    case "${option}" in
      g) security_group_id="${OPTARG}" ;;
      h)
        usage
        return 0
```

```
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local query="SecurityGroups[*].[GroupName, GroupId, VpcId, IpPermissions[*].
[IpProtocol, FromPort, ToPort, IpRanges[*].CidrIp]]"

  if [[ -n "$security_group_id" ]]; then
    response=$(aws ec2 describe-security-groups --group-ids "$security_group_id" --
query "${query}" --output text)
  else
    response=$(aws ec2 describe-security-groups --query "${query}" --output text)
  fi

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports describe-security-groups operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

此範例中使用的公用程式函數。

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
```

```
}

##############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
##############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 的詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeSecurityGroups。

DisassociateAddress

以下程式碼範例顯示如何使用 DisassociateAddress。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function ec2_disassociate_address
#
# This function disassociates an Elastic IP address from an Amazon Elastic Compute
 Cloud (Amazon EC2) instance.
#
# Parameters:
#       -a association_id - The association ID that represents the association of
 the Elastic IP address with an instance.
#
# And:
#       0 - If successful.
#       1 - If it fails.
#
##############################################################################
function ec2_disassociate_address() {
  local association_id response

  # Function to display usage information
  function usage() {
    echo "function ec2_disassociate_address"
    echo "Disassociates an Elastic IP address from an Amazon Elastic Compute Cloud
 (Amazon EC2) instance."
    echo "  -a association_id - The association ID that represents the association
 of the Elastic IP address with an instance."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "a:h" option; do
    case "${option}" in
      a) association_id="${OPTARG}" ;;
      h)
        usage
```

```
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$association_id" ]]; then
    errecho "ERROR: You must provide an association ID with the -a parameter."
    return 1
  fi

  response=$(aws ec2 disassociate-address \
    --association-id "$association_id") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports disassociate-address operation failed."
    errecho "$response"
    return 1
  }

  return 0
}
```

此範例中使用的公用程式函數。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function aws_cli_error_log()
#
```

```
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DisassociateAddress。

`ReleaseAddress`

以下程式碼範例顯示如何使用 `ReleaseAddress`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function ec2_release_address
#
# This function releases an Elastic IP address from an Amazon Elastic Compute Cloud
 (Amazon EC2) instance.
#
# Parameters:
#       -a allocation_id - The allocation ID of the Elastic IP address to release.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
#
##############################################################################
function ec2_release_address() {
  local allocation_id response

  # Function to display usage information
  function usage() {
    echo "function ec2_release_address"
    echo "Releases an Elastic IP address from an Amazon Elastic Compute Cloud
 (Amazon EC2) instance."
    echo "  -a allocation_id - The allocation ID of the Elastic IP address to
 release."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "a:h" option; do
    case "${option}" in
      a) allocation_id="${OPTARG}" ;;
      h)
        usage
        return 0
```

```
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$allocation_id" ]]; then
    errecho "ERROR: You must provide an allocation ID with the -a parameter."
    return 1
  fi

  response=$(aws ec2 release-address \
    --allocation-id "$allocation_id") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports release-address operation failed."
    errecho "$response"
    return 1
  }

  return 0
}
```

此範例中使用的公用程式函數。

```
############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
```

```
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
########################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ReleaseAddress。

RunInstances

以下程式碼範例顯示如何使用 RunInstances。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
#############################################################################
# function ec2_run_instances
#
# This function launches one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
#
# Parameters:
#       -i image_id - The ID of the Amazon Machine Image (AMI) to use.
#       -t instance_type - The instance type to use (e.g., t2.micro).
#       -k key_pair_name - The name of the key pair to use.
#       -s security_group_id - The ID of the security group to use.
#       -c count - The number of instances to launch (default: 1).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
#############################################################################
function ec2_run_instances() {
  local image_id instance_type key_pair_name security_group_id count response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_run_instances"
    echo "Launches one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo "  -i image_id - The ID of the Amazon Machine Image (AMI) to use."
    echo "  -t instance_type - The instance type to use (e.g., t2.micro)."
    echo "  -k key_pair_name - The name of the key pair to use."
    echo "  -s security_group_id - The ID of the security group to use."
    echo "  -c count - The number of instances to launch (default: 1)."
    echo "  -h - Display help."
    echo ""
  }
```

```
 # Retrieve the calling parameters.
 while getopts "i:t:k:s:c:h" option; do
   case "${option}" in
     i) image_id="${OPTARG}" ;;
     t) instance_type="${OPTARG}" ;;
     k) key_pair_name="${OPTARG}" ;;
     s) security_group_id="${OPTARG}" ;;
     c) count="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$image_id" ]]; then
   errecho "ERROR: You must provide an Amazon Machine Image (AMI) ID with the -i
parameter."
   usage
   return 1
 fi

 if [[ -z "$instance_type" ]]; then
   errecho "ERROR: You must provide an instance type with the -t parameter."
   usage
   return 1
 fi

 if [[ -z "$key_pair_name" ]]; then
   errecho "ERROR: You must provide a key pair name with the -k parameter."
   usage
   return 1
 fi

 if [[ -z "$security_group_id" ]]; then
   errecho "ERROR: You must provide a security group ID with the -s parameter."
   usage
```

```
    return 1
  fi

  if [[ -z "$count" ]]; then
    count=1
  fi

  response=$(aws ec2 run-instances \
    --image-id "$image_id" \
    --instance-type "$instance_type" \
    --key-name "$key_pair_name" \
    --security-group-ids "$security_group_id" \
    --count "$count" \
    --query 'Instances[*].[InstanceId]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports run-instances operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}
```

此範例中使用的公用程式函數。

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
```

```
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
################################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 RunInstances。

`StartInstances`

以下程式碼範例顯示如何使用 `StartInstances`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function ec2_start_instances
#
# This function starts one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
#
# Parameters:
#       -i instance_id - The ID(s) of the instance(s) to start (comma-separated).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_start_instances() {
  local instance_ids
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_start_instances"
    echo "Starts one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo "  -i instance_id - The ID(s) of the instance(s) to start (comma-
separated)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) instance_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
```

```
  export OPTIND=1

  if [[ -z "$instance_ids" ]]; then
    errecho "ERROR: You must provide one or more instance IDs with the -i
 parameter."
    usage
    return 1
  fi

  response=$(aws ec2 start-instances \
    --instance-ids "${instance_ids}") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports start-instances operation failed with $response."
    return 1
  }

  return 0
}
```

此範例中使用的公用程式函數。

```
############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
############################################################################
```

```
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 StartInstances。

`StopInstances`

以下程式碼範例顯示如何使用 `StopInstances`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function ec2_stop_instances
#
# This function stops one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
```

```
#
# Parameters:
#       -i instance_id - The ID(s) of the instance(s) to stop (comma-separated).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_stop_instances() {
  local instance_ids
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_stop_instances"
    echo "Stops one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo "  -i instance_id - The ID(s) of the instance(s) to stop (comma-
separated)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) instance_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$instance_ids" ]]; then
    errecho "ERROR: You must provide one or more instance IDs with the -i
 parameter."
    usage
```

```
    return 1
  fi

  response=$(aws ec2 stop-instances \
    --instance-ids "${instance_ids}") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports stop-instances operation failed with $response."
    return 1
  }

  return 0
}
```

此範例中使用的公用程式函數。

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
#############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
```

```
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 StopInstances。

`TerminateInstances`

以下程式碼範例顯示如何使用 `TerminateInstances`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function ec2_terminate_instances
#
# This function terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances using the AWS CLI.
#
# Parameters:
#       -i instance_ids - A space-separated list of instance IDs.
#       -h - Display help.
#
# Returns:
```

```
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_terminate_instances() {
  local instance_ids response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_terminate_instances"
    echo "Terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances."
    echo "  -i instance_ids - A space-separated list of instance IDs."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) instance_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Check if instance ID is provided
  if [[ -z "${instance_ids}" ]]; then
    echo "Error: Missing required instance IDs parameter."
    usage
    return 1
  fi

  # shellcheck disable=SC2086
  response=$(aws ec2 terminate-instances \
    "--instance-ids" $instance_ids \
```

```
    --query 'TerminatingInstances[*].[InstanceId,CurrentState.Name]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports terminate-instances operation failed.$response"
    return 1
  }

  return 0
}
```

此範例中使用的公用程式函數。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
##############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
```

```
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 TerminateInstances。

HealthImaging 範例 AWS CLI 搭配 Bash 指令碼使用

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Bash 指令碼搭配 HealthImaging 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`CreateDatastore`

以下程式碼範例顯示如何使用 `CreateDatastore`。

AWS CLI 使用 Bash 指令碼

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
```

```
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function imaging_create_datastore
#
# This function creates an AWS HealthImaging data store for importing DICOM P10
 files.
#
# Parameters:
#       -n data_store_name - The name of the data store.
#
# Returns:
#       The datastore ID.
#    And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function imaging_create_datastore() {
  local datastore_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function imaging_create_datastore"
    echo "Creates an AWS HealthImaging data store for importing DICOM P10 files."
    echo "  -n data_store_name - The name of the data store."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) datastore_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
```

```
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$datastore_name" ]]; then
    errecho "ERROR: You must provide a data store name with the -n parameter."
    usage
    return 1
  fi

  response=$(aws medical-imaging create-datastore \
    --datastore-name "$datastore_name" \
    --output text \
    --query 'datastoreId')

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports medical-imaging create-datastore operation failed.
$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateDatastore。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

`DeleteDatastore`

以下程式碼範例顯示如何使用 `DeleteDatastore`。

AWS CLI 使用 Bash 指令碼

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function imaging_delete_datastore
#
# This function deletes an AWS HealthImaging data store.
#
# Parameters:
#       -i datastore_id - The ID of the data store.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function imaging_delete_datastore() {
  local datastore_id response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function imaging_delete_datastore"
    echo "Deletes an AWS HealthImaging data store."
    echo "  -i datastore_id - The ID of the data store."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) datastore_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
```

```
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$datastore_id" ]]; then
    errecho "ERROR: You must provide a data store ID with the -i parameter."
    usage
    return 1
  fi

  response=$(aws medical-imaging delete-datastore \
    --datastore-id "$datastore_id")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports medical-imaging delete-datastore operation failed.
$response"
    return 1
  fi

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteDatastore。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

`GetDatastore`

以下程式碼範例顯示如何使用 `GetDatastore`。

AWS CLI 使用 Bash 指令碼

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function imaging_get_datastore
#
# Get a data store's properties.
#
# Parameters:
#       -i data_store_id - The ID of the data store.
#
# Returns:
#       [datastore_name, datastore_id, datastore_status, datastore_arn,  created_at,
 updated_at]
#    And:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function imaging_get_datastore() {
  local datastore_id option OPTARG # Required to use getopts command in a function.
  local error_code
  # bashsupport disable=BP5008
  function usage() {
    echo "function imaging_get_datastore"
    echo "Gets a data store's properties."
    echo "  -i datastore_id - The ID of the data store."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) datastore_id="${OPTARG}" ;;
      h)
        usage
        return 0
```

```
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$datastore_id" ]]; then
    errecho "ERROR: You must provide a data store ID with the -i parameter."
    usage
    return 1
  fi

  local response

  response=$(
    aws medical-imaging get-datastore \
      --datastore-id "$datastore_id" \
      --output text \
      --query "[ datastoreProperties.datastoreName,
 datastoreProperties.datastoreId, datastoreProperties.datastoreStatus,
 datastoreProperties.datastoreArn,  datastoreProperties.createdAt,
 datastoreProperties.updatedAt]"
  )
  error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports list-datastores operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetDatastore。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

`ListDatastores`

以下程式碼範例顯示如何使用 `ListDatastores`。

AWS CLI 使用 Bash 指令碼

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function imaging_list_datastores
#
# List the HealthImaging data stores in the account.
#
# Returns:
#       [[datastore_name, datastore_id, datastore_status]]
#    And:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function imaging_list_datastores() {
  local option OPTARG # Required to use getopts command in a function.
  local error_code
  # bashsupport disable=BP5008
  function usage() {
    echo "function imaging_list_datastores"
    echo "Lists the AWS HealthImaging data stores in the account."
    echo ""
  }

  # Retrieve the calling parameters.
```

```
  while getopts "h" option; do
    case "${option}" in
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local response
  response=$(aws medical-imaging list-datastores \
    --output text \
    --query "datastoreSummaries[*][datastoreName, datastoreId, datastoreStatus]")
  error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports list-datastores operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListDatastores。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

搭配 Bash 指令碼使用 AWS CLI 的 IAM 範例

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Bash 指令碼搭配 IAM 來執行動作和實作常見案例。

基本概念是程式碼範例，這些範例說明如何在服務內執行基本操作。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

案例是向您展示如何呼叫服務中的多個函數或與其他 AWS 服務組合來完成特定任務的程式碼範例。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 基本概念
- 動作
- 案例

基本概念

了解基本概念

下列程式碼範例示範如何建立使用者並擔任角色。

> **Warning**
>
> 為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 AWS IAM Identity Center。

- 建立沒有許可的使用者。
- 建立一個可授予許可的角色，以列出帳戶的 Amazon S3 儲存貯體。
- 新增政策，讓使用者擔任該角色。
- 使用暫時憑證，擔任角色並列出 Amazon S3 儲存貯體，然後清理資源。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function iam_create_user_assume_role
#
# Scenario to create an IAM user, create an IAM role, and apply the role to the
 user.
#
#     "IAM access" permissions are needed to run this code.
#     "STS assume role" permissions are needed to run this code. (Note: It might be
 necessary to
#           create a custom policy).
#
# Returns:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function iam_create_user_assume_role() {
  {
    if [ "$IAM_OPERATIONS_SOURCED" != "True" ]; then

      source ./iam_operations.sh
    fi
  }

  echo_repeat "*" 88
  echo "Welcome to the IAM create user and assume role demo."
  echo
  echo "This demo will create an IAM user, create an IAM role, and apply the role to
 the user."
  echo_repeat "*" 88
  echo

  echo -n "Enter a name for a new IAM user: "
  get_input
  user_name=$get_input_result
```

```
  local user_arn
  user_arn=$(iam_create_user -u "$user_name")

  # shellcheck disable=SC2181
  if [[ ${?} == 0 ]]; then
    echo "Created demo IAM user named $user_name"
  else
    errecho "$user_arn"
    errecho "The user failed to create. This demo will exit."
    return 1
  fi

  local access_key_response
  access_key_response=$(iam_create_user_access_key -u "$user_name")
  # shellcheck disable=SC2181
  if [[ ${?} != 0 ]]; then
    errecho "The access key failed to create. This demo will exit."
    clean_up "$user_name"
    return 1
  fi

  IFS=$'\t ' read -r -a access_key_values <<<"$access_key_response"
  local key_name=${access_key_values[0]}
  local key_secret=${access_key_values[1]}

  echo "Created access key named $key_name"

  echo "Wait 10 seconds for the user to be ready."
  sleep 10
  echo_repeat "*" 88
  echo

  local iam_role_name
  iam_role_name=$(generate_random_name "test-role")
  echo "Creating a role named $iam_role_name with user $user_name as the principal."

  local assume_role_policy_document="{
    \"Version\": \"2012-10-17\",
    \"Statement\": [{
        \"Effect\": \"Allow\",
        \"Principal\": {\"AWS\": \"$user_arn\"},
        \"Action\": \"sts:AssumeRole\"
        }]
```

```
  }"

local role_arn
role_arn=$(iam_create_role -n "$iam_role_name" -p "$assume_role_policy_document")

# shellcheck disable=SC2181
if [ ${?} == 0 ]; then
  echo "Created IAM role named $iam_role_name"
else
  errecho "The role failed to create. This demo will exit."
  clean_up "$user_name" "$key_name"
  return 1
fi

local policy_name
policy_name=$(generate_random_name "test-policy")
local policy_document="{
              \"Version\": \"2012-10-17\",
              \"Statement\": [{
                  \"Effect\": \"Allow\",
                  \"Action\": \"s3:ListAllMyBuckets\",
                  \"Resource\": \"arn:aws:s3:::*\"}]}"

local policy_arn
policy_arn=$(iam_create_policy -n "$policy_name" -p "$policy_document")
# shellcheck disable=SC2181
if [[ ${?} == 0 ]]; then
  echo "Created  IAM policy named $policy_name"
else
  errecho "The policy failed to create."
  clean_up "$user_name" "$key_name" "$iam_role_name"
  return 1
fi

if (iam_attach_role_policy -n "$iam_role_name" -p "$policy_arn"); then
  echo "Attached policy $policy_arn to role $iam_role_name"
else
  errecho "The policy failed to attach."
  clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
  return 1
fi

local assume_role_policy_document="{
              \"Version\": \"2012-10-17\",
```

```
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"sts:AssumeRole\",
                    \"Resource\": \"$role_arn\"}]}"

 local assume_role_policy_name
 assume_role_policy_name=$(generate_random_name "test-assume-role-")

 # shellcheck disable=SC2181
 local assume_role_policy_arn
 assume_role_policy_arn=$(iam_create_policy -n "$assume_role_policy_name" -p
"$assume_role_policy_document")
 # shellcheck disable=SC2181
 if [ ${?} == 0 ]; then
   echo "Created  IAM policy named $assume_role_policy_name for sts assume role"
 else
   errecho "The policy failed to create."
   clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
   return 1
 fi

 echo "Wait 10 seconds to give AWS time to propagate these new resources and
connections."
 sleep 10
 echo_repeat "*" 88
 echo

 echo "Try to list buckets without the new user assuming the role."
 echo_repeat "*" 88
 echo

 # Set the environment variables for the created user.
 # bashsupport disable=BP2001
 export AWS_ACCESS_KEY_ID=$key_name
 # bashsupport disable=BP2001
 export AWS_SECRET_ACCESS_KEY=$key_secret

 local buckets
 buckets=$(s3_list_buckets)

 # shellcheck disable=SC2181
 if [ ${?} == 0 ]; then
   local bucket_count
   bucket_count=$(echo "$buckets" | wc -w | xargs)
```

```
    echo "There are $bucket_count buckets in the account. This should not have
happened."
  else
    errecho "Because the role with permissions has not been assumed, listing buckets
failed."
  fi

  echo
  echo_repeat "*" 88
  echo "Now assume the role $iam_role_name and list the buckets."
  echo_repeat "*" 88
  echo

  local credentials

  credentials=$(sts_assume_role -r "$role_arn" -n "AssumeRoleDemoSession")
  # shellcheck disable=SC2181
  if [ ${?} == 0 ]; then
    echo "Assumed role $iam_role_name"
  else
    errecho "Failed to assume role."
    export AWS_ACCESS_KEY_ID=""
    export AWS_SECRET_ACCESS_KEY=""
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
"$assume_role_policy_arn"
    return 1
  fi

  IFS=$'\t ' read -r -a credentials <<<"$credentials"

  export AWS_ACCESS_KEY_ID=${credentials[0]}
  export AWS_SECRET_ACCESS_KEY=${credentials[1]}
  # bashsupport disable=BP2001
  export AWS_SESSION_TOKEN=${credentials[2]}

  buckets=$(s3_list_buckets)

  # shellcheck disable=SC2181
  if [ ${?} == 0 ]; then
    local bucket_count
    bucket_count=$(echo "$buckets" | wc -w | xargs)
    echo "There are $bucket_count buckets in the account. Listing buckets succeeded
because of "
    echo "the assumed role."
```

```
  else
    errecho "Failed to list buckets. This should not happen."
    export AWS_ACCESS_KEY_ID=""
    export AWS_SECRET_ACCESS_KEY=""
    export AWS_SESSION_TOKEN=""
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
 "$assume_role_policy_arn"
    return 1
  fi

  local result=0
  export AWS_ACCESS_KEY_ID=""
  export AWS_SECRET_ACCESS_KEY=""

  echo
  echo_repeat "*" 88
  echo "The created resources will now be deleted."
  echo_repeat "*" 88
  echo

  clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
 "$assume_role_policy_arn"

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    result=1
  fi

  return $result
}
```

此案例中使用的 IAM 函數。

```
###############################################################################
# function iam_user_exists
#
# This function checks to see if the specified AWS Identity and Access Management
 (IAM) user already exists.
#
# Parameters:
#       $1 - The name of the IAM user to check.
#
```

```
# Returns:
#       0 - If the user already exists.
#       1 - If the user doesn't exist.
###############################################################################
function iam_user_exists() {
  local user_name
  user_name=$1

  # Check whether the IAM user already exists.
  # We suppress all output - we're interested only in the return code.

  local errors
  errors=$(aws iam get-user \
    --user-name "$user_name" 2>&1 >/dev/null)

  local error_code=${?}

  if [[ $error_code -eq 0 ]]; then
    return 0 # 0 in Bash script means true.
  else
    if [[ $errors != *"error"*"(NoSuchEntity)"* ]]; then
      aws_cli_error_log $error_code
      errecho "Error calling iam get-user $errors"
    fi

    return 1 # 1 in Bash script means false.
  fi
}

###############################################################################
# function iam_create_user
#
# This function creates the specified IAM user, unless
# it already exists.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       The ARN of the user.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
```

```
function iam_create_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user"
    echo "Creates an AWS Identity and Access Management (IAM) user. You must supply
 a username:"
    echo "  -u user_name    The name of the user. It must be unique within the
 account."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    User name:   $user_name"
  iecho ""

  # If the user already exists, we don't want to try to create it.
  if (iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name already exists in the account."
```

```
    return 1
  fi

  response=$(aws iam create-user --user-name "$user_name" \
    --output text \
    --query 'User.Arn')

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-user operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function iam_create_user_access_key
#
# This function creates an IAM access key for the specified user.
#
# Parameters:
#       -u user_name -- The name of the IAM user.
#       [-f file_name] -- The optional file name for the access key output.
#
# Returns:
#       [access_key_id access_key_secret]
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_user_access_key() {
  local user_name file_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) key pair."
    echo "  -u user_name   The name of the IAM user."
```

```
  echo "  [-f file_name]   Optional file name for the access key output."
  echo ""
}

# Retrieve the calling parameters.
while getopts "u:f:h" option; do
  case "${option}" in
    u) user_name="${OPTARG}" ;;
    f) file_name="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
  errecho "ERROR: You must provide a username with the -u parameter."
  usage
  return 1
fi

response=$(aws iam create-access-key \
  --user-name "$user_name" \
  --output text)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports create-access-key operation failed.$response"
  return 1
fi

if [[ -n "$file_name" ]]; then
  echo "$response" >"$file_name"
fi
```

```
  local key_id key_secret
  # shellcheck disable=SC2086
  key_id=$(echo $response | cut -f 2 -d ' ')
  # shellcheck disable=SC2086
  key_secret=$(echo $response | cut -f 4 -d ' ')

  echo "$key_id $key_secret"

  return 0
}

###############################################################################
# function iam_create_role
#
# This function creates an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_json -- The assume role policy document.
#
# Returns:
#       The ARN of the role.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_role() {
  local role_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_json -- The assume role policy document."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      p) policy_document="${OPTARG}" ;;
```

```
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam create-role \
    --role-name "$role_name" \
    --assume-role-policy-document "$policy_document" \
    --output text \
    --query Role.Arn)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

```
###############################################################################
# function iam_create_policy
#
# This function creates an IAM policy.
#
# Parameters:
#       -n policy_name -- The name of the IAM policy.
#       -p policy_json -- The policy document.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_policy() {
  local policy_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_policy"
    echo "Creates an AWS Identity and Access Management (IAM) policy."
    echo "  -n policy_name   The name of the IAM policy."
    echo "  -p policy_json -- The policy document."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) policy_name="${OPTARG}" ;;
      p) policy_document="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1
```

```
  if [[ -z "$policy_name" ]]; then
    errecho "ERROR: You must provide a policy name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam create-policy \
    --policy-name "$policy_name" \
    --policy-document "$policy_document" \
    --output text \
    --query Policy.Arn)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"
}

###############################################################################
# function iam_attach_role_policy
#
# This function attaches an IAM policy to a tole.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_attach_role_policy() {
  local role_name policy_arn response
```

```
 local option OPTARG # Required to use getopts command in a function.

 # bashsupport disable=BP5008
 function usage() {
   echo "function iam_attach_role_policy"
   echo "Attaches an AWS Identity and Access Management (IAM) policy to an IAM
role."
   echo "  -n role_name   The name of the IAM role."
   echo "  -p policy_ARN -- The IAM policy document ARN."
   echo ""
 }

 # Retrieve the calling parameters.
 while getopts "n:p:h" option; do
   case "${option}" in
     n) role_name="${OPTARG}" ;;
     p) policy_arn="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$role_name" ]]; then
   errecho "ERROR: You must provide a role name with the -n parameter."
   usage
   return 1
 fi

 if [[ -z "$policy_arn" ]]; then
   errecho "ERROR: You must provide a policy ARN with the -p parameter."
   usage
   return 1
 fi

 response=$(aws iam attach-role-policy \
   --role-name "$role_name" \
```

```
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports attach-role-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function iam_detach_role_policy
#
# This function detaches an IAM policy to a tole.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_detach_role_policy() {
  local role_name policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_detach_role_policy"
    echo "Detaches an AWS Identity and Access Management (IAM) policy to an IAM
 role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_ARN -- The IAM policy document ARN."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
```

```
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      p) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam detach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports detach-role-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

```
###############################################################################
# function iam_delete_policy
#
# This function deletes an IAM policy.
#
# Parameters:
#       -n policy_arn -- The name of the IAM policy arn.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_policy() {
  local policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_policy"
    echo "Deletes an AWS Identity and Access Management (IAM) policy"
    echo "  -n policy_arn -- The name of the IAM policy arn."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy arn with the -n parameter."
```

```
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Policy arn:  $policy_arn"
  iecho ""

  response=$(aws iam delete-policy \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-policy operation failed.\n$response"
    return 1
  fi

  iecho "delete-policy response:$response"
  iecho

  return 0
}

###############################################################################
# function iam_delete_role
#
# This function deletes an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_role() {
  local role_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_role"
```

```
  echo "Deletes an AWS Identity and Access Management (IAM) role"
  echo "  -n role_name -- The name of the IAM role."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:h" option; do
  case "${option}" in
    n) role_name="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

echo "role_name:$role_name"
if [[ -z "$role_name" ]]; then
  errecho "ERROR: You must provide a role name with the -n parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "    Role name:  $role_name"
iecho ""

response=$(aws iam delete-role \
  --role-name "$role_name")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports delete-role operation failed.\n$response"
  return 1
fi
```

```
  iecho "delete-role response:$response"
  iecho

  return 0
}

###############################################################################
# function iam_delete_access_key
#
# This function deletes an IAM access key for the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user.
#       -k access_key -- The access key to delete.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_access_key() {
  local user_name access_key response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_access_key"
    echo "Deletes an AWS Identity and Access Management (IAM) access key for the
 specified IAM user"
    echo "  -u user_name    The name of the user."
    echo "  -k access_key   The access key to delete."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:k:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      k) access_key="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
```

```
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  if [[ -z "$access_key" ]]; then
    errecho "ERROR: You must provide an access key with the -k parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Username:   $user_name"
  iecho "    Access key:   $access_key"
  iecho ""

  response=$(aws iam delete-access-key \
    --user-name "$user_name" \
    --access-key-id "$access_key")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-access-key operation failed.\n$response"
    return 1
  fi

  iecho "delete-access-key response:$response"
  iecho

  return 0
}

###############################################################################
# function iam_delete_user
```

```
#
# This function deletes the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_user"
    echo "Deletes an AWS Identity and Access Management (IAM) user. You must supply
 a username:"
    echo "  -u user_name    The name of the user."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
```

```
  fi

  iecho "Parameters:\n"
  iecho "    User name:   $user_name"
  iecho ""

  # If the user does not exist, we don't want to try to delete it.
  if (! iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name does not exist in the account."
    return 1
  fi

  response=$(aws iam delete-user \
    --user-name "$user_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-user operation failed.$response"
    return 1
  fi

  iecho "delete-user response:$response"
  iecho

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser

- DeleteUserPolicy
- DetachRolePolicy
- PutUserPolicy

動作

`AttachRolePolicy`

以下程式碼範例顯示如何使用 `AttachRolePolicy`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_attach_role_policy
#
# This function attaches an IAM policy to a tole.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_attach_role_policy() {
```

```
 local role_name policy_arn response
 local option OPTARG # Required to use getopts command in a function.

 # bashsupport disable=BP5008
 function usage() {
   echo "function iam_attach_role_policy"
   echo "Attaches an AWS Identity and Access Management (IAM) policy to an IAM
role."
   echo "  -n role_name   The name of the IAM role."
   echo "  -p policy_ARN -- The IAM policy document ARN."
   echo ""
 }

 # Retrieve the calling parameters.
 while getopts "n:p:h" option; do
   case "${option}" in
     n) role_name="${OPTARG}" ;;
     p) policy_arn="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$role_name" ]]; then
   errecho "ERROR: You must provide a role name with the -n parameter."
   usage
   return 1
 fi

 if [[ -z "$policy_arn" ]]; then
   errecho "ERROR: You must provide a policy ARN with the -p parameter."
   usage
   return 1
 fi

 response=$(aws iam attach-role-policy \
```

```
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports attach-role-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AttachRolePolicy。

`CreateAccessKey`

以下程式碼範例顯示如何使用 `CreateAccessKey`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_create_user_access_key
```

```
#
# This function creates an IAM access key for the specified user.
#
# Parameters:
#       -u user_name -- The name of the IAM user.
#       [-f file_name] -- The optional file name for the access key output.
#
# Returns:
#       [access_key_id access_key_secret]
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_user_access_key() {
  local user_name file_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) key pair."
    echo "  -u user_name   The name of the IAM user."
    echo "  [-f file_name]   Optional file name for the access key output."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:f:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      f) file_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1
```

```
  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  response=$(aws iam create-access-key \
    --user-name "$user_name" \
    --output text)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
  fi

  if [[ -n "$file_name" ]]; then
    echo "$response" >"$file_name"
  fi

  local key_id key_secret
  # shellcheck disable=SC2086
  key_id=$(echo $response | cut -f 2 -d ' ')
  # shellcheck disable=SC2086
  key_secret=$(echo $response | cut -f 4 -d ' ')

  echo "$key_id $key_secret"

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateAccessKey。

CreatePolicy

以下程式碼範例顯示如何使用 CreatePolicy。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_create_policy
#
# This function creates an IAM policy.
#
# Parameters:
#       -n policy_name -- The name of the IAM policy.
#       -p policy_json -- The policy document.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function iam_create_policy() {
  local policy_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_policy"
    echo "Creates an AWS Identity and Access Management (IAM) policy."
    echo "  -n policy_name   The name of the IAM policy."
    echo "  -p policy_json -- The policy document."
    echo ""
  }
```

```
# Retrieve the calling parameters.
while getopts "n:p:h" option; do
  case "${option}" in
    n) policy_name="${OPTARG}" ;;
    p) policy_document="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$policy_name" ]]; then
  errecho "ERROR: You must provide a policy name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_document" ]]; then
  errecho "ERROR: You must provide a policy document with the -p parameter."
  usage
  return 1
fi

response=$(aws iam create-policy \
  --policy-name "$policy_name" \
  --policy-document "$policy_document" \
  --output text \
  --query Policy.Arn)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports create-policy operation failed.\n$response"
  return 1
fi
```

```
  echo "$response"
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreatePolicy。

`CreateRole`

以下程式碼範例顯示如何使用 `CreateRole`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function iam_create_role
#
# This function creates an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_json -- The assume role policy document.
#
# Returns:
#       The ARN of the role.
#     And:
#       0 - If successful.
#       1 - If it fails.
```

```
###############################################################################
function iam_create_role() {
  local role_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_json -- The assume role policy document."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      p) policy_document="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
  fi
```

```
  response=$(aws iam create-role \
    --role-name "$role_name" \
    --assume-role-policy-document "$policy_document" \
    --output text \
    --query Role.Arn)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateRole。

`CreateUser`

以下程式碼範例顯示如何使用 `CreateUser`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
##############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
```

```
    echo "$@"
  fi
}

###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_create_user
#
# This function creates the specified IAM user, unless
# it already exists.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       The ARN of the user.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user"
    echo "Creates an AWS Identity and Access Management (IAM) user. You must supply
 a username:"
    echo "  -u user_name    The name of the user. It must be unique within the
 account."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
```

```
  case "${option}" in
    u) user_name="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
  errecho "ERROR: You must provide a username with the -u parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "    User name:   $user_name"
iecho ""

# If the user already exists, we don't want to try to create it.
if (iam_user_exists "$user_name"); then
  errecho "ERROR: A user with that name already exists in the account."
  return 1
fi

response=$(aws iam create-user --user-name "$user_name" \
  --output text \
  --query 'User.Arn')

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports create-user operation failed.$response"
  return 1
fi

echo "$response"
```

```
  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateUser。

`DeleteAccessKey`

以下程式碼範例顯示如何使用 `DeleteAccessKey`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_delete_access_key
#
# This function deletes an IAM access key for the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user.
#       -k access_key -- The access key to delete.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function iam_delete_access_key() {
```

```
  local user_name access_key response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_access_key"
    echo "Deletes an AWS Identity and Access Management (IAM) access key for the
specified IAM user"
    echo "  -u user_name    The name of the user."
    echo "  -k access_key   The access key to delete."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:k:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      k) access_key="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  if [[ -z "$access_key" ]]; then
    errecho "ERROR: You must provide an access key with the -k parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
```

```
  iecho "    Username:   $user_name"
  iecho "    Access key:   $access_key"
  iecho ""

  response=$(aws iam delete-access-key \
    --user-name "$user_name" \
    --access-key-id "$access_key")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-access-key operation failed.\n$response"
    return 1
  fi

  iecho "delete-access-key response:$response"
  iecho

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteAccessKey。

`DeletePolicy`

以下程式碼範例顯示如何使用 `DeletePolicy`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
#####################################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
```

```
###############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_delete_policy
#
# This function deletes an IAM policy.
#
# Parameters:
#       -n policy_arn -- The name of the IAM policy arn.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_policy() {
  local policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_policy"
    echo "Deletes an AWS Identity and Access Management (IAM) policy"
    echo "  -n policy_arn -- The name of the IAM policy arn."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) policy_arn="${OPTARG}" ;;
```

```
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy arn with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Policy arn:  $policy_arn"
  iecho ""

  response=$(aws iam delete-policy \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-policy operation failed.\n$response"
    return 1
  fi

  iecho "delete-policy response:$response"
  iecho

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeletePolicy。

`DeleteRole`

以下程式碼範例顯示如何使用 `DeleteRole`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
###############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_delete_role
#
# This function deletes an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#
# Returns:
#       0 - If successful.
```

```
#       1 - If it fails.
###############################################################################
function iam_delete_role() {
  local role_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_role"
    echo "Deletes an AWS Identity and Access Management (IAM) role"
    echo "  -n role_name -- The name of the IAM role."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  echo "role_name:$role_name"
  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Role name:  $role_name"
  iecho ""

  response=$(aws iam delete-role \
    --role-name "$role_name")
```

```
  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-role operation failed.\n$response"
    return 1
  fi

  iecho "delete-role response:$response"
  iecho

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteRole。

`DeleteUser`

以下程式碼範例顯示如何使用 `DeleteUser`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
##############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}
```

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_delete_user
#
# This function deletes the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_user"
    echo "Deletes an AWS Identity and Access Management (IAM) user. You must supply
 a username:"
    echo "  -u user_name    The name of the user."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
```

```
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    User name:   $user_name"
  iecho ""

  # If the user does not exist, we don't want to try to delete it.
  if (! iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name does not exist in the account."
    return 1
  fi

  response=$(aws iam delete-user \
    --user-name "$user_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-user operation failed.$response"
    return 1
  fi

  iecho "delete-user response:$response"
  iecho

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteUser。

DetachRolePolicy

以下程式碼範例顯示如何使用 DetachRolePolicy。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_detach_role_policy
#
# This function detaches an IAM policy to a tole.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function iam_detach_role_policy() {
  local role_name policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_detach_role_policy"
    echo "Detaches an AWS Identity and Access Management (IAM) policy to an IAM
 role."
```

```
  echo "  -n role_name   The name of the IAM role."
  echo "  -p policy_ARN -- The IAM policy document ARN."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
  case "${option}" in
    n) role_name="${OPTARG}" ;;
    p) policy_arn="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
  errecho "ERROR: You must provide a role name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_arn" ]]; then
  errecho "ERROR: You must provide a policy ARN with the -p parameter."
  usage
  return 1
fi

response=$(aws iam detach-role-policy \
  --role-name "$role_name" \
  --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports detach-role-policy operation failed.\n$response"
```

```
    return 1
  fi

  echo "$response"

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DetachRolePolicy。

GetUser

以下程式碼範例顯示如何使用 GetUser。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_user_exists
#
# This function checks to see if the specified AWS Identity and Access Management
 (IAM) user already exists.
#
# Parameters:
#       $1 - The name of the IAM user to check.
#
# Returns:
```

```
#       0 - If the user already exists.
#       1 - If the user doesn't exist.
###############################################################################
function iam_user_exists() {
  local user_name
  user_name=$1

  # Check whether the IAM user already exists.
  # We suppress all output - we're interested only in the return code.

  local errors
  errors=$(aws iam get-user \
    --user-name "$user_name" 2>&1 >/dev/null)

  local error_code=${?}

  if [[ $error_code -eq 0 ]]; then
    return 0 # 0 in Bash script means true.
  else
    if [[ $errors != *"error"*"(NoSuchEntity)"* ]]; then
      aws_cli_error_log $error_code
      errecho "Error calling iam get-user $errors"
    fi

    return 1 # 1 in Bash script means false.
  fi
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetUser。

`ListAccessKeys`

以下程式碼範例顯示如何使用 `ListAccessKeys`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_list_access_keys
#
# This function lists the access keys for the specified user.
#
# Parameters:
#       -u user_name -- The name of the IAM user.
#
# Returns:
#       access_key_ids
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_list_access_keys() {

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_list_access_keys"
    echo "Lists the AWS Identity and Access Management (IAM) access key IDs for the
 specified user."
    echo "  -u user_name   The name of the IAM user."
    echo ""
  }

  local user_name response
  local option OPTARG # Required to use getopts command in a function.
  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
```

```
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  response=$(aws iam list-access-keys \
    --user-name "$user_name" \
    --output text \
    --query 'AccessKeyMetadata[].AccessKeyId')

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports list-access-keys operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListAccessKeys。

`ListUsers`

以下程式碼範例顯示如何使用 `ListUsers`。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_list_users
#
# List the IAM users in the account.
#
# Returns:
#       The list of users names
#    And:
#       0 - If the user already exists.
#       1 - If the user doesn't exist.
##############################################################################
function iam_list_users() {
  local option OPTARG # Required to use getopts command in a function.
  local error_code
  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_list_users"
    echo "Lists the AWS Identity and Access Management (IAM) user in the account."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "h" option; do
    case "${option}" in
      h)
```

```
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local response

  response=$(aws iam list-users \
    --output text \
    --query "Users[].UserName")
  error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports list-users operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListUsers。

UpdateAccessKey

以下程式碼範例顯示如何使用 UpdateAccessKey。

AWS CLI 使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function iam_update_access_key
#
# This function can activate or deactivate an IAM access key for the specified IAM
 user.
#
# Parameters:
#       -u user_name  -- The name of the user.
#       -k access_key -- The access key to update.
#       -a            -- Activate the selected access key.
#       -d            -- Deactivate the selected access key.
#
# Example:
#       # To deactivate the selected access key for IAM user Bob
#       iam_update_access_key -u Bob -k AKIAIOSFODNN7EXAMPLE -d
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function iam_update_access_key() {
  local user_name access_key status response
  local option OPTARG # Required to use getopts command in a function.
  local activate_flag=false deactivate_flag=false

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_update_access_key"
    echo "Updates the status of an AWS Identity and Access Management (IAM) access
 key for the specified IAM user"
    echo "  -u user_name    The name of the user."
    echo "  -k access_key   The access key to update."
    echo "  -a              Activate the access key."
    echo "  -d              Deactivate the access key."
```

```
    echo ""
  }

  # Retrieve the calling parameters.
    while getopts "u:k:adh" option; do
      case "${option}" in
        u) user_name="${OPTARG}" ;;
        k) access_key="${OPTARG}" ;;
        a) activate_flag=true ;;
        d) deactivate_flag=true ;;
        h)
          usage
          return 0
          ;;
        \?)
          echo "Invalid parameter"
          usage
          return 1
          ;;
      esac
    done
    export OPTIND=1

   # Validate input parameters
    if [[ -z "$user_name" ]]; then
      errecho "ERROR: You must provide a username with the -u parameter."
      usage
      return 1
    fi

    if [[ -z "$access_key" ]]; then
      errecho "ERROR: You must provide an access key with the -k parameter."
      usage
      return 1
    fi

    # Ensure that only -a or -d is specified
    if [[ "$activate_flag" == true && "$deactivate_flag" == true ]]; then
      errecho "ERROR: You cannot specify both -a (activate) and -d (deactivate) at
the same time."
      usage
      return 1
    fi
```

```
    # If neither -a nor -d is provided, return an error
    if [[ "$activate_flag" == false && "$deactivate_flag" == false ]]; then
      errecho "ERROR: You must specify either -a (activate) or -d (deactivate)."
      usage
      return 1
    fi

    # Determine the status based on the flag
    if [[ "$activate_flag" == true ]]; then
      status="Active"
    elif [[ "$deactivate_flag" == true ]]; then
      status="Inactive"
    fi

    iecho "Parameters:\n"
    iecho "    Username:   $user_name"
    iecho "    Access key: $access_key"
    iecho "    New status: $status"
    iecho ""

    # Update the access key status
    response=$(aws iam update-access-key \
      --user-name "$user_name" \
      --access-key-id "$access_key" \
      --status "$status" 2>&1)

    local error_code=${?}

    if [[ $error_code -ne 0 ]]; then
      aws_cli_error_log $error_code
      errecho "ERROR: AWS reports update-access-key operation failed.\n$response"
      return 1
    fi

    iecho "update-access-key response: $response"
    iecho

    return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 UpdateAccessKey。

案例

設定屬性型存取控制

下列程式碼範例示範如何實作 DynamoDB 的屬性型存取控制 (ABAC)。

- 建立 ABAC 的 IAM 政策。
- 為不同部門建立具有標籤的資料表。
- 根據標籤列出和篩選資料表。

AWS CLI 使用 Bash 指令碼

建立 ABAC 的 IAM 政策。

```
# Step 1: Create a policy document for ABAC
cat > abac-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:BatchGetItem",
        "dynamodb:Query",
        "dynamodb:Scan"
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Department": "${aws:PrincipalTag/Department}"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:DeleteItem",
        "dynamodb:BatchWriteItem"
      ],
```

```
      "Resource": "arn:aws:dynamodb:*:*:table/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Department": "${aws:PrincipalTag/Department}",
          "aws:ResourceTag/Environment": "Development"
        }
      }
    }
  ]
}
EOF

# Step 2: Create the IAM policy
aws iam create-policy \
    --policy-name DynamoDBDepartmentBasedAccess \
    --policy-document file://abac-policy.json
```

為不同部門建立具有標籤的資料表。

```
# Create a DynamoDB table with tags for ABAC
aws dynamodb create-table \
    --table-name FinanceData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Department,Value=Finance \
        Key=Environment,Value=Development

# Create another table with different tags
aws dynamodb create-table \
    --table-name MarketingData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Department,Value=Marketing \
        Key=Environment,Value=Production
```

根據標籤列出和篩選資料表。

```
# List all DynamoDB tables
echo "Listing all tables:"
aws dynamodb list-tables

# Get ARNs for all tables
echo -e "\nGetting ARNs for all tables:"
TABLE_ARNS=$(aws dynamodb list-tables --query "TableNames[*]" --output text | xargs
 -I {} aws dynamodb describe-table --table-name {} --query "Table.TableArn" --output
 text)

# For each table ARN, list its tags
echo -e "\nListing tags for each table:"
for ARN in $TABLE_ARNS; do
    TABLE_NAME=$(echo $ARN | awk -F/ '{print $2}')
    echo -e "\nTags for table: $TABLE_NAME"
    aws dynamodb list-tags-of-resource --resource-arn $ARN
done

# Example: Find tables with a specific tag
echo -e "\nFinding tables with Environment=Production tag:"
for ARN in $TABLE_ARNS; do
    TABLE_NAME=$(echo $ARN | awk -F/ '{print $2}')
    TAGS=$(aws dynamodb list-tags-of-resource --resource-arn $ARN --query "Tags[?
Key=='Environment' && Value=='Production']" --output text)
    if [ ! -z "$TAGS" ]; then
        echo "Table with Production tag: $TABLE_NAME"
    fi
done
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - CreatePolicy
  - CreateTable
  - ListTables

使用串流和Time-to-Live

下列程式碼範例示範如何管理 DynamoDB 串流和Time-to-Live功能。

- 建立已啟用串流的資料表。
- 描述串流。
- 建立 Lambda 函數以處理串流。
- 在資料表上啟用 TTL。
- 新增具有 TTL 屬性的項目。
- 描述 TTL 設定。

AWS CLI 使用 Bash 指令碼

建立已啟用串流的資料表。

```
# Create a table with DynamoDB Streams enabled
aws dynamodb create-table \
    --table-name StreamsDemo \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --stream-specification StreamEnabled=true,StreamViewType=NEW_AND_OLD_IMAGES
```

描述串流。

```
# Get information about the stream
aws dynamodb describe-table \
    --table-name StreamsDemo \
    --query "Table.StreamSpecification"

# Get the stream ARN
STREAM_ARN=$(aws dynamodb describe-table \
    --table-name StreamsDemo \
    --query "Table.LatestStreamArn" \
    --output text)

echo "Stream ARN: $STREAM_ARN"
```

```
# Describe the stream
aws dynamodbstreams describe-stream \
    --stream-arn $STREAM_ARN
```

為 Streams 建立 Lambda 函數。

```
# Step 1: Create an IAM role for the Lambda function
cat > trust-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

aws iam create-role \
    --role-name DynamoDBStreamsLambdaRole \
    --assume-role-policy-document file://trust-policy.json

# Step 2: Attach permissions to the role
aws iam attach-role-policy \
    --role-name DynamoDBStreamsLambdaRole \
    --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaDynamoDBExecutionRole

# Step 3: Create a Lambda function (code would be in a separate file)
echo "Lambda function creation would be done separately with appropriate code"

# Step 4: Create an event source mapping
echo "Example command to create event source mapping:"
echo "aws lambda create-event-source-mapping \\"
echo "    --function-name ProcessDynamoDBRecords \\"
echo "    --event-source $STREAM_ARN \\"
echo "    --batch-size 100 \\"
echo "    --starting-position LATEST"
```

在資料表上啟用 TTL。

```
# Create a table for TTL demonstration
aws dynamodb create-table \
    --table-name TTLDemo \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST

# Wait for table to become active
aws dynamodb wait table-exists --table-name TTLDemo

# Enable TTL on the table
aws dynamodb update-time-to-live \
    --table-name TTLDemo \
    --time-to-live-specification "Enabled=true, AttributeName=ExpirationTime"
```

新增具有 TTL 屬性的項目。

```
# Calculate expiration time (current time + 1 day in seconds)
EXPIRATION_TIME=$(date -d "+1 day" +%s)

# Add an item with TTL attribute
aws dynamodb put-item \
    --table-name TTLDemo \
    --item '{
        "ID": {"S": "item1"},
        "Data": {"S": "This item will expire in 1 day"},
        "ExpirationTime": {"N": "'$EXPIRATION_TIME'"}
    }'

# Add an item that expires in 1 hour
EXPIRATION_TIME_HOUR=$(date -d "+1 hour" +%s)
aws dynamodb put-item \
    --table-name TTLDemo \
    --item '{
        "ID": {"S": "item2"},
        "Data": {"S": "This item will expire in 1 hour"},
```

```
        "ExpirationTime": {"N": "'$EXPIRATION_TIME_HOUR'"}
    }'
```

描述 TTL 設定。

```
# Describe TTL settings for a table
aws dynamodb describe-time-to-live \
    --table-name TTLDemo
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - AttachRolePolicy
  - CreateRole
  - CreateTable
  - DescribeTable
  - DescribeTimeToLive
  - PutItem
  - UpdateTimeToLive

AWS KMSAWS CLI 搭配 Bash 指令碼使用 的範例

下列程式碼範例說明如何搭配 AWS Command Line Interface Bash 指令碼使用 來執行動作和實作常見案例 AWS KMS。

案例是向您展示如何呼叫服務中的多個函數或與其他 AWS 服務組合來完成特定任務的程式碼範例。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 案例

案例

使用資料表加密

下列程式碼範例示範如何管理 DynamoDB 資料表的加密選項。

- 建立具有預設加密的資料表。
- 使用客戶受管 CMK 建立資料表。
- 更新資料表加密設定。
- 描述資料表加密。

AWS CLI 搭配 Bash 指令碼

建立具有預設加密的資料表。

```
# Create a table with default encryption (AWS owned key)
aws dynamodb create-table \
    --table-name CustomerData \
    --attribute-definitions \
        AttributeName=CustomerID,AttributeType=S \
    --key-schema \
        AttributeName=CustomerID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --sse-specification Enabled=true,SSEType=KMS
```

使用客戶受管 CMK 建立資料表。

```
# Step 1: Create a customer managed key in KMS
aws kms create-key \
    --description "Key for DynamoDB table encryption" \
    --key-usage ENCRYPT_DECRYPT \
    --customer-master-key-spec SYMMETRIC_DEFAULT

# Store the key ID for later use
KEY_ID=$(aws kms list-keys --query "Keys[?contains(KeyArn, 'Key for
 DynamoDB')].KeyId" --output text)

# Step 2: Create a table with the customer managed key
aws dynamodb create-table \
    --table-name SensitiveData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

更新資料表加密。

```
# Update a table to use a different KMS key
aws dynamodb update-table \
    --table-name CustomerData \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

描述資料表加密。

```
# Describe the table to see encryption settings
aws dynamodb describe-table \
    --table-name CustomerData \
    --query "Table.SSEDescription"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - CreateKey
  - CreateTable
  - DescribeTable
  - UpdateTable

搭配 Bash 指令碼使用 AWS CLI 的 Amazon S3 範例

下列程式碼範例示範如何使用 AWS Command Line Interface 搭配 Bash 指令碼搭配 Amazon S3 來執行動作和實作常見案例。

基本概念是程式碼範例，這些範例說明如何在服務內執行基本操作。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 基本概念
- 動作

基本概念

了解基本概念

以下程式碼範例顯示做法：

- 建立儲存貯體並上傳檔案到該儲存貯體。
- 從儲存貯體下載物件。
- 將物件複製至儲存貯體中的子文件夾。
- 列出儲存貯體中的物件。
- 刪除儲存貯體物件和該儲存貯體。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function s3_getting_started
#
# This function creates, copies, and deletes S3 buckets and objects.
#
# Returns:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function s3_getting_started() {
  {
    if [ "$BUCKET_OPERATIONS_SOURCED" != "True" ]; then
      cd bucket-lifecycle-operations || exit

      source ./bucket_operations.sh
      cd ..
    fi
  }

  echo_repeat "*" 88
```

```
 echo "Welcome to the Amazon S3 getting started demo."
 echo_repeat "*" 88
   echo "A unique bucket will be created by appending a Universally Unique
Identifier to a bucket name prefix."
   echo -n "Enter a prefix for the S3 bucket that will be used in this demo: "
   get_input
   bucket_name_prefix=$get_input_result
 local bucket_name
 bucket_name=$(generate_random_name "$bucket_name_prefix")

 local region_code
 region_code=$(aws configure get region)

 if create_bucket -b "$bucket_name" -r "$region_code"; then
   echo "Created demo bucket named $bucket_name"
 else
   errecho "The bucket failed to create. This demo will exit."
   return 1
 fi

 local file_name
 while [ -z "$file_name" ]; do
   echo -n "Enter a file you want to upload to your bucket: "
   get_input
   file_name=$get_input_result

   if [ ! -f "$file_name" ]; then
     echo "Could not find file $file_name. Are you sure it exists?"
     file_name=""
   fi
 done

 local key
 key="$(basename "$file_name")"

 local result=0
 if copy_file_to_bucket "$bucket_name" "$file_name" "$key"; then
   echo "Uploaded file $file_name into bucket $bucket_name with key $key."
 else
   result=1
 fi

 local destination_file
 destination_file="$file_name.download"
```

```
  if yes_no_input "Would you like to download $key to the file $destination_file?
 (y/n) "; then
    if download_object_from_bucket "$bucket_name" "$destination_file" "$key"; then
      echo "Downloaded $key in the bucket $bucket_name to the file
 $destination_file."
    else
      result=1
    fi
  fi

  if yes_no_input "Would you like to copy $key a new object key in your bucket? (y/
n) "; then
    local to_key
    to_key="demo/$key"
    if copy_item_in_bucket "$bucket_name" "$key" "$to_key"; then
      echo "Copied $key in the bucket $bucket_name to the  $to_key."
    else
      result=1
    fi
  fi

  local bucket_items
  bucket_items=$(list_items_in_bucket "$bucket_name")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    result=1
  fi

  echo "Your bucket contains the following items."
  echo -e "Name\t\tSize"
  echo "$bucket_items"

  if yes_no_input "Delete the bucket, $bucket_name, as well as the objects in it?
 (y/n) "; then
    bucket_items=$(echo "$bucket_items" | cut -f 1)

    if delete_items_in_bucket "$bucket_name" "$bucket_items"; then
      echo "The following items were deleted from the bucket $bucket_name"
      echo "$bucket_items"
    else
      result=1
    fi
```

```
    if delete_bucket "$bucket_name"; then
      echo "Deleted the bucket $bucket_name"
    else
      result=1
    fi
  fi

  return $result
}
```

在此案例中使用的 Amazon S3 函數。

```
###############################################################################
# function create-bucket
#
# This function creates the specified bucket in the specified AWS Region, unless
# it already exists.
#
# Parameters:
#       -b bucket_name  -- The name of the bucket to create.
#       -r region_code  -- The code for an AWS Region in which to
#                          create the bucket.
#
# Returns:
#       The URL of the bucket that was created.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function create_bucket() {
  local bucket_name region_code response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function create_bucket"
    echo "Creates an Amazon S3 bucket. You must supply a bucket name:"
    echo "  -b bucket_name    The name of the bucket. It must be globally unique."
    echo "  [-r region_code]    The code for an AWS Region in which the bucket is
 created."
    echo ""
  }
```

```
  # Retrieve the calling parameters.
  while getopts "b:r:h" option; do
    case "${option}" in
      b) bucket_name="${OPTARG}" ;;
      r) region_code="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done

  if [[ -z "$bucket_name" ]]; then
    errecho "ERROR: You must provide a bucket name with the -b parameter."
    usage
    return 1
  fi

  local bucket_config_arg
  # A location constraint for "us-east-1" returns an error.
  if [[ -n "$region_code" ]] && [[ "$region_code" != "us-east-1" ]]; then
    bucket_config_arg="--create-bucket-configuration LocationConstraint=
$region_code"
  fi

  iecho "Parameters:\n"
  iecho "    Bucket name:   $bucket_name"
  iecho "    Region code:   $region_code"
  iecho ""

  # If the bucket already exists, we don't want to try to create it.
  if (bucket_exists "$bucket_name"); then
    errecho "ERROR: A bucket with that name already exists. Try again."
    return 1
  fi

  # shellcheck disable=SC2086
  response=$(aws s3api create-bucket \
```

```
    --bucket "$bucket_name" \
    $bucket_config_arg)

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports create-bucket operation failed.\n$response"
    return 1
  fi
}

################################################################################
# function copy_file_to_bucket
#
# This function creates a file in the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket to copy the file to.
#       $2 - The path and file name of the local file to copy to the bucket.
#       $3 - The key (name) to call the copy of the file in the bucket.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
################################################################################
function copy_file_to_bucket() {
  local response bucket_name source_file destination_file_name
  bucket_name=$1
  source_file=$2
  destination_file_name=$3

  response=$(aws s3api put-object \
    --bucket "$bucket_name" \
    --body "$source_file" \
    --key "$destination_file_name")

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
  fi
}

################################################################################
# function download_object_from_bucket
```

```
#
# This function downloads an object in a bucket to a file.
#
# Parameters:
#       $1 - The name of the bucket to download the object from.
#       $2 - The path and file name to store the downloaded bucket.
#       $3 - The key (name) of the object in the bucket.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function download_object_from_bucket() {
  local bucket_name=$1
  local destination_file_name=$2
  local object_name=$3
  local response

  response=$(aws s3api get-object \
    --bucket "$bucket_name" \
    --key "$object_name" \
    "$destination_file_name")

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
  fi
}

###############################################################################
# function copy_item_in_bucket
#
# This function creates a copy of the specified file in the same bucket.
#
# Parameters:
#       $1 - The name of the bucket to copy the file from and to.
#       $2 - The key of the source file to copy.
#       $3 - The key of the destination file.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
```

```
function copy_item_in_bucket() {
  local bucket_name=$1
  local source_key=$2
  local destination_key=$3
  local response

  response=$(aws s3api copy-object \
    --bucket "$bucket_name" \
    --copy-source "$bucket_name/$source_key" \
    --key "$destination_key")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR:  AWS reports s3api copy-object operation failed.\n$response"
    return 1
  fi
}

################################################################################
# function list_items_in_bucket
#
# This function displays a list of the files in the bucket with each file's
# size. The function uses the --query parameter to retrieve only the key and
# size fields from the Contents collection.
#
# Parameters:
#       $1 - The name of the bucket.
#
# Returns:
#       The list of files in text format.
#     And:
#       0 - If successful.
#       1 - If it fails.
################################################################################
function list_items_in_bucket() {
  local bucket_name=$1
  local response

  response=$(aws s3api list-objects \
    --bucket "$bucket_name" \
    --output text \
    --query 'Contents[].{Key: Key, Size: Size}')

  # shellcheck disable=SC2181
```

```
  if [[ ${?} -eq 0 ]]; then
    echo "$response"
  else
    errecho "ERROR: AWS reports s3api list-objects operation failed.\n$response"
    return 1
  fi
}

################################################################################
# function delete_items_in_bucket
#
# This function deletes the specified list of keys from the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket.
#       $2 - A list of keys in the bucket to delete.

# Returns:
#       0 - If successful.
#       1 - If it fails.
################################################################################
function delete_items_in_bucket() {
  local bucket_name=$1
  local keys=$2
  local response

  # Create the JSON for the items to delete.
  local delete_items
  delete_items="{\"Objects\":["
  for key in $keys; do
    delete_items="$delete_items{\"Key\": \"$key\"},"
  done
  delete_items=${delete_items%?} # Remove the final comma.
  delete_items="$delete_items]}"

  response=$(aws s3api delete-objects \
    --bucket "$bucket_name" \
    --delete "$delete_items")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR:  AWS reports s3api delete-object operation failed.\n$response"
    return 1
  fi
```

```
}

##############################################################################
# function delete_bucket
#
# This function deletes the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket.

# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function delete_bucket() {
  local bucket_name=$1
  local response

  response=$(aws s3api delete-bucket \
    --bucket "$bucket_name")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR: AWS reports s3api delete-bucket failed.\n$response"
    return 1
  fi
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

動作

`CopyObject`

以下程式碼範例顯示如何使用 `CopyObject`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function copy_item_in_bucket
#
# This function creates a copy of the specified file in the same bucket.
#
# Parameters:
#       $1 - The name of the bucket to copy the file from and to.
#       $2 - The key of the source file to copy.
#       $3 - The key of the destination file.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function copy_item_in_bucket() {
  local bucket_name=$1
  local source_key=$2
  local destination_key=$3
  local response
```

```
  response=$(aws s3api copy-object \
    --bucket "$bucket_name" \
    --copy-source "$bucket_name/$source_key" \
    --key "$destination_key")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR:  AWS reports s3api copy-object operation failed.\n$response"
    return 1
  fi
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CopyObject。

`CreateBucket`

以下程式碼範例顯示如何使用 `CreateBucket`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
#####################################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####################################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

#####################################################################################
# function errecho
#
```

```
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function create-bucket
#
# This function creates the specified bucket in the specified AWS Region, unless
# it already exists.
#
# Parameters:
#       -b bucket_name  -- The name of the bucket to create.
#       -r region_code  -- The code for an AWS Region in which to
#                          create the bucket.
#
# Returns:
#       The URL of the bucket that was created.
#     And:
#       0 - If successful.
#       1 - If it fails.
#############################################################################
function create_bucket() {
  local bucket_name region_code response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function create_bucket"
    echo "Creates an Amazon S3 bucket. You must supply a bucket name:"
    echo "  -b bucket_name    The name of the bucket. It must be globally unique."
    echo "  [-r region_code]    The code for an AWS Region in which the bucket is
 created."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "b:r:h" option; do
    case "${option}" in
      b) bucket_name="${OPTARG}" ;;
      r) region_code="${OPTARG}" ;;
      h)
        usage
```

```
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done

  if [[ -z "$bucket_name" ]]; then
    errecho "ERROR: You must provide a bucket name with the -b parameter."
    usage
    return 1
  fi

  local bucket_config_arg
  # A location constraint for "us-east-1" returns an error.
  if [[ -n "$region_code" ]] && [[ "$region_code" != "us-east-1" ]]; then
    bucket_config_arg="--create-bucket-configuration LocationConstraint=
$region_code"
  fi

  iecho "Parameters:\n"
  iecho "    Bucket name:   $bucket_name"
  iecho "    Region code:   $region_code"
  iecho ""

  # If the bucket already exists, we don't want to try to create it.
  if (bucket_exists "$bucket_name"); then
    errecho "ERROR: A bucket with that name already exists. Try again."
    return 1
  fi

  # shellcheck disable=SC2086
  response=$(aws s3api create-bucket \
    --bucket "$bucket_name" \
    $bucket_config_arg)

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports create-bucket operation failed.\n$response"
    return 1
  fi
```

```
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateBucket。

`DeleteBucket`

以下程式碼範例顯示如何使用 `DeleteBucket`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function delete_bucket
#
# This function deletes the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket.

# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function delete_bucket() {
  local bucket_name=$1
  local response
```

```
  response=$(aws s3api delete-bucket \
    --bucket "$bucket_name")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR: AWS reports s3api delete-bucket failed.\n$response"
    return 1
  fi
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteBucket。

`DeleteObject`

以下程式碼範例顯示如何使用 `DeleteObject`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function delete_item_in_bucket
#
# This function deletes the specified file from the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket.
```

```
#       $2 - The key (file name) in the bucket to delete.

# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function delete_item_in_bucket() {
  local bucket_name=$1
  local key=$2
  local response

  response=$(aws s3api delete-object \
    --bucket "$bucket_name" \
    --key "$key")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR:  AWS reports s3api delete-object operation failed.\n$response"
    return 1
  fi
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteObject。

`DeleteObjects`

以下程式碼範例顯示如何使用 `DeleteObjects`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
```

```
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function delete_items_in_bucket
#
# This function deletes the specified list of keys from the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket.
#       $2 - A list of keys in the bucket to delete.

# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function delete_items_in_bucket() {
  local bucket_name=$1
  local keys=$2
  local response

  # Create the JSON for the items to delete.
  local delete_items
  delete_items="{\"Objects\":["
  for key in $keys; do
    delete_items="$delete_items{\"Key\": \"$key\"},"
  done
  delete_items=${delete_items%?} # Remove the final comma.
  delete_items="$delete_items]}"

  response=$(aws s3api delete-objects \
    --bucket "$bucket_name" \
    --delete "$delete_items")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR:  AWS reports s3api delete-object operation failed.\n$response"
    return 1
  fi
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteObjects。

`GetObject`

以下程式碼範例顯示如何使用 `GetObject`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function download_object_from_bucket
#
# This function downloads an object in a bucket to a file.
#
# Parameters:
#       $1 - The name of the bucket to download the object from.
#       $2 - The path and file name to store the downloaded bucket.
#       $3 - The key (name) of the object in the bucket.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function download_object_from_bucket() {
  local bucket_name=$1
  local destination_file_name=$2
  local object_name=$3
  local response

  response=$(aws s3api get-object \
    --bucket "$bucket_name" \
```

```
    --key "$object_name" \
    "$destination_file_name")

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
  fi
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 GetObject。

`HeadBucket`

以下程式碼範例顯示如何使用 HeadBucket。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
#############################################################################
# function bucket_exists
#
# This function checks to see if the specified bucket already exists.
#
# Parameters:
#       $1 - The name of the bucket to check.
#
# Returns:
#       0 - If the bucket already exists.
#       1 - If the bucket doesn't exist.
#############################################################################
function bucket_exists() {
  local bucket_name
  bucket_name=$1

  # Check whether the bucket already exists.
```

```
  # We suppress all output - we're interested only in the return code.

  if aws s3api head-bucket \
    --bucket "$bucket_name" \
    >/dev/null 2>&1; then
    return 0 # 0 in Bash script means true.
  else
    return 1 # 1 in Bash script means false.
  fi
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 HeadBucket。

`ListObjectsV2`

以下程式碼範例顯示如何使用 `ListObjectsV2`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function list_items_in_bucket
#
# This function displays a list of the files in the bucket with each file's
# size. The function uses the --query parameter to retrieve only the key and
# size fields from the Contents collection.
```

```
#
# Parameters:
#       $1 - The name of the bucket.
#
# Returns:
#       The list of files in text format.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function list_items_in_bucket() {
  local bucket_name=$1
  local response

  response=$(aws s3api list-objects \
    --bucket "$bucket_name" \
    --output text \
    --query 'Contents[].{Key: Key, Size: Size}')

  # shellcheck disable=SC2181
  if [[ ${?} -eq 0 ]]; then
    echo "$response"
  else
    errecho "ERROR: AWS reports s3api list-objects operation failed.\n$response"
    return 1
  fi
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ListObjectsV2。

`PutObject`

以下程式碼範例顯示如何使用 `PutObject`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function copy_file_to_bucket
#
# This function creates a file in the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket to copy the file to.
#       $2 - The path and file name of the local file to copy to the bucket.
#       $3 - The key (name) to call the copy of the file in the bucket.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
#############################################################################
function copy_file_to_bucket() {
  local response bucket_name source_file destination_file_name
  bucket_name=$1
  source_file=$2
  destination_file_name=$3

  response=$(aws s3api put-object \
    --bucket "$bucket_name" \
    --body "$source_file" \
    --key "$destination_file_name")

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
  fi
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 PutObject。

AWS STSAWS CLI 搭配 Bash 指令碼使用 的範例

下列程式碼範例說明如何搭配 AWS Command Line Interface Bash 指令碼使用 來執行動作和實作常見案例 AWS STS。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- 動作

動作

`AssumeRole`

以下程式碼範例顯示如何使用 `AssumeRole`。

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
#############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

#############################################################################
# function errecho
```

```
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function sts_assume_role
#
# This function assumes a role in the AWS account and returns the temporary
#  credentials.
#
# Parameters:
#       -n role_session_name -- The name of the session.
#       -r role_arn -- The ARN of the role to assume.
#
# Returns:
#       [access_key_id, secret_access_key, session_token]
#     And:
#       0 - If successful.
#       1 - If an error occurred.
#############################################################################
function sts_assume_role() {
  local role_session_name role_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function sts_assume_role"
    echo "Assumes a role in the AWS account and returns the temporary credentials:"
    echo "  -n role_session_name -- The name of the session."
    echo "  -r role_arn -- The ARN of the role to assume."
    echo ""
  }

  while getopts n:r:h option; do
    case "${option}" in
      n) role_session_name=${OPTARG} ;;
      r) role_arn=${OPTARG} ;;
      h)
        usage
        return 0
        ;;
```

```
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done

  response=$(aws sts assume-role \
    --role-session-name "$role_session_name" \
    --role-arn "$role_arn" \
    --output text \
    --query "Credentials.[AccessKeyId, SecretAccessKey, SessionToken]")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 AssumeRole。

中的安全性 AWS CLI

的雲端安全 AWS 是最高優先順序。身為 AWS 客戶，您可以從資料中心和網路架構中受益，該架構旨在滿足最安全敏感組織的需求。

安全是 AWS 與您之間的共同責任。共同責任模型將其描述為雲端本身的安全和雲端內部的安全：

- 雲端的安全性 – AWS 負責保護在 Cloud AWS 中執行 AWS 服務的基礎設施。 AWS 也為您提供可安全使用的服務。作為AWS 合規計劃的一部分，第三方稽核人員會定期測試和驗證我們安全的有效性。若要了解適用的合規計劃 AWS Command Line Interface，請參閱AWS 合規計劃的 服務範圍。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件可協助您了解如何在使用 AWS Command Line Interface () 時套用共同責任模型AWS CLI。下列主題說明如何設定 AWS CLI 以符合您的安全與合規目標。您也會了解如何使用 AWS CLI 來協助您監控和保護 AWS 資源。

主題

- 中的資料保護 AWS CLI
- 身分和存取權管理
- 此 AWS 產品或服務的合規驗證
- 此 AWS 產品或服務的彈性
- 此 AWS 產品或服務的基礎設施安全
- 為 強制執行最低版本的 TLS AWS CLI

中的資料保護 AWS CLI

AWS 共同責任模型適用於 中的資料保護 AWS Command Line Interface。如此模型所述， AWS 負責保護執行所有 的全域基礎設施 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 AWS 共同的責任模型和 GDPR 部落格文章。

基於資料保護目的，建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 設定 API 和使用者活動記錄 AWS CloudTrail。如需有關使用 CloudTrail 追蹤擷取 AWS 活動的資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 CloudTrail 追蹤。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在 AWS 透過命令列界面或 API 存取 時需要 FIPS 140-3 驗證的密碼編譯模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-3。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用 AWS CLI 或使用主控台、API AWS CLI或其他 AWS 服務 AWS SDKs 時。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

資料加密

任何安全服務都有一項重要功能，就是當資訊處於非使用中狀態時，就會將資訊加密。

靜態加密

除了代表 AWS 使用者與服務互動所需的登入資料之外， 本身 AWS CLI 不會存放任何客戶資料。

如果您使用 AWS CLI 叫用 AWS 服務，將客戶資料傳輸到本機電腦進行儲存，請參閱該服務使用者指南中的安全與合規章節，以取得如何存放、保護和加密該資料的資訊。

傳輸中加密

根據預設，從執行 AWS CLI AWS 和服務端點的用戶端電腦傳輸的所有資料都會透過 HTTPS/TLS 連線傳送所有內容來加密。

您不須採取任何行動即可啟用 HTTPS/TLS。除非您使用 `--no-verify-ssl` 命令列選項明確地停用個別命令，否則會一直處於啟用狀態。

身分和存取權管理

AWS Identity and Access Management (IAM) 是 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制誰可以進行驗證 （ 登入 ） 和授權 （ 具有許可 ） 來使用 AWS 資源。IAM 是您可以免費使用 AWS 服務 的 。

主題

- 目標對象
- 使用身分驗證
- 使用政策管理存取權
- AWS 服務 如何使用 IAM
- 對 AWS 身分和存取進行故障診斷

目標對象

您使用 AWS Identity and Access Management (IAM) 的方式會有所不同，取決於您在 中執行的工作 AWS。

服務使用者 – 如果您使用 AWS 服務 執行任務，管理員會為您提供所需的登入資料和許可。當您使用更多 AWS 功能來執行工作時，您可能需要額外的許可。了解存取許可的管理方式可協助您向管理員請求正確的許可。如果您無法存取 中的功能 AWS，請參閱 對 AWS 身分和存取進行故障診斷或 AWS 服務 您正在使用的 使用者指南。

服務管理員 – 如果您負責公司 AWS 的資源，您可能擁有 的完整存取權 AWS。您的任務是判斷服務使用者應存取哪些 AWS 功能和資源。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步了解貴公司如何使用 IAM AWS，請參閱您正在使用的 使用者指南 AWS 服務 。

IAM 管理員：如果您是 IAM 管理員，建議您掌握如何撰寫政策以管理 AWS存取權的詳細資訊。若要檢視您可以在 IAM 中使用的以 AWS 身分為基礎的政策範例，請參閱 AWS 服務 您正在使用的 使用者指南。

使用身分驗證

身分驗證是您 AWS 使用身分憑證登入 的方式。您必須以 AWS 帳戶根使用者身分、IAM 使用者身分或擔任 IAM 角色來驗證 （ 登入 AWS)。

您可以使用透過身分來源提供的憑證，以聯合身分 AWS 身分身分登入 。 AWS IAM Identity Center (IAM Identity Center) 使用者、您公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料，都是聯合身分的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使用聯合 AWS 身分存取 時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需登入的詳細資訊 AWS，請參閱AWS 登入 《 使用者指南》中的如何登入您的 AWS 帳戶 。

如果您以 AWS 程式設計方式存取 ， AWS 會提供軟體開發套件 (SDK) 和命令列界面 (CLI)，以使用您的 憑證以密碼編譯方式簽署您的請求。如果您不使用 AWS 工具，則必須自行簽署請求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱《IAM 使用者指南》中的適用於 API 請求的AWS Signature 第 4 版。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如， AWS 建議您使用多重驗證 (MFA) 來提高帳戶的安全性。如需更多資訊，請參閱《AWS IAM Identity Center 使用者指南》中的多重要素驗證和《IAM 使用者指南》中的 IAM 中的AWS 多重要素驗證。

AWS 帳戶 根使用者

當您建立 時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取帳戶中的所有 AWS 服務 和資源。此身分稱為 AWS 帳戶 Theroot 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱 IAM 使用者指南中的需要根使用者憑證的任務。

聯合身分

最佳實務是， 要求人類使用者，包括需要管理員存取權的使用者，使用臨時 AWS 服務 憑證與身分提供者聯合來存取 。

聯合身分是您企業使用者目錄、Web 身分提供者、 AWS Directory Service、Identity Center 目錄或任何使用透過身分來源提供的登入資料 AWS 服務 存取的使用者。當聯合身分存取時 AWS 帳戶，它們會擔任 角色，而角色會提供臨時登入資料。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，也可以連接並同步到您自己的身分來源中的一組使用者 AWS 帳戶 和群組，以便在所有 和應用程式中使用。如需 IAM Identity Center 的詳細資訊，請參閱 AWS IAM Identity Center 使用者指南中的什麼是 IAM Identity Center？。

IAM 使用者和群組

IAM 使用者是 中的身分 AWS 帳戶 ，具有單一人員或應用程式的特定許可。建議您盡可能依賴臨時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 IAM 使用者指南中的為需要長期憑證的使用案例定期輪換存取金鑰。

IAM 群組是一種指定 IAM 使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供臨時憑證。如需更多資訊，請參閱《IAM 使用者指南》中的 IAM 使用者的使用案例。

IAM 角色

IAM 角色是 中具有特定許可 AWS 帳戶 的身分。它類似 IAM 使用者，但不與特定的人員相關聯。若要暫時在 中擔任 IAM 角色 AWS Management Console，您可以從使用者切換至 IAM 角色 （主控台）。您可以透過呼叫 AWS CLI 或 AWS API 操作或使用自訂 URL 來擔任角色。如需使用角色的方法詳細資訊，請參閱《IAM 使用者指南》中的擔任角色的方法。

使用臨時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 — 如需向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱《IAM 使用者指南》中的為第三方身分提供者 (聯合) 建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的許可集。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權：您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的主體) 存取您帳戶的資源。角色是授予跨帳戶存取權的主要方式。不過，對於某些 AWS 服務，您可以將政策直接連接到資源（而不是使用角色做為代理）。如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取。
- 跨服務存取 – 有些 AWS 服務 使用其他 中的功能 AWS 服務。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。

- 轉送存取工作階段 (FAS) – 當您使用 IAM 使用者或角色在其中執行動作時 AWS，您會被視為委託人。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 使用呼叫 的委託人許可 AWS 服務，結合 AWS 服務 請求向下游服務提出請求。只有當服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會提出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。
- 服務角色 – 服務角色是服務擔任的 IAM 角色，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的建立角色以委派許可權給 AWS 服務。
- 服務連結角色 – 服務連結角色是一種連結至 的服務角色。 AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

- 在 Amazon EC2 上執行的應用程式 – 您可以使用 IAM 角色來管理在 EC2 執行個體上執行之應用程式的臨時登入資料，以及提出 AWS CLI 或 AWS API 請求。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並將其提供給其所有應用程式，您可以建立連接至執行個體的執行個體描述檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得臨時憑證。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM 角色來授予許可權給 Amazon EC2 執行個體上執行的應用程式。

使用政策管理存取權

您可以透過建立政策並將其連接到身分或資源 AWS 來控制 AWS 中的存取。政策是 中的物件，當與身分或資源相關聯時， AWS 會定義其許可。當委託人 （使用者、根使用者或角色工作階段） 發出請求時， 會 AWS 評估這些政策。政策中的許可決定是否允許或拒絕請求。大多數政策會以 JSON 文件 AWS 的形式存放在 中。如需 JSON 政策文件結構和內容的詳細資訊，請參閱 IAM 使用者指南中的 JSON 政策概觀。

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該政策的使用者可以從 AWS Management Console AWS CLI、 或 AWS API 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管政策是獨立的政策，您可以連接到 中的多個使用者、群組和角色 AWS 帳戶。受管政策包括 AWS 受管政策和客戶受管政策。如需了解如何在受管政策及內嵌政策之間選擇，請參閱《IAM 使用者指南》中的在受管政策和內嵌政策間選擇。

資源型政策

資源型政策是連接到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。委託人可以包含帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的政策中使用來自 IAM 的 AWS 受管政策。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 AWS WAF和 Amazon VPC 是支援 ACLs的服務範例。如需進一步了解 ACL，請參閱 Amazon Simple Storage Service 開發人員指南中的存取控制清單 (ACL) 概觀。

其他政策類型

AWS 支援其他較不常見的政策類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 `Principal` 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可界限的詳細資訊，請參閱 IAM 使用者指南中的 IAM 實體許可界限。
- 服務控制政策 SCPs) – SCPs是 JSON 政策，可指定 中組織或組織單位 (OU) 的最大許可 AWS Organizations。 AWS Organizations 是一種用於分組和集中管理您企業擁有 AWS 帳戶 之多個 的服

務。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 會限制成員帳戶中實體的許可，包括每個實體 AWS 帳戶根使用者。如需 Organizations 和 SCP 的詳細資訊，請參閱《AWS Organizations 使用者指南》中的服務控制政策。

- 資源控制政策 (RCP) - RCP 是 JSON 政策，可用來設定您帳戶中資源的可用許可上限，採取這種方式就不需要更新附加至您所擁有的每個資源的 IAM 政策。RCP 會限制成員帳戶中資源的許可，並可能影響身分的有效許可，包括 AWS 帳戶根使用者，無論它們是否屬於您的組織。如需 Organizations 和 RCPs的詳細資訊，包括 AWS 服務 支援 RCPs 清單，請參閱AWS Organizations 《 使用者指南》中的資源控制政策 (RCPs)。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過撰寫程式的方式建立角色或聯合使用者的暫時工作階段時，做為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需詳細資訊，請參閱 IAM 使用者指南中的工作階段政策。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。若要了解如何 AWS 決定是否在涉及多個政策類型時允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

AWS 服務 如何使用 IAM

若要深入了解 如何使用 AWS 服務 大多數 IAM 功能，請參閱《IAM 使用者指南》中的與 IAM AWS 搭配使用的 服務。

若要了解如何 AWS 服務 搭配 IAM 使用特定 ，請參閱相關服務使用者指南的安全章節。

對 AWS 身分和存取進行故障診斷

使用以下資訊來協助您診斷和修正使用 AWS 和 IAM 時可能遇到的常見問題。

主題

- 我無權在 中執行動作 AWS
- 我未獲得執行 iam:PassRole 的授權
- 我想要允許 以外的人員 AWS 帳戶 存取我的 AWS 資源

我無權在 中執行動作 AWS

如果您收到錯誤，告知您未獲授權執行動作，您的政策必須更新，允許您執行動作。

下列範例錯誤會在mateojackson IAM 使用者嘗試使用主控台檢視一個虛構 *my-example-widget* 資源的詳細資訊，但卻無虛構 awes:*GetWidget* 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 awes:GetWidget on resource: my-example-widget
```

在此情況下，必須更新 mateojackson 使用者的政策，允許使用 awes:*GetWidget* 動作存取 *my-example-widget* 資源。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我未獲得執行 iam:PassRole 的授權

如果您收到錯誤，告知您未獲授權執行 iam:PassRole 動作，您的政策必須更新，允許您將角色傳遞給 AWS。

有些 AWS 服務 可讓您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

名為 marymajor 的 IAM 使用者嘗試使用主控台在 AWS中執行動作時，發生下列範例錯誤。但是，動作要求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 iam:PassRole 動作。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我想要允許 以外的人員 AWS 帳戶 存取我的 AWS 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要了解 是否 AWS 支援這些功能，請參閱 AWS 服務 如何使用 IAM。
- 若要了解如何 AWS 帳戶 在您擁有的 資源之間提供存取權，請參閱《IAM 使用者指南》中的在您擁有 AWS 帳戶 的另一個 中提供存取權給 IAM 使用者。

- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱《IAM 使用者指南》中的將存取權提供給第三方 AWS 帳戶 擁有。
- 如需了解如何透過聯合身分提供存取權，請參閱 IAM 使用者指南中的將存取權提供給在外部進行身分驗證的使用者 (聯合身分)。
- 如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取。

此 AWS 產品或服務的合規驗證

若要了解 是否 AWS 服務 在特定合規計劃的範圍內，請參閱AWS 服務 合規計劃範圍內然後選擇您感興趣的合規計劃。如需一般資訊，請參閱 AWS Compliance Programs。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載報告 in AWS Artifact

您使用 時的合規責任 AWS 服務 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。AWS 提供下列資源來協助合規：

- 安全合規與治理 - 這些解決方案實作指南內容討論了架構考量，並提供部署安全與合規功能的步驟。
- HIPAA 合格服務參考 – 列出 HIPAA 合格服務。並非所有 AWS 服務 都符合 HIPAA 資格。
- AWS 合規資源 – 此工作手冊和指南集合可能適用於您的產業和位置。
- AWS 客戶合規指南 – 透過合規的角度了解共同責任模型。本指南摘要說明跨多個架構 （包括國家標準技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO)) 保護 AWS 服務和映射指南至安全控制的最佳實務。
- 《 AWS Config 開發人員指南》中的使用 規則評估資源 – AWS Config 服務會評估資源組態符合內部實務、產業準則和法規的程度。
- AWS Security Hub – 這 AWS 服務 可讓您全面檢視其中的安全狀態 AWS。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱「Security Hub 控制參考」。
- Amazon GuardDuty – 這會監控您的環境是否有可疑和惡意活動，以 AWS 服務 偵測對您 AWS 帳戶、工作負載、容器和資料的潛在威脅。GuardDuty 可滿足特定合規架構所規定的入侵偵測需求，以協助您因應 PCI DSS 等各種不同的合規需求。
- AWS Audit Manager – 這 AWS 服務 可協助您持續稽核 AWS 用量，以簡化您管理風險的方式，以及符合法規和業界標準的方式。

此 AWS 產品或服務會透過其支援的特定 Amazon Web Services (AWS) 服務，遵循共同責任模型。如需 AWS 服務安全資訊，請參閱AWS 服務安全文件頁面AWS ，以及合規計劃在 AWS 合規工作範圍內的服務。

此 AWS 產品或服務的彈性

AWS 全球基礎設施是以 AWS 區域 和可用區域為基礎建置。

AWS 區域 提供多個實體分隔和隔離的可用區域，這些可用區域與低延遲、高輸送量和高備援聯網連接。

透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域和可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

此 AWS 產品或服務會透過其支援的特定 Amazon Web Services (AWS) 服務，遵循共同責任模型。如需 AWS 服務安全資訊，請參閱AWS 服務安全文件頁面，以及AWS 合規計劃在 AWS 合規工作範圍內的服務。

此 AWS 產品或服務的基礎設施安全

此 AWS 產品或服務使用 受管服務，因此受到 全球網路安全的 AWS 保護。如需 AWS 安全服務以及如何 AWS 保護基礎設施的資訊，請參閱AWS 雲端安全。若要使用基礎設施安全的最佳實務來設計您的 AWS 環境，請參閱安全支柱 AWS Well-Architected Framework 中的基礎設施保護。

您可以使用 AWS 發佈的 API 呼叫，透過網路存取此 AWS 產品或服務。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 AWS Security Token Service (AWS STS) 來產生暫時安全憑證來簽署請求。

此 AWS 產品或服務會透過其支援的特定 Amazon Web Services (AWS) 服務，遵循共同責任模型。如需 AWS 服務安全資訊，請參閱AWS 服務安全文件頁面，以及AWS 合規計劃在 AWS 合規工作範圍內的服務。

為 強制執行最低版本的 TLS AWS CLI

使用 AWS Command Line Interface (AWS CLI) 時，Transport Layer Security (TLS) 通訊協定在保護 AWS CLI 和 之間的通訊方面扮演重要角色 AWS 服務。若要在與 AWS 服務通訊時增加安全性，您應該使用 TLS 1.2 或更新版本。

AWS CLI 第 2 版使用內部 Python 指令碼，當服務與之交談時，該指令碼會編譯為使用最低 TLS 1.2。只要您使用 第 2 版 AWS CLI，就不需要進一步的步驟即可強制執行此最小值。為確保安全性提高，請務必更新至最新版本的 AWS CLI。

AWS CLI 和 AWS 服務 可以使用提供加密、身分驗證和資料完整性的 TLS 通訊協定安全地交換資料。透過利用 TLS AWS CLI 通訊協定， 可確保您與 的互動 AWS 服務 不受未經授權的存取和資料洩露影響，從而增強 AWS 生態系統的整體安全性。

AWS 共同的責任模型適用於 中的資料保護 AWS Command Line Interface。如此模型所述， AWS 負責保護執行所有 的全域基礎設施 AWS 服務。您負責維護在此基礎設施上託管內容的控制權。您也必須負責 AWS 服務 使用 的安全組態和管理任務。如需資料保護的詳細資訊，請參閱 the section called “資料保護”。

故障診斷 的錯誤 AWS CLI

本節介紹常見錯誤和解決問題時應遵循的故障診斷步驟。我們建議先遵循一般故障診斷。

內容

- 先嘗試的一般故障診斷
  - 檢查您的 AWS CLI 命令格式
  - 檢查您的 AWS 區域AWS CLI 命令是否正在使用
  - 確認您執行的是最新版本的 AWS CLI
  - 設定 --debug 選項
  - 啟用和檢閱 AWS CLI 命令歷史記錄日誌
  - 確認您的 AWS CLI 已設定
- 命令未找到錯誤
- 「aws --version」命令傳回的版本與您安裝的版本不同
- 解除安裝 後，「aws --version」命令會傳回版本 AWS CLI
- 已 AWS CLI 處理參數名稱不完整的命令
- 存取遭拒錯誤
- 無效的憑證和金鑰錯誤
- 簽章不相符錯誤
- SSL 憑證錯誤
- JSON 無效錯誤
- 其他資源

先嘗試的一般故障診斷

如果您收到錯誤或遇到 問題 AWS CLI，我們建議您使用下列一般提示來協助您進行故障診斷。

回到頁首

檢查您的 AWS CLI 命令格式

如果您收到錯誤，指出命令不存在，或無法辨識文件表示可用的參數 (`Parameter validation failed`)，則您的命令格式可能不正確。我們建議您檢查以下內容：

- 檢查您的命令是否有拼寫和格式錯誤。
- 確認所有的引號和逸出適合您的終端機在您的命令中是正確的。
- 產生 AWS CLI 骨架以確認您的命令結構。
- 對於 JSON，請另參閱 JSON 值故障診斷。如果您在終端處理 JSON 格式時遇到問題，我們建議透過使用 Blob 將 JSON 資料直接傳遞至 AWS CLI，以跳過終端機的引用規則。

如需如何建構特定命令的詳細資訊，請參閱第 AWS CLI 2 版參考指南。

回到頁首

檢查您的 AWS 區域AWS CLI 命令是否正在使用

> **Note**
>
> 使用 AWS 區域 時 AWS CLI，您必須明確或透過設定預設區域來指定 。如需 AWS 區域 您可以指定之所有 的清單，請參閱《》中的AWS 區域和端點Amazon Web Services 一般參考。所使用的 AWS 區域 指標 AWS CLI 與您在 AWS Management Console URLs和服務端點中看到的名稱相同。

如果您指定的 AWS 服務 無法使用，或您的 資源位於不同的 中，則可能會發生錯誤 AWS 區域 或非預期的結果 AWS 區域。依優先順序， AWS 區域 會以下列方式設定：

- `--region` 命令列選項。
- 開發套件相容的`AWS_REGION`環境變數。
- `AWS_DEFAULT_REGION` 環境變數。
- `region` 設定檔設定。

確認您使用 AWS 區域 正確的 資源。

回到頁首

確認您執行的是最新版本的 AWS CLI

如果您收到錯誤，指出命令不存在，或無法辨識第 AWS CLI 2 版參考指南指出可用的參數，請先確認您的命令格式正確。如果格式正確，我們建議您升級到最新版本的 AWS CLI。更新版本的 AWS CLI

幾乎會在每個工作天發行。新的 AWS 服務、功能和參數會在這些新版本的 中推出 AWS CLI。要存取這些新服務、功能或參數的唯一方法，就是升級到該元素首次推出之後所發行的版本。

如何更新 的版本 AWS CLI ，取決於您最初安裝的方式，如 中所述the section called “安裝/更新”。

如果您使用其中一個隨附的安裝程式，您可能需要移除現有的安裝，然後下載並安裝適用於您作業系統的最新版本安裝程式。

回到頁首

設定 --debug 選項

當 AWS CLI 回報您未立即了解的錯誤，或產生您未預期的結果時，您可以使用 --debug選項再次執行 命令，以取得有關錯誤的詳細資訊。使用此選項時， AWS CLI 會輸出有關處理命令所需的每個步驟的詳細資訊。輸出中的詳細資訊可協助您判斷錯誤發生在哪個步驟，以及可提供觸發錯誤的線索內容。

您可以將輸出傳送到文字檔案，以供日後檢閱，或在收到要求時將其傳送給 AWS 支援 。

當您包含 --debug 選項時，所包括的部分詳細資訊為：

- 正在尋找的憑證
- 正在剖析所提供的參數
- 建構傳送至 AWS 伺服器的請求
- 傳送至 的請求內容 AWS
- 原始回應的內容
- 格式化的輸出

以下是使用和不使用 --debug 選項執行命令的範例。

```
$ aws iam list-groups --profile MyTestProfile
{
    "Groups": [
        {
            "Path": "/",
            "GroupName": "MyTestGroup",
            "GroupId": "AGPA0123456789EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/MyTestGroup",
            "CreateDate": "2019-08-12T19:34:04Z"
        }
```

```
    ]
}
```

```
$ aws iam list-groups --profile MyTestProfile --debug
2019-08-12 12:36:18,305 - MainThread - awscli.clidriver - DEBUG - CLI version: aws-
cli/1.16.215 Python/3.7.3 Linux/4.14.133-113.105.amzn2.x86_64 botocore/1.12.205
2019-08-12 12:36:18,305 - MainThread - awscli.clidriver - DEBUG - Arguments entered to
 CLI: ['iam', 'list-groups', '--debug']
2019-08-12 12:36:18,305 - MainThread - botocore.hooks - DEBUG - Event session-
initialized: calling handler <function add_scalar_parsers at 0x7fdf173161e0>
2019-08-12 12:36:18,305 - MainThread - botocore.hooks - DEBUG - Event session-
initialized: calling handler <function register_uri_param_handler at 0x7fdf17dec400>
2019-08-12 12:36:18,305 - MainThread - botocore.hooks - DEBUG - Event session-
initialized: calling handler <function inject_assume_role_provider_cache at
 0x7fdf17da9378>
2019-08-12 12:36:18,307 - MainThread - botocore.credentials - DEBUG - Skipping
 environment variable credential check because profile name was explicitly set.
2019-08-12 12:36:18,307 - MainThread - botocore.hooks - DEBUG - Event session-
initialized: calling handler <function attach_history_handler at 0x7fdf173ed9d8>
2019-08-12 12:36:18,308 - MainThread - botocore.loaders - DEBUG - Loading JSON
 file: /home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/iam/2010-05-08/
service-2.json
2019-08-12 12:36:18,317 - MainThread - botocore.hooks - DEBUG - Event building-command-
table.iam: calling handler <function add_waiters at 0x7fdf1731a840>
2019-08-12 12:36:18,320 - MainThread - botocore.loaders - DEBUG - Loading JSON
 file: /home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/iam/2010-05-08/
waiters-2.json
2019-08-12 12:36:18,321 - MainThread - awscli.clidriver - DEBUG - OrderedDict([('path-
prefix', <awscli.arguments.CLIArgument object at 0x7fdf171ac780>), ('marker',
 <awscli.arguments.CLIArgument object at 0x7fdf171b09e8>), ('max-items',
 <awscli.arguments.CLIArgument object at 0x7fdf171b09b0>)])
2019-08-12 12:36:18,322 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function add_streaming_output_arg at
 0x7fdf17316510>
2019-08-12 12:36:18,322 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function add_cli_input_json at
 0x7fdf17da9d90>
2019-08-12 12:36:18,322 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function unify_paging_params at
 0x7fdf17328048>
2019-08-12 12:36:18,326 - MainThread - botocore.loaders - DEBUG - Loading JSON
 file: /home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/iam/2010-05-08/
paginators-1.json
```

```
2019-08-12 12:36:18,326 - MainThread - awscli.customizations.paginate - DEBUG -
 Modifying paging parameters for operation: ListGroups
2019-08-12 12:36:18,326 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function add_generate_skeleton at
 0x7fdf1737eae8>
2019-08-12 12:36:18,326 - MainThread - botocore.hooks - DEBUG - Event
 before-building-argument-table-parser.iam.list-groups: calling handler
 <bound method OverrideRequiredArgsArgument.override_required_args of
 <awscli.customizations.cliinputjson.CliInputJSONArgument object at 0x7fdf171b0a58>>
2019-08-12 12:36:18,327 - MainThread - botocore.hooks - DEBUG - Event
 before-building-argument-table-parser.iam.list-groups: calling handler
 <bound method GenerateCliSkeletonArgument.override_required_args of
 <awscli.customizations.generatecliskeleton.GenerateCliSkeletonArgument object at
 0x7fdf171c5978>>
2019-08-12 12:36:18,327 - MainThread - botocore.hooks - DEBUG - Event operation-
args-parsed.iam.list-groups: calling handler functools.partial(<function
 check_should_enable_pagination at 0x7fdf17328158>, ['marker', 'max-items'], {'max-
items': <awscli.arguments.CLIArgument object at 0x7fdf171b09b0>}, OrderedDict([('path-
prefix', <awscli.arguments.CLIArgument object at 0x7fdf171ac780>), ('marker',
 <awscli.arguments.CLIArgument object at 0x7fdf171b09e8>), ('max-items',
 <awscli.customizations.paginate.PageArgument object at 0x7fdf171c58d0>), ('cli-
input-json', <awscli.customizations.cliinputjson.CliInputJSONArgument object at
 0x7fdf171b0a58>), ('starting-token', <awscli.customizations.paginate.PageArgument
 object at 0x7fdf171b0a20>), ('page-size', <awscli.customizations.paginate.PageArgument
 object at 0x7fdf171c5828>), ('generate-cli-skeleton',
 <awscli.customizations.generatecliskeleton.GenerateCliSkeletonArgument object at
 0x7fdf171c5978>)]))
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-
arg.iam.list-groups.path-prefix: calling handler <awscli.paramfile.URIArgumentHandler
 object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-
arg.iam.list-groups.marker: calling handler <awscli.paramfile.URIArgumentHandler object
 at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-
arg.iam.list-groups.max-items: calling handler <awscli.paramfile.URIArgumentHandler
 object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG -
 Event load-cli-arg.iam.list-groups.cli-input-json: calling handler
 <awscli.paramfile.URIArgumentHandler object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG -
 Event load-cli-arg.iam.list-groups.starting-token: calling handler
 <awscli.paramfile.URIArgumentHandler object at 0x7fdf1725c978>
```

```
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-
arg.iam.list-groups.page-size: calling handler <awscli.paramfile.URIArgumentHandler
 object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event
 load-cli-arg.iam.list-groups.generate-cli-skeleton: calling handler
 <awscli.paramfile.URIArgumentHandler object at 0x7fdf1725c978>
2019-08-12 12:36:18,329 - MainThread - botocore.hooks - DEBUG
 - Event calling-command.iam.list-groups: calling handler
 <bound method CliInputJSONArgument.add_to_call_parameters of
 <awscli.customizations.cliinputjson.CliInputJSONArgument object at 0x7fdf171b0a58>>
2019-08-12 12:36:18,329 - MainThread - botocore.hooks - DEBUG -
 Event calling-command.iam.list-groups: calling handler <bound
 method GenerateCliSkeletonArgument.generate_json_skeleton of
 <awscli.customizations.generatecliskeleton.GenerateCliSkeletonArgument object at
 0x7fdf171c5978>>
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - DEBUG - Looking for
 credentials via: assume-role
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - DEBUG - Looking for
 credentials via: assume-role-with-web-identity
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - DEBUG - Looking for
 credentials via: shared-credentials-file
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - INFO - Found credentials
 in shared credentials file: ~/.aws/credentials
2019-08-12 12:36:18,330 - MainThread - botocore.loaders - DEBUG - Loading JSON file: /
home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/endpoints.json
2019-08-12 12:36:18,334 - MainThread - botocore.hooks - DEBUG - Event choose-service-
name: calling handler <function handle_service_name_alias at 0x7fdf1898eb70>
2019-08-12 12:36:18,337 - MainThread - botocore.hooks - DEBUG - Event creating-client-
class.iam: calling handler <function add_generate_presigned_url at 0x7fdf18a028c8>
2019-08-12 12:36:18,337 - MainThread - botocore.regions - DEBUG - Using partition
 endpoint for iam, us-west-2: aws-global
2019-08-12 12:36:18,337 - MainThread - botocore.args - DEBUG - The s3 config key is not
 a dictionary type, ignoring its value of: None
2019-08-12 12:36:18,340 - MainThread - botocore.endpoint - DEBUG - Setting iam timeout
 as (60, 60)
2019-08-12 12:36:18,341 - MainThread - botocore.loaders - DEBUG - Loading JSON file: /
home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/_retry.json
2019-08-12 12:36:18,341 - MainThread - botocore.client - DEBUG - Registering retry
 handlers for service: iam
2019-08-12 12:36:18,342 - MainThread - botocore.hooks - DEBUG - Event before-
parameter-build.iam.ListGroups: calling handler <function generate_idempotent_uuid at
 0x7fdf189b10d0>
```

```
2019-08-12 12:36:18,342 - MainThread - botocore.hooks - DEBUG - Event before-
call.iam.ListGroups: calling handler <function inject_api_version_header_if_needed at
 0x7fdf189b2a60>
2019-08-12 12:36:18,343 - MainThread - botocore.endpoint - DEBUG - Making
 request for OperationModel(name=ListGroups) with params: {'url_path': '/',
 'query_string': '', 'method': 'POST', 'headers': {'Content-Type': 'application/x-
www-form-urlencoded; charset=utf-8', 'User-Agent': 'aws-cli/1.16.215 Python/3.7.3
 Linux/4.14.133-113.105.amzn2.x86_64 botocore/1.12.205'}, 'body': {'Action':
 'ListGroups', 'Version': '2010-05-08'}, 'url': 'https://iam.amazonaws.com/',
 'context': {'client_region': 'aws-global', 'client_config': <botocore.config.Config
 object at 0x7fdf16e9a4a8>, 'has_streaming_input': False, 'auth_type': None}}
2019-08-12 12:36:18,343 - MainThread - botocore.hooks - DEBUG - Event request-
created.iam.ListGroups: calling handler <bound method RequestSigner.handler of
 <botocore.signers.RequestSigner object at 0x7fdf16e9a470>>
2019-08-12 12:36:18,343 - MainThread - botocore.hooks - DEBUG - Event choose-
signer.iam.ListGroups: calling handler <function set_operation_specific_signer at
 0x7fdf18996f28>
2019-08-12 12:36:18,343 - MainThread - botocore.auth - DEBUG - Calculating signature
 using v4 auth.
2019-08-12 12:36:18,343 - MainThread - botocore.auth - DEBUG - CanonicalRequest:
POST
/

content-type:application/x-www-form-urlencoded; charset=utf-8
host:iam.amazonaws.com
x-amz-date:20190812T193618Z

content-type;host;x-amz-date
5f776d91EXAMPLE9b8cb5eb5d6d4a787a33ae41c8cd6eEXAMPLEca69080e1e1f
2019-08-12 12:36:18,344 - MainThread - botocore.auth - DEBUG - StringToSign:
AWS4-HMAC-SHA256
20190812T193618Z
20190812/us-east-1/iam/aws4_request
ab7e367eEXAMPLE2769f178ea509978cf8bfa054874b3EXAMPLE8d043fab6cc9
2019-08-12 12:36:18,344 - MainThread - botocore.auth - DEBUG - Signature:
d85a0EXAMPLEb40164f2f539cdc76d4f294fe822EXAMPLE18ad1ddf58a1a3ce7
2019-08-12 12:36:18,344 - MainThread - botocore.endpoint - DEBUG - Sending
 http request: <AWSPreparedRequest stream_output=False, method=POST,
 url=https://iam.amazonaws.com/, headers={'Content-Type': b'application/
x-www-form-urlencoded; charset=utf-8', 'User-Agent': b'aws-cli/1.16.215
 Python/3.7.3 Linux/4.14.133-113.105.amzn2.x86_64 botocore/1.12.205',
 'X-Amz-Date': b'20190812T193618Z', 'Authorization': b'AWS4-HMAC-SHA256
 Credential=AKIA01234567890EXAMPLE-east-1/iam/aws4_request, SignedHeaders=content-
```

```
type;host;x-amz-date, Signature=d85a07692aceb401EXAMPLEa1b18ad1ddf58a1a3ce7EXAMPLE',
 'Content-Length': '36'}>
2019-08-12 12:36:18,344 - MainThread - urllib3.util.retry - DEBUG - Converted retries
 value: False -> Retry(total=False, connect=None, read=None, redirect=0, status=None)
2019-08-12 12:36:18,344 - MainThread - urllib3.connectionpool - DEBUG - Starting new
 HTTPS connection (1): iam.amazonaws.com:443
2019-08-12 12:36:18,664 - MainThread - urllib3.connectionpool - DEBUG - https://
iam.amazonaws.com:443 "POST / HTTP/1.1" 200 570
2019-08-12 12:36:18,664 - MainThread - botocore.parsers - DEBUG - Response headers:
 {'x-amzn-RequestId': '74c11606-bd38-11e9-9c82-559da0adb349', 'Content-Type': 'text/
xml', 'Content-Length': '570', 'Date': 'Mon, 12 Aug 2019 19:36:18 GMT'}
2019-08-12 12:36:18,664 - MainThread - botocore.parsers - DEBUG - Response body:
b'<ListGroupsResponse xmlns="https://iam.amazonaws.com/doc/2010-05-08/">\n
 <ListGroupsResult>\n    <IsTruncated>false</IsTruncated>\n    <Groups>\n
  <member>\n        <Path>/</Path>\n        <GroupName>MyTestGroup</GroupName>
\n        <Arn>arn:aws:iam::123456789012:group/MyTestGroup</Arn>\n
 <GroupId>AGPA1234567890EXAMPLE</GroupId>\n        <CreateDate>2019-08-12T19:34:04Z</
CreateDate>\n      </member>\n    </Groups>\n  </ListGroupsResult>\n
 <ResponseMetadata>\n    <RequestId>74c11606-bd38-11e9-9c82-559da0adb349</RequestId>\n
 </ResponseMetadata>\n</ListGroupsResponse>\n'
2019-08-12 12:36:18,665 - MainThread - botocore.hooks - DEBUG - Event needs-
retry.iam.ListGroups: calling handler <botocore.retryhandler.RetryHandler object at
 0x7fdf16e9a780>
2019-08-12 12:36:18,665 - MainThread - botocore.retryhandler - DEBUG - No retry needed.
2019-08-12 12:36:18,665 - MainThread - botocore.hooks - DEBUG - Event after-
call.iam.ListGroups: calling handler <function json_decode_policies at 0x7fdf189b1d90>
{
    "Groups": [
        {
            "Path": "/",
            "GroupName": "MyTestGroup",
            "GroupId": "AGPA123456789012EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/MyTestGroup",
            "CreateDate": "2019-08-12T19:34:04Z"
        }
    ]
}
```

回到頁首

啟用和檢閱 AWS CLI 命令歷史記錄日誌

您可以使用 `cli_history` 檔案設定來啟用 AWS CLI 命令歷史記錄日誌。啟用此設定後， 會 AWS CLI 記錄 `aws` 命令的歷史記錄。

您可以使用 `aws history list` 命令來列出歷史記錄，並且在 `aws history show` 命令中使用產生的 command_ids 取得詳細資訊。如需詳細資訊，請參閱 AWS CLI 參考指南中的 `aws history`。

當您包含 `--debug` 選項時，所包括的部分詳細資訊為：

- 對 botocore 執行的 API 呼叫
- 狀態碼
- HTTP 回應
- 標頭
- 傳回碼

您可以使用此資訊來確認參數資料和 API 呼叫以您預期的方式運作，然後推斷出您的命令在過程中的哪個步驟失敗。

回到頁首

確認您的 AWS CLI 已設定

如果您的 `config` 和 `credentials` 檔案或您的 IAM 使用者或角色未正確配置，可能會出現各種錯誤。有關解決 `config` 和 `credentials` 檔案或您的 IAM 使用者或角色等錯誤的詳細資訊，請參閱 the section called “存取遭拒錯誤” 和 the section called “無效的憑證和金鑰錯誤”。

回到頁首

命令未找到錯誤

此錯誤表示作業系統找不到 AWS CLI 命令。安裝可能不完整或需要更新。

可能原因：您嘗試使用比已安裝版本更新的 AWS CLI 功能，或格式不正確

範例錯誤文字：

```
$ aws s3 copy
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:
```

```
  aws help
  aws <command> help
  aws <command> <subcommand> help
aws: error: argument subcommand: Invalid choice, valid choices are:

ls                                       | website
cp                                       | mv
....
```

如果您的命令格式不正確，或者您使用的是該功能發佈之前的早期版本，則可能會出現各種錯誤。若要解決圍繞這兩個問題的錯誤的詳細資訊，請參閱the section called “檢查您的 AWS CLI 命令格式”和the section called “確認您執行的是最新版本的 AWS CLI”。

回到頁首

可能的原因：安裝後需要重新啟動終端機

範例錯誤文字：

```
$ aws --version
command not found: aws
```

如果在第一次安裝或更新 後找不到 aws命令 AWS CLI，您可能需要重新啟動終端機，才能辨識任何PATH更新。

回到頁首

可能原因： AWS CLI 未完全安裝

範例錯誤文字：

```
$ aws --version
command not found: aws
```

如果在第一次安裝或更新 之後找不到aws命令 AWS CLI，則可能尚未完全安裝。請遵循 the section called “安裝/更新” 中適用於您平台的步驟，嘗試重新安裝。

回到頁首

可能原因： AWS CLI 沒有許可 (Linux)

如果在 Linux AWS CLI 上第一次安裝或更新 之後找不到aws命令，則它可能沒有其安裝所在資料夾的execute許可。使用 AWS CLI 安裝PATH的 執行下列命令，以提供 chmod許可 AWS CLI：

```
$ sudo chmod -R 755 /usr/local/aws-cli/
```

回到頁首

可能原因：安裝期間作業系統的 **PATH** 未更新。

範例錯誤文字：

```
$ aws --version
command not found: aws
```

您可能需要將 aws 可執行檔新增到作業系統的 PATH 環境變數中。若要將 AWS CLI 新增至您的 PATH，請針對您的作業系統使用以下指示。

Linux and macOS

1. 在您的使用者目錄中尋找 Shell 的描述檔命令碼。如果您不確定您擁有哪個 Shell，請執行 `echo $SHELL`。

   ```
   $ ls -a ~
   .  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
   ```

   - Bash – `.bash_profile`、`.profile` 或 `.bash_login`
   - Zsh – `.zshrc`
   - Tcsh – `.tcshrc`、`.cshrc` 或 `.login`
2. 將匯出命令新增至您的描述檔指令碼。以下命令將您的本地 bin 新增至目前的 PATH 變數。

   ```
   export PATH=/usr/local/bin:$PATH
   ```
3. 將更新過的設定檔載入到您目前的工作階段。

   ```
   $ source ~/.bash_profile
   ```

Windows

1. 在 Windows 命令提示符下，使用 `where` 命令與 `/R` *path* 參數來尋找 aws 檔案位置。結果將傳回所有包含 aws 的資料夾。

   ```
   C:\> where /R c:\ aws
   ```

```
c:\Program Files\Amazon\AWSCLIV2\aws.exe
...
```

根據預設，第 2 AWS CLI 版位於：

```
c:\Program Files\Amazon\AWSCLIV2\aws.exe
```

2. 按下 Windows 鍵並輸入 **environment variables**。
3. 從建議清單中，選擇Edit environment variables for your account (編輯您帳戶的環境變數)。
4. 選擇 PATH，然後選擇 Edit (編輯)。
5. 將您在第一個步驟中找到的路徑新增至Variable value (變數值)欄位中，例如 ***C:\Program Files\Amazon\AWSCLIV2\aws.exe***。
6. 選擇 OK (確定) 兩次以套用新的設定。
7. 關閉任何正在執行的命令提示，並重新開啟命令提示字元視窗。

回到頁首

「aws --version」命令傳回的版本與您安裝的版本不同

您的終端機可能PATH針對 傳回 AWS CLI 不同於您預期的 。

可能的原因：安裝後需要重新啟動終端機

如果 aws 命令顯示錯誤的版本，您可能需要重新啟動終端機才能辨識任何 PATH 更新。所有開放終端機都需要被關閉，不僅是您正在使用的終端機而已。

回到頁首

可能的原因：安裝後需要重新啟動系統

如果 aws 命令顯示錯誤的版本，且終端機重新啟動無效，您可能需要為其重新啟動系統以辨識您的 PATH 更新。

回到頁首

可能原因：您有多個版本的 AWS CLI

如果您更新 AWS CLI 並使用與預先存在安裝不同的安裝方法，可能會導致安裝多個版本。例如，如果您在 Linux 或 macOS 上使用 `pip`，但嘗試使用 `.pkg` 安裝檔案，這可能會導致一些衝突，特別是如果您的 `PATH` 指向舊版本。

若要解決此問題，請解除安裝所有版本的 AWS CLI，並執行全新安裝。

解除安裝所有版本後，請遵循適用於您作業系統的指示，安裝所需版本的 AWS CLI 第 1 版或 AWS CLI 第 2 版。

> **Note**
>
> 如果您在安裝 AWS CLI 版本 2 並預先安裝 AWS CLI 版本 1 之後發生這種情況，請在the section called “遷移指示”。

回到頁首

解除安裝 後，「`aws --version`」命令會傳回版本 AWS CLI

這通常發生在系統上仍有 AWS CLI 安裝 的某個位置時。

可能的原因：解除安裝後需要重新啟動終端機

如果 `aws --version` 命令仍可運作，您可能需要重新啟動終端機才能辨識任何終端機更新。

回到頁首

可能原因： AWS CLI 您在系統上有多個版本的 ，或未使用與最初安裝 相同的解除安裝方法 AWS CLI

如果您 AWS CLI 使用與安裝方法不同的方法來解除安裝 ，或安裝多個版本，則 AWS CLI 可能無法正確解除安裝。例如，如果您目前的安裝使用 `pip`，您就必須使用 `pip` 將它解除安裝。若要解決此問題， AWS CLI 請使用您用來安裝的相同方法解除安裝 。

1. 遵循適用於您作業系統的指示和原始的安裝方法，解除安裝 AWS CLI 第 1 版和 AWS CLI 第 2 版。
2. 關閉所有已打開的終端機。
3. 打開您的首選終端機，輸入以下命令並確認沒有傳回任何版本。

```
$ aws --version
command not found: aws
```

如果您仍然在輸出中列出版本，則很可能使用不同的方法安裝 AWS CLI ，或者有多個版本。如果您不知道安裝了 的方法 AWS CLI，請遵循適用於您作業系統第 AWS CLI 1 版和第 AWS CLI 2 版的每個解除安裝方法的說明，直到沒有收到版本輸出為止。

> **Note**
>
> 如果您使用套件管理工具來安裝 AWS CLI (pip、apt、brew 等)，您就必須使用相同的套件管理工具來予以解除安裝。請務必遵循套件管理工具所提供有關如何解除所有版本安裝套件的指示。

回到頁首

已 AWS CLI 處理參數名稱不完整的命令

可能的原因：您使用了 AWS CLI 參數的可辨識縮寫

由於 AWS CLI 是使用 Python 建置，因此 AWS CLI 使用 Python `argparse`程式庫，包括 `allow_abbrev`引數。參數的縮寫由 辨識 AWS CLI 並處理。

以下 `create-change-set` 命令範例會將 CloudFormation 堆疊名稱變更。參數`--change-set-n`會辨識為 的縮寫`--change-set-name`，而 會 AWS CLI 處理 命令。

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set-n my-change-set
```

當縮寫可能是多個命令時，參數將不會識別為縮寫。

以下 `create-change-set` 命令範例會將 CloudFormation 堆疊名稱變更。參數 `--change-set-` 不能以縮寫被識別，因為有可能是多個參數的縮寫，例如 `--change-set-name` 和 `--change-set-type`。因此， AWS CLI 不會處理 命令。

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set- my-change-set
```

> **Warning**
>
> 請勿特意使用參數縮寫。它們不可靠，不具向後相容性。如果將任何新參數加入混淆縮寫的命令，它將會中斷您的命令。
> 此外，如果參數是單值引數，可能會造成您的命令出現非預期的行為。如果傳遞單值引數的多個執行個體，只會執行最後一個執行個體。在下列範例中，參數 `--filters` 是單值引數。已指定參數 `--filters` 和 `--filter`。`--filter` 參數是 `--filters` 的縮寫。這會導致應用 `--filters` 的兩個實例，並且只應用最後一個 `--filter` 參數。
>
> ```
> $ aws ec2 describe-vpc-peering-connections \
>     --filters Name=tag:TagName,Values=VpcPeeringConnection \
>     --filter Name=status-code,Values=active
> ```
>
> 執行命令之前，請確認您使用的是有效的參數，以防止未預期的行為。

回到頁首

存取遭拒錯誤

可能原因： AWS CLI 程式檔案沒有「執行」許可

在 Linux 或 macOS 上，確定 aws 程式具有呼叫端使用者的執行許可。一般而言，許可會設定為 755。

若要為您的使用者新增執行許可，請執行下列命令，並以電腦上的程式路徑取代 *~/.local/bin/aws* 路徑。

```
$ chmod +x ~/.local/bin/aws
```

回到頁首

可能原因：您的 IAM 身分沒有執行此操作的許可

範例錯誤文字：

```
$ aws s3 ls
An error occurred (AcessDenied) when calling the ListBuckets operation: Access
 denied.
```

當您執行 AWS CLI 命令時，系統會代表您執行 AWS 操作，並使用 登入資料將您與 IAM 帳戶或角色建立關聯。連接的政策必須授予您許可，才能呼叫與您使用 AWS CLI執行之命令相對應的 API 動作。

大部分的命令均呼叫單一個名稱與命令名稱相符的動作。但是如 `aws s3 sync` 的自訂命令會呼叫多個 API。您可以利用 `--debug` 選項來查看命令呼叫了哪一些 API。

如果您確定使用者或角色具有政策指派的適當許可，請確定您的 AWS CLI 命令使用您預期的登入資料。請參閱下一節有關登入資料，以確認 AWS CLI 使用的登入資料是您所預期的登入資料。

如需指派 IAM 許可的相關資訊，請參閱《IAM 使用者指南》中的存取管理概觀：許可和政策。

回到頁首

無效的憑證和金鑰錯誤

範例錯誤文字：

```
$ aws s3 ls
An error occurred (InvalidAccessKeyId) when calling the ListBuckets operation: The AWS
 Access Key Id
you provided does not exist in our records.
```

```
$ aws s3 ls
An error occurred (InvalidClientTokenId) when calling the ListBuckets operation: The
 security token
included in the request is invalid.
```

可能原因： AWS CLI 正在讀取不正確的登入資料，或從非預期的位置讀取

AWS CLI 可能會從您預期的不同位置讀取登入資料，或您的金鑰對資訊不正確。您可以執行 `aws configure list` 以確認使用哪些憑證。

以下範例說明如何檢查用於預設描述檔的憑證。

```
$ aws configure list
      Name                    Value             Type    Location
      ----                    -----             ----    --------
   profile                <not set>             None    None
access_key     ****************XYVA shared-credentials-file
```

```
secret_key     ****************ZAGY shared-credentials-file
    region                us-west-2      config-file    ~/.aws/config
```

以下範例說明如何檢查具名描述檔的憑證。

```
$ aws configure list --profile saanvi
      Name                    Value             Type    Location
      ----                    -----             ----    --------
   profile                    saanvi           manual    --profile
access_key         **************** shared-credentials-file
secret_key         **************** shared-credentials-file
    region                 us-west-2      config-file    ~/.aws/config
```

若要確認您的金鑰對詳細資訊，請查看 `config` 和 `credentials` 檔案。如需 `config` 和 `credentials` 檔案的詳細資訊，請參閱 the section called “組態設定”。如需深入了解憑證與身分驗證，包含憑證優先順序，請參閱 驗證與存取憑證。

回到頁首

可能原因：電腦的時鐘不同步。

如果您使用的是有效憑證，則您的時鐘可能不同步。在 Linux 或 macOS 上，執行 `date` 來檢查時間。

```
$ date
```

如果您的系統時鐘誤差在幾分鐘內，則請使用 `ntpd` 來進行同步。

```
$ sudo service ntpd stop
$ sudo ntpdate time.nist.gov
$ sudo service ntpd start
$ ntpstat
```

在 Windows 上，使用控制台的日期與時間選項來設定系統時鐘。

回到頁首

簽章不相符錯誤

範例錯誤文字：

```
$ aws s3 ls
An error occurred (SignatureDoesNotMatch) when calling the ListBuckets operation: The
 request signature we
calculated does not match the signature you provided. Check your key and signing
 method.
```

當 AWS CLI 執行命令時，它會將加密的請求傳送至 AWS 伺服器，以執行適當的 AWS 服務操作。您的登入資料（存取金鑰和私密金鑰）會參與加密，並啟用 AWS 來驗證提出請求的人員。有幾種情況可能會干擾此程序的正確操作，如下所述。

可能原因：您的時鐘與 AWS 伺服器不同步

為了協助防止重送攻擊，可以在加密/解密程序期間使用目前時間。如果用戶端和伺服器的時間差異超過允許的數量，程序可能會失敗並拒絕請求。當您在時鐘與主機時鐘不同步的虛擬機器中執行命令，也可能發生這種情況。其中一個可能的原因是虛擬機器休眠，在喚醒後需要一些時間才能與主機的時鐘同步。

在 Linux 或 macOS 上，執行 `date` 來檢查時間。

```
$ date
```

如果您的系統時鐘誤差在幾分鐘內，則請使用 `ntpd` 來進行同步。

```
$ sudo service ntpd stop
$ sudo ntpdate time.nist.gov
$ sudo service ntpd start
$ ntpstat
```

在 Windows 上，使用控制台的日期與時間選項來設定系統時鐘。

回到頁首

可能原因：您的作業系統正在錯誤處理包含特定特殊字元的 AWS 金鑰

如果您的 AWS 索引鍵包含某些特殊字元，例如 `-`、`/`、 `+`或 `%`，則某些作業系統變體會不當處理字串，並導致索引鍵字串解譯不正確。

如果您使用其他工具或指令碼來處理金鑰，例如在新執行個體上建置登入資料檔案的工具，這些工具和指令碼可能會自行處理特殊字元，導致它們轉換為 AWS 無法辨識的項目。

我們建議重新產生私密金鑰，以取得不包含導致問題的特殊字元的私密金鑰。

回到頁首

SSL 憑證錯誤

可能原因： AWS CLI 不信任代理的憑證

範例錯誤文字：

```
$ aws s3 ls
[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed
```

當您使用 AWS CLI 命令時，會收到`[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed`錯誤訊息。這是因為 AWS CLI 不信任代理的憑證，因為代理的憑證是自我簽署的因素，而您的公司設定為憑證授權機構 (CA)。這可防止 AWS CLI 在本機 CA 登錄中找到您公司的 CA 根憑證。

若要修正此問題，請指示 使用`ca_bundle`組態`.pem`檔案設定、`--ca-bundle`命令列選項或`AWS_CA_BUNDLE`環境變數來尋找公司檔案 AWS CLI 的位置。

回到頁首

可能的原因：您的配置未指向正確的 CA 根憑證位置

範例錯誤文字：

```
$ aws s3 ls
SSL validation failed for regionname [Errno 2] No such file or directory
```

這是由於您的憑證授權機構 (CA) 套件檔案位置在 AWS CLI中被錯誤設定。若要解決這個問題，請確認您公司 `.pem` 檔案的位置，然後使用 `ca_bundle` 組態檔案設定、`--ca-bundle` 命令列選項或 `AWS_CA_BUNDLE` 環境變數，更新 AWS CLI 組態。

回到頁首

可能原因：您的組態未使用正確的 AWS 區域

範例錯誤文字：

```
$ aws s3 ls
[SSL: CERTIFICATE_ VERIFY_FAILED] certificate verify failed
```

如果您指定的 AWS 服務 無法使用，或您的資源位於不同的 中，則可能會發生錯誤 AWS 區域 或非預期的結果 AWS 區域。如需疑難排解步驟，請參閱the section called “檢查您的 AWS 區域AWS CLI 命令是否正在使用”。

回到頁首

可能原因：您的 TLS 版本需要更新

範例錯誤文字：

```
$ aws s3 ls
[SSL: UNSAFE_LEGACY_RENEGOTIATION_DISABLED] unsafe legacy renegotiation disabled
```

AWS 服務 使用的 TLS 版本與您裝置的 TLS 版本不相容。若要解決此問題，請更新至支援的 TLS 版本。如需詳細資訊，請參閱the section called “強制執行最低 TLS 版本”。

回到頁首

JSON 無效錯誤

範例錯誤文字：

```
$ aws dynamodb update-table \
    --provisioned-throughput '{"ReadCapacityUnits":15,WriteCapacityUnits":10}' \
    --table-name MyDDBTable
Error parsing parameter '--provisioned-throughput': Invalid JSON: Expecting property
 name enclosed in
double quotes: line 1 column 25 (char 24)
JSON received: {"ReadCapacityUnits":15,WriteCapacityUnits":10}
```

當您使用 AWS CLI 命令時，會收到「Invalid JSON」錯誤訊息。當您輸入具有預期 JSON 格式的命令，且 AWS CLI 無法正確讀取 JSON 時，通常會看到此錯誤。

可能原因：您未輸入有效的 JSON AWS CLI 供 使用

請確定您已為命令輸入有效的 JSON。我們建議您為有格式化問題的 JSON 使用 JSON 驗證器。

若要在命令列中使用更進階的 JSON，請考慮使用 jq 之類的命令列 JSON 處理器來建立 JSON 字串。如需 jq 的詳細資訊，請參閱 GitHub 上的 jq 儲存庫。

回到頁首

可能原因：終端機的引號規則阻止有效的 JSON 傳送至 AWS CLI

從命令 AWS CLI 接收任何內容之前，終端機會使用自己的引號和逸出規則來處理命令。由於終端機的格式化規則，您的部分 JSON 內容可能在命令傳遞至 AWS CLI之前遭除去。格式化命令時，務必使用終端機的引號規則。

若要進行疑難排解，請使用 echo 命令查看 shell 如何處理您的參數：

```
$ echo {"ReadCapacityUnits":15,"WriteCapacityUnits":10}
ReadCapacityUnits:15 WriteCapacityUnits:10
```

```
$ echo '{"ReadCapacityUnits":15,"WriteCapacityUnits":10}'
{"ReadCapacityUnits":15,"WriteCapacityUnits":10}
```

修改您的命令，直到傳回有效的 JSON 為止。

如需深入疑難排解的詳細資訊，請使用 --debug 參數查看偵錯日誌，日誌將準確顯示傳遞至 AWS CLI的內容：

```
$ aws dynamodb update-table \
    --provisioned-throughput '{"ReadCapacityUnits":15,WriteCapacityUnits":10}' \
    --table-name MyDDBTable \
    --debug
2022-07-19 22:25:07,741 - MainThread - awscli.clidriver - DEBUG - CLI version: aws-
cli/1.18.147
Python/2.7.18 Linux/5.4.196-119.356.amzn2int.x86_64 botocore/1.18.6
2022-07-19 22:25:07,741 - MainThread - awscli.clidriver - DEBUG - Arguments entered
 to CLI:
['dynamodb', 'update-table', '--provisioned-throughput',
 '{"ReadCapacityUnits":15,WriteCapacityUnits":10}',
 '--table-name', 'MyDDBTable', '--debug']
```

請用終端機的引號規則修正 JSON 輸入在傳送至 AWS CLI時可能發生的任何問題。如需引號規則的詳細資訊，請參閱the section called “具有字串的引號”。

> **Note**
>
> 如果您在取得有效的 JSON 到 時遇到問題 AWS CLI，建議您使用 Blob 直接將 JSON 資料傳遞至 ，以略過終端機的 JSON 資料輸入引號規則 AWS CLI。如需 Blobs 的詳細資訊，請參閱 Blob。

回到頁首

其他資源

如需 AWS CLI 問題的其他協助，請造訪 GitHub 上的 AWS CLI 社群或 AWS re:Post 社群。

回到頁首

第 2 AWS CLI 版的遷移指南

本節包含將第 1 AWS CLI 版更新為第 2 AWS CLI 版的說明。 AWS CLI 第 2 版以第 1 AWS CLI 版為基礎，並包含以社群意見回饋為基礎的功能和增強功能。 AWS CLI 第 2 版是 的最新主要版本， AWS CLI 並支援所有最新的功能。第 2 版中引進的某些功能不會回溯到第 1 版，您必須升級才能存取這些功能。

為了避免意外問題，在您遷移至版本 2 之前，請先了解版本之間的差異。第 2 AWS CLI 版包含可能需要您更新指令碼或命令才能回溯相容性的新功能和變更。

AWS CLI 第 1 版和第 2 版使用相同的aws命令名稱。如果您已安裝兩個版本，您的電腦會使用搜尋路徑中找到的第一個版本。這可能會導致您的aws命令名稱呼叫舊 AWS CLI 版本，即使您已安裝新的版本。

若要更新到第 2 AWS CLI 版，請遵循下列其中一個指示：

- 如果您先前已安裝 第 1 AWS CLI 版，請遵循 中的指示the section called “遷移指示”。
- 如果您先前尚未安裝第 1 AWS CLI 版，請遵循 中的指示開始使用。

主題

- 第 2 AWS CLI 版中的新功能和變更
- 從第 1 AWS CLI 版安裝第 2 AWS CLI 版

第 2 AWS CLI 版中的新功能和變更

本主題說明第 1 AWS CLI 版和第 2 AWS CLI 版之間的新功能和行為變更。這些變更可能需要您更新指令碼或命令，才能在第 2 版中獲得與在第 1 版中執行的相同行為。

主題

- AWS CLI 第 2 版新功能
- 中斷第 1 AWS CLI 版和第 2 AWS CLI 版之間的變更

AWS CLI 第 2 版新功能

AWS CLI 第 2 版是 的最新主要版本， AWS CLI 並支援所有最新的功能。在 第 2 版中推出的某些功能無法回溯至第 1 版，因此您必須升級才能存取這些功能。重要功能如下所示：

不需要 Python 解譯器

第 2 AWS CLI 版不需要另外安裝 Python。它包括一個嵌入式版本。

精靈

您可以搭配 第 2 AWS CLI 版使用精靈。該精靈會逐步引導您建構某些命令。

IAM Identity Center 驗證

如果您的組織使用 AWS IAM Identity Center (IAM Identity Center)，您的使用者可以登入 Active Directory、內建 IAM Identity Center 目錄，或連線至 IAM Identity Center 的其他 IdP。然後，它們會映射到可讓您執行 AWS CLI 命令的 AWS Identity and Access Management (IAM) 角色。

自動提示

啟用時，第 2 AWS CLI 版會在您執行 命令時提示您輸入aws命令、參數和資源。

執行 的官方 Amazon ECR Public 或 Docker 映像 AWS CLI

的官方 Docker 映像 AWS CLI 提供 AWS 直接支援和維護的隔離性、可攜性和安全性。如此一來，您就可以在容器型環境中使用 第 2 AWS CLI 版，而無需自行管理安裝。

用戶端分頁程式

第 2 AWS CLI 版提供用戶端分頁程式用於輸出。根據預設，此功能會開啟並透過作業系統的預設分頁程式傳回所有輸出。

`aws configure import`

匯入從 AWS Management Console產生的 `.csv` 憑證。系統會匯入 `.csv` 檔案，其設定檔名稱與 IAM 使用者名稱相符。

`aws configure list-profiles`

列出已配置的所有設定檔的名稱。

the section called “YAML 串流輸出格式”

`yaml` 和 `yaml-stream` 格式會利用 YAML 格式，同時透過將資料串流傳輸給您，提供更快的大型資料集檢視回應速度。您可以先開始檢視和使用 YAML 資料，再下載整體查詢。

新增 DynamoDB 的高階 **ddb** 命令

第 2 AWS CLI 版具有高階 Amazon DynamoDB 命令 `ddb put`和 `ddb select`。這些命令提供簡化的界面，用於將項目放入 DynamoDB 資料表，以及在 DynamoDB 資料表或索引中進行搜尋。

`aws logs tail`

第 2 AWS CLI 版具有自訂`aws logs tail`命令，可追蹤 Amazon CloudWatch Logs 群組的日誌。預設情況下，該命令會傳回過去十分鐘內來自所有關聯的 CloudWatch Logs 串流的日誌。

為高階 `s3` 命令新增中繼資料支援

第 2 AWS CLI 版會將 `--copy-props` 參數新增至高階`s3`命令。使用此參數，您可以為 Amazon Simple Storage Service (Amazon S3) 設定更多的中繼資料和標籤。

`AWS_REGION`

第 2 AWS CLI 版有一個 SDK AWS 相容的環境變數，稱為 `AWS_REGION`。此變數會指定 AWS 區域 要傳送請求的 。它會覆寫 `AWS_DEFAULT_REGION` 環境變數，該變數僅適用於 AWS CLI。

中斷第 1 AWS CLI 版和第 2 AWS CLI 版之間的變更

本節說明第 1 AWS CLI 版和第 2 AWS CLI 版之間行為的所有變更。這些變更可能需要您更新指令碼或命令，才能在第 2 版中獲得與在第 1 版中執行的相同行為。

主題

- 已新增環境變數以設定文字檔案編碼
- 預設情況下，會以 base64 編碼字串的格式傳遞二進位參數
- 改善 Amazon S3 針對分段副本的檔案屬性和標籤的處理
- 不會為參數自動擷取 http:// 或 https:// URL
- 預設情況下，對於所有輸出使用的分頁程式
- 時間戳記輸出值會標準化為 ISO 8601 格式
- 改善未造成任何變更的 CloudFormation 部署處理
- 已變更 us-east-1 區域的區域性 Amazon S3 端點的預設行為
- 變更區域 AWS STS 端點的預設行為
- 已移除 ecr get-login 並替換為 ecr get-login-password
- AWS CLI 外掛程式的第 2 版支援正在變更
- 已移除隱藏別名支援
- 不支援 api_versions 組態檔案設定
- AWS CLI 第 2 版僅使用 Signature v4 來驗證 Amazon S3 請求
- AWS CLI 第 2 版與分頁參數更一致

- AWS CLI 第 2 版在所有命令中提供更一致的傳回碼

已新增環境變數以設定文字檔案編碼

預設情況下，文字檔案使用與安裝的地區設定相同的編碼。the section called “Blob”由於 AWS CLI 版本 2 使用內嵌版本的 Python，因此不支援 PYTHONUTF8和 PYTHONIOENCODING環境變數。若要將文字檔案的編碼設定為與地區設定不同，請使用 AWS_CLI_FILE_ENCODING 環境變數。下列範例會將設定為 AWS CLI 在 Windows UTF-8 上使用 開啟文字檔案。

```
AWS_CLI_FILE_ENCODING=UTF-8
```

如需詳細資訊，請參閱 設定 的環境變數 AWS CLI。

預設情況下，會以 base64 編碼字串的格式傳遞二進位參數

在 中 AWS CLI，某些命令需要 base64 編碼的字串，而其他命令則需要 UTF-8-encoded的位元組字串。在 AWS CLI 版本 1 中，在兩種編碼字串類型之間傳遞資料通常需要一些中繼處理。第 2 AWS CLI 版可讓處理二進位參數更加一致，這有助於更可靠地將值從一個命令傳遞到另一個命令。

根據預設， AWS CLI 版本 2 會將所有二進位輸入和二進位輸出參數傳遞為 base64 編碼字串 blobs（二進位大型物件）。如需詳細資訊，請參閱the section called “Blob”。

若要還原到第 1 AWS CLI 版行為，請使用 cli_binary_format 檔案組態或 --cli-binary-format 參數。

改善 Amazon S3 針對分段副本的檔案屬性和標籤的處理

當您使用 aws s3 命名空間中的 AWS CLI 版本 1 命令，將檔案從一個 S3 儲存貯體位置複製到另一個儲存貯體，且該操作使用分段複製時，不會將來源物件中的檔案屬性複製到目的地物件。

根據預設， AWS CLI 版本 2 中的對應命令會將所有標籤和部分屬性從來源傳輸到目的地複本。相較於第 1 AWS CLI 版，這可能會導致對 Amazon S3 端點進行更多的 AWS API 呼叫。若要變更 AWS CLI 版本 2 中s3命令的預設行為，請使用 --copy-props 參數。

如需詳細資訊，請參閱the section called “分段副本中的檔案屬性和標籤”。

不會為參數自動擷取 **http://** 或 **https://** URL

當參數值以 http://或 開頭時，第 2 AWS CLI 版不會執行GET操作https://，也不會使用傳回的內容做為參數值。因此，相關聯的命令列選項cli_follow_urlparam會從第 2 AWS CLI 版中移除。

如果您需要擷取 URL 並將該 URL 內容傳遞至參數值，我們建議您使用 `curl` 或類似的工具，將 URL 的內容下載至本機檔案。然後使用 `file://` 語法讀取該檔案的內容，並將其作為參數值使用。

例如，下列命令不再會嘗試擷取在 `http://www.example.com` 找到的頁面內容，並傳遞這些內容作為參數。而是會傳遞文字字串 `https://example.com` 作為參數。

```
$ aws ssm put-parameter \
    --value http://www.example.com \
    --name prod.microservice1.db.secret \
    --type String 2
```

如果您需要擷取並使用 Web URL 的內容作為參數，則可以在第 2 版中執行以下操作。

```
$ curl https://my.example.com/mypolicyfile.json -o mypolicyfile.json
$ aws iam put-role-policy \
    --policy-document file://./mypolicyfile.json \
    --role-name MyRole \
    --policy-name MyReadOnlyPolicy
```

在先前的範例中，`-o` 參數會告訴 `curl` 將檔案以與來源檔案相同的名稱儲存在目前的資料夾中。第二個命令會擷取該下載檔案的內容，並將內容傳遞作為 `--policy-document` 的數值。

預設情況下，對於所有輸出使用的分頁程式

根據預設，第 2 AWS CLI 版會透過作業系統的預設分頁程式傳回所有輸出。此程式是在 Linux 和 macOS 上的 `less` 程式，以及在 Windows 上的 `more` 程式。這可以透過一次顯示一個輸出頁面，讓您比較容易瀏覽來自服務的大量輸出。

您可以設定第 2 AWS CLI 版使用不同的分頁程式，或完全不使用。如需詳細資訊，請參閱the section called “用戶端分頁程式”。

時間戳記輸出值會標準化為 ISO 8601 格式

根據預設，第 2 AWS CLI 版會以 ISO 8601 格式傳回所有時間戳記回應值。在第 1 AWS CLI 版中，命令會以 HTTP API 回應傳回的任何格式傳回時間戳記值，這可能因服務而異。

若要以 HTTP API 回應傳回的格式查看時間戳記，請使用您的 `config` 檔案中的 `wire` 值。如需詳細資訊，請參閱`cli_timestamp_format`。

改善未造成任何變更的 CloudFormation 部署處理

根據預設，在 AWS CLI 版本 1 中，如果您部署的 AWS CloudFormation 範本不會造成任何變更，則會 AWS CLI 傳回失敗的錯誤碼。如果您不認為這是錯誤，並希望指令碼繼續進行，將會發生問題。您可以在 AWS CLI 版本 1 中新增 旗標 來解決此問題`--no-fail-on-empty-changeset`，該旗標會傳回 `0`。

由於這是常見的使用案例，因此當部署未造成任何變更，且操作傳回空的變更集`0`時，第 2 AWS CLI 版預設會傳回 的成功結束碼。

若要還原為原始的行為，請新增旗標 `--fail-on-empty-changeset`。

已變更 **`us-east-1`** 區域的區域性 Amazon S3 端點的預設行為

當您將AWS CLI 版本 1 設定為使用 `us-east-1`區域時， AWS CLI 會使用實際託管在 `us-east-1`區域中的 全域`s3.amazonaws.com`端點。指定區域`s3.us-east-1.amazonaws.com`時，第 2 AWS CLI 版會使用真正的區域端點。若要強制 AWS CLI 版本 2 使用全域端點，您可以將命令的區域設定為 `aws-global`。

變更區域 AWS STS 端點的預設行為

根據預設，第 2 AWS CLI 版會將 all AWS Security Token Service (AWS STS) API 請求傳送至目前設定的 區域端點 AWS 區域。

根據預設，第 1 AWS CLI 版會將 AWS STS 請求傳送至 全域 AWS STS 端點。您可以藉由使用 `sts_regional_endpoints` 設定，在第 1 版中控制此預設行為。

已移除 **`ecr get-login`** 並替換為 **`ecr get-login-password`**

第 2 AWS CLI 版會將 命令取代`aws ecr get-login`為 `aws ecr get-login-password`命令，以改善與容器身分驗證的自動化整合。

`aws ecr get-login-password` 命令可降低在程式清單、Shell 歷史記錄或其他日誌檔中公開憑證的風險。它也改善了與 `docker login` 命令的相容性，以獲得更好的自動化。

此`aws ecr get-login-password`命令可在 1.17.10 AWS CLI 版和更新版本，以及 第 2 AWS CLI 版中使用。舊版 `aws ecr get-login` 命令仍可在 AWS CLI 版本 1 中使用，以便回溯相容。

您可以使用 `aws ecr get-login-password` 命令取代用於擷取密碼的下列程式碼。

```
$ (aws ecr get-login --no-include-email)
```

若要降低將密碼公開給 Shell 歷史記錄或日誌檔的風險，請改用下列範例命令。在此範例中，密碼會直接傳送到 `docker login` 命令，此命令會由 `--password-stdin` 選項指派給密碼參數。

```
$ aws ecr get-login-password | docker login --username AWS --password-stdin MY-REGISTRY-URL
```

如需詳細資訊，請參閱 AWS CLI 第 2 版參考指南中的 `aws ecr get-login-password`。

AWS CLI 外掛程式的第 2 版支援正在變更

第 2 AWS CLI 版中的外掛程式支援完全是暫時性的，旨在協助使用者從第 1 AWS CLI 版遷移，直到發行穩定且更新的外掛程式界面為止。無法保證未來 AWS CLI 第 2 版將支援特定外掛程式或甚至 AWS CLI 外掛程式界面。如果您依賴外掛程式，請務必鎖定特定版本的 ， AWS CLI 並在升級時測試外掛程式的功能。

若要啟用外掛程式支援，請在 `~/.aws/config` 中建立 `[plugins]` 區段。

```
[plugins]
cli_legacy_plugin_path = <path-to-plugins>/python3.7/site-packages
<plugin-name> = <plugin-module>
```

在 `[plugins]` 區段中，定義 `cli_legacy_plugin_path` 變數並將其值設定為您的外掛程式模組所在的 Python 站台套件路徑。然後，您可以透過提供外掛程式的名稱 (`plugin-name`) 和 Python 模組的檔案名稱 (`plugin-module`)，其中包含您的外掛程式的原始程式碼來設定外掛程式。會透過匯入 `plugin-module`並呼叫其 `awscli_initialize`函數來 AWS CLI 載入每個外掛程式。

已移除隱藏別名支援

AWS CLI 第 2 版不再支援第 1 版中支援的下列隱藏別名。

在下表中，第一欄顯示可在所有版本中運作的服務、命令和參數，包括第 2 AWS CLI 版。第二欄顯示不再適用於第 2 AWS CLI 版的別名。

運作的服務、命令和參數	過時的別名
cognito-identity create-identity-pool open-id-connect-provider-arns	open-id-connect-provider-ar-ns
storagegateway describe-tapes tape-arns	tape-ar-ns

運作的服務、命令和參數	過時的別名
storagegateway.describe-tape-archives.tape-arns	tape-ar-ns
storagegateway.describe-vtl-devices.vtl-device-arns	vtl-device-ar-ns
storagegateway.describe-cached-iscsi-volumes.volume-arns	volume-ar-ns
storagegateway.describe-stored-iscsi-volumes.volume-arns	volume-ar-ns
route53domains.view-billing.start-time	start
deploy.create-deployment-group.ec2-tag-set	ec-2-tag-set
deploy.list-application-revisions.s3-bucket	s-3-bucket
deploy.list-application-revisions.s3-key-prefix	s-3-key-prefix
deploy.update-deployment-group.ec2-tag-set	ec-2-tag-set
iam.enable-mfa-device.authentication-code1	authentication-code-1
iam.enable-mfa-device.authentication-code2	authentication-code-2
iam.resync-mfa-device.authentication-code1	authentication-code-1
iam.resync-mfa-device.authentication-code2	authentication-code-2
importexport.get-shipping-label.street1	street-1
importexport.get-shipping-label.street2	street-2
importexport.get-shipping-label.street3	street-3
lambda.publish-version.code-sha256	code-sha-256
lightsail.import-key-pair.public-key-base64	public-key-base-64
opsworks.register-volume.ec2-volume-id	ec-2-volume-id

不支援 **api_versions** 組態檔案設定

AWS CLI 第 2 版不支援使用`api_versions`組態檔案設定呼叫舊版 AWS 的服務 APIs。所有 AWS CLI 命令現在都會呼叫端點目前支援之服務 APIs 的最新版本。

AWS CLI 第 2 版僅使用 Signature v4 來驗證 Amazon S3 請求

第 2 AWS CLI 版不支援舊版簽章演算法，以密碼編譯方式驗證傳送至 Amazon S3 端點的服務請求。此簽署會在每個 Amazon S3 請求中自動執行，並且僅支援 Signature Version 4 簽署程序。您無法設定簽章版本。所有 Amazon S3 儲存貯體預先簽章的 URL 現在僅使用 Sigv4，最長有效期為一週。

AWS CLI 第 2 版與分頁參數更一致

在 AWS CLI 版本 1 中，如果您在命令列指定分頁參數，則自動分頁會如預期關閉。但是，當您使用具有 `--cli-input-json` 參數的檔案指定分頁參數，而自動分頁未關閉時，可能會導致非預期的輸出。無論您如何提供參數，第 2 AWS CLI 版都會關閉自動分頁。

AWS CLI 第 2 版在所有命令中提供更一致的傳回碼

AWS CLI 與第 1 版相比，第 2 AWS CLI 版在所有命令中更為一致，並正確傳回適當的結束程式碼。我們也新增了結束代碼 252、253 和 254。如需結束代碼的詳細資訊，請參閱 the section called “傳回碼”。

如果您依賴 AWS CLI 版本 1 如何使用傳回碼值，建議您檢查結束碼，以確保您取得預期的值。

從第 1 AWS CLI 版安裝第 2 AWS CLI 版

本主題提供從第 1 AWS CLI 版遷移到第 2 AWS CLI 版的說明。

AWS CLI 第 1 版和第 2 版使用相同的`aws`命令名稱。如果您已安裝兩個版本，您的電腦會使用搜尋路徑中找到的第一個版本。如果您之前已安裝 第 1 AWS CLI 版，建議您執行下列其中一項操作，以使用第 2 AWS CLI 版：

- 建議 – 解除安裝第 1 AWS CLI 版，並僅使用第 2 AWS CLI 版。
- 若要同時安裝兩個版本，請使用作業系統的功能，為兩個 `aws` 命令之一建立具有不同名稱的符號連結 (symlink) 或別名。

如需第 1 版和第 2 版之間重大變更的詳細資訊，請參閱 the section called “新功能和變更”。

將版本 1 替換為版本 2

執行下列步驟，將第 1 AWS CLI 版取代為第 2 AWS CLI 版。

將 AWS CLI 第 1 版取代為第 2 AWS CLI 版

1. 透過確認 the section called “新功能和變更” 第 1 版和第 2 版之間的重大變更，為您現有的所有指令碼做好遷移的準備。
2. 依照安裝、更新和解除安裝第 1 AWS CLI 版中作業系統的解除安裝說明，解除安裝第 1 版。AWS CLI
3. 使用以下命令確認 AWS CLI 已完全解除安裝。

   ```
   $ aws --version
   ```

   根據輸出完成下列其中一項操作：

   - 未傳回版本：您已成功解除安裝 AWS CLI 版本 1，可以繼續下一個步驟。
   - 傳回版本：您仍然安裝 第 1 AWS CLI 版。如需疑難排解步驟，請參閱the section called “解除安裝 後，「aws --version」命令會傳回版本 AWS CLI”。執行故障診斷步驟，直到未接收到版本輸出。
4. 遵循 中適用於您作業系統的適當安裝說明，安裝第 2 AWS CLI 版安裝或更新至最新版本的 AWS CLI。

並排安裝

若要同時安裝兩個版本，請使用作業系統的功能，為兩個 aws 命令之一建立具有不同名稱的符號連結 (symlink) 或別名。

1. 遵循 中適用於您作業系統的適當安裝說明，安裝第 2 AWS CLI 版安裝或更新至最新版本的 AWS CLI。
2. 使用您作業系統的能力，為兩個aws命令之一建立不同名稱的符號連結或別名，例如*aws2*針對第 2 AWS CLI 版使用 。以下是第 2 AWS CLI 版的符號連結範例。以您的安裝位置替換 *PATH*。

   Linux and macOS

   您可以在 Linux 和 macOS 上si y使用符號連結或別名。

```
$ alias aws2='PATH'
```

Windows command prompt

在 Windows 上使用 DOSKEY。

```
C:\> doskey aws2=PATH
```

解除安裝第 2 AWS CLI 版

本主題說明如何解除安裝第 2 AWS Command Line Interface 版（第 2 AWS CLI 版）。

AWS CLI 第 2 版解除安裝說明：

Linux

若要解除安裝，請遵循您用來安裝 的相同方法 AWS CLI。

Command line installer

若要解除安裝第 2 AWS CLI 版，請執行下列命令。

1. 找到符號連結並安裝路徑。

   - 使用 `which` 命令尋找符號連結。這會顯示您與 `--bin-dir` 參數搭配使用的路徑。

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

   - 使用 `ls` 命令尋找符號連結指向的目錄。這為您提供了與 `--install-dir` 參數一起使用的路徑。

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

2. 現在，刪除 `--bin-dir` 目錄中的兩個符號連結。如果您的使用者具有這些目錄的寫入許可，則不需要使用 `sudo`。

   ```
   $ sudo rm /usr/local/bin/aws
   $ sudo rm /usr/local/bin/aws_completer
   ```

3. 刪除 `--install-dir` 目錄。如果您的使用者具有此資料夾的寫入許可，則不需要使用 `sudo`。

   ```
   $ sudo rm -rf /usr/local/aws-cli
   ```

4. （選用）移除 `.aws` 資料夾中的共用 AWS SDK 和 AWS CLI 設定資訊。

> **Warning**
>
> 這些組態和登入資料設定會跨 AWS SDKs和 共用 AWS CLI。如果您移除此資料夾，則系統上仍存在的任何 AWS SDKs無法存取這些資料夾。

.aws 資料夾的預設位置在平台之間各不相同，預設情況下，資料夾位於 *~/.aws/*。如果您的使用者具有此資料夾的寫入許可，則不需要使用 sudo。

```
$ sudo rm -rf ~/.aws/
```

Snap

我們在 AWS CLI 上提供官方 AWS 支援的 版本snap。如果您曾經snap安裝 AWS CLI，請使用下列步驟：

1. 針對 執行下列snap remove命令 AWS CLI。

   ```
   $ snap remove aws-cli --classic
   ```

   如果您曾經sudo安裝 AWS CLI，則需要將其新增至 命令。

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

2. 您可能需要重新啟動命令提示字元視窗或電腦，才能移除所有檔案。
3. （選用）移除 .aws 資料夾中的共用 AWS SDK 和 AWS CLI 設定資訊。

   > **Warning**
   >
   > 這些組態和登入資料設定會跨 AWS SDKs和 共用 AWS CLI。如果您移除此資料夾，則系統上仍存在的任何 AWS SDKs無法存取這些資料夾。

   .aws 資料夾的預設位置在平台之間各不相同，預設情況下，資料夾位於 *~/.aws/*。如果您有寫入此目錄的許可，則不需要使用 sudo。

```
$ sudo rm -r ~/.aws/
```

macOS

若要解除安裝第 2 AWS CLI 版，請執行下列命令，以取代您用來安裝的路徑。範例命令使用預設安裝路徑。

1. 尋找包含主程式和完成標籤之符號連結的資料夾。

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

2. 使用該資訊，執行下列命令，以尋找符號連結指向的安裝資料夾。

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

3. 現在，刪除第一個資料夾中的兩個符號連結。如果您的使用者具有這些資料夾的寫入權限，則不需要使用 sudo。

   ```
   $ sudo rm /usr/local/bin/aws
   $ sudo rm /usr/local/bin/aws_completer
   ```

4. 刪除主要安裝資料夾。使用 sudo 來取得 /usr/local 資料夾的寫入權限。

   ```
   $ sudo rm -rf /usr/local/aws-cli
   ```

5. （選用）移除 .aws 資料夾中的共用 AWS SDK 和 AWS CLI 設定資訊。

   > **Warning**
   >
   > 這些組態和憑據設定在所有 AWS 開發套件和 AWS CLI之間共用。如果您移除此資料夾，則系統上仍存在的任何 AWS SDKs無法存取這些資料夾。

   .aws 資料夾的預設位置在平台之間各不相同，預設情況下，資料夾位於 *~/.aws/*。如果您的使用者具有此資料夾的寫入許可，則不需要使用 sudo。

```
$ sudo rm -rf ~/.aws/
```

Windows

1. 請執行下列其中一個動作來開啟程式和功能：

   - 開啟控制台，然後選擇程式和功能。
   - 開啓命令提示並輸入下列命令。

   ```
   C:\> appwiz.cpl
   ```

2. 選擇名為 AWS Command Line Interface 的項目，然後選擇解除安裝以啟動解除安裝程式。
3. 確認您要解除安裝 AWS CLI。
4. （選用）移除 `.aws` 資料夾中的共用 AWS SDK 和 AWS CLI 設定資訊。

   > **Warning**
   >
   > 這些組態和憑據設定在所有 AWS 開發套件和 AWS CLI之間共用。如果您移除此資料夾，則系統上仍存在的任何 AWS SDKs無法存取這些資料夾。

   `.aws` 資料夾的預設位置在平台之間各不相同，預設情況下，資料夾位於 *`%UserProfile%\.aws`*。

   ```
   $ rmdir %UserProfile%\.aws
   ```

針對 AWS CLI 安裝和解除安裝錯誤進行故障診斷

如果您在安裝或解除安裝 之後遇到問題 AWS CLI，請參閱 故障診斷錯誤 以取得故障診斷步驟。如需最相關的故障診斷步驟，請參閱 the section called “命令未找到錯誤”、the section called “「aws --version」命令傳回的版本與您安裝的版本不同” 和 the section called “解除安裝 後，「aws --version」命令會傳回版本 AWS CLI”。

AWS CLI 使用者指南文件歷史記錄

下表說明從 2019 年 1 月開始的《AWS Command Line Interface 使用者指南》重要增補項目。如需有關此文件更新的通知，您可以訂閱 RSS 摘要。

變更	描述	日期
已將帳戶型端點支援新增至第 2 AWS CLI 版	第 2 AWS CLI 版現在支援使用您 AWS 帳戶 ID 的帳戶型端點。	2025 年 3 月 20 日
帳戶型端點支援已新增至第 1 AWS CLI 版	第 2 AWS CLI 版現在支援使用您 AWS 帳戶 ID 的帳戶型端點。	2025 年 2 月 24 日
已更新憑證與驗證資訊。	已更新憑證和驗證方法指示與範例。其中包含更新的相關入門頁面和配置頁面。為了因應日漸增加的文件數量，相關憑證主題已移至新的驗證和存取憑證區段。	2023 年 3 月 31 日
新增 AWS IAM Identity Center 的自動身分驗證重新整理的字符提供者組態	設定 AWS CLI 以使用 SSO 權杖提供者組態向 AWS IAM Identity Center (IAM Identity Center) 驗證使用者的新程序，可自動擷取重新整理的身分驗證權杖。	2022 年 12 月 7 日
發行第 2 AWS CLI 版的官方 Amazon ECR Public 映像	Linux、macOS 和 Windows 發行 2 AWS CLI 版官方支援的 Amazon ECR Public 映像。	2022 年 11 月 18 日
更新從 AWS CLI V1 遷移至 V2 的指南	擴充重大變更指南，納入從第 1 AWS CLI 版到第 2 AWS CLI 版的遷移指示。包含故障診斷	2022 年 5 月 13 日

	頁面的更新，以協助解決安裝問題。	
從來源建置 AWS CLI 安裝程式的新程序。	從來源安裝或更新到支援作業系統最新版 AWS CLI 的新程序。	2022 年 2 月 17 日
AWS CLI V1 和 V2 的內容現在已分為各自的指南	為了清楚易懂，第 1 AWS CLI 版和第 2 AWS CLI 版的內容現在會分成自己的指南。 如需第 1 AWS CLI 版，請參閱 第 AWS CLI 1 版使用者指南。	2021 年 11 月 2 日
新增 AWS CLI 別名資訊	新增 AWS CLI 別名資訊。別名是您可以在 AWS Command Line Interface (AWS CLI) 中建立的快速鍵，能夠用來縮短您經常使用的命令或指令碼。	2021 年 3 月 11 日
已更新篩選條件輸出資訊	已更新篩選條件資訊，並將資訊移動到各篩選條件的專屬頁面。	2021 年 2 月 1 日
已新增精靈的資訊	新增第 2 AWS CLI 版精靈資訊。	2020 年 11 月 20 日
已更新自動提示	使用目前功能更新第 2 AWS CLI 版自動提示資訊。	2020 年 11 月 10 日
已新增 Amazon S3 指令碼範例	已新增 Amazon S3 生命週期指令碼範例。	2020 年 10 月 15 日
已新增 Amazon EC2 指令碼範例	已新增 Amazon EC2 執行個體類型指令碼範例。	2020 年 10 月 15 日
已新增重試資訊	已為 AWS CLI中的重試功能和行為新增重試頁面。	2020 年 9 月 17 日

伺服器端和用戶端分頁頁面	已更新分頁資訊，並集中在單一頁面上。	2020 年 8 月 17 日
已更新 s3 命令頁面	已使用新的範例和資源更新高階 s3 命令頁面。	2020 年 7 月 30 日
更新的安裝資訊	Linux、macOS 和 Windows 的安裝、更新和解除安裝資訊都會更新。	2020 年 5 月 19 日
已新增 AWS CLI 第 2 版文字檔案編碼的資訊	根據預設，第 2 AWS CLI 版使用與本機相同的文字檔案編碼。您現在可以使用環境變數來設定文字檔案編碼。	2020 年 5 月 14 日
發行第 2 AWS CLI 版的官方 Docker 映像	已針對所有 Linux、macOS 和 Windows 發行 AWS CLI 版本 2 的官方支援 Docker 映像。	2020 年 3 月 31 日
已新增有關 AWS CLI 第 2 版用戶端分頁程式的資訊	根據預設， AWS CLI 版本 2 `less`會針對所有用戶端輸出使用分頁程式。	2020 年 2 月 19 日
AWS Command Line Interface (AWS CLI) 正式發行第 2 版	AWS CLI 第 2 版已全面推出，是建議客戶安裝的版本。	2020 年 2 月 10 日
第 2 AWS CLI 版的 macOS 安裝程式現在是 Apple Package 安裝程式`.pkg`檔案。	第 2 AWS CLI 版的 macOS 安裝程式已從具有 shell 指令碼`.zip`的檔案更新為完整的 macOS Installer 套件。這簡化了安裝程序，並使其與最新的 macOS 版本相容。	2020 年 2 月 3 日
新增第 2 AWS CLI 版改善 S3 和 STS 區域端點預設處理的內容	根據預設，第 2 AWS CLI 版現在會將 Amazon S3 AWS STS 和服務的要求導向目前設定的區域端點，而非全域端點。	2020 年 1 月 13 日

第 2 AWS CLI 版的開發人員預覽版	宣布 第 2 AWS CLI 版的預覽版本。已新增有關安裝版本 2 的說明。新增移轉主題，討論版本 1 和版本 2 之間的差異。	2019 年 11 月 7 日
AWS IAM Identity Center 已將的支援新增至 AWS CLI 具名設定檔	AWS CLI 第 2 版新增了建立具名設定檔的支援，該設定檔可以直接登入 IAM Identity Center，並取得 AWS 臨時登入資料以用於後續 AWS CLI 命令。	2019 年 11 月 7 日
新的 MFA 部分	新增新的部分，描述如何使用多重要素驗證和角色來存取 CLI。	2019 年 5 月 3 日
更新為「使用 CLI」部分	使用說明和程序的重大改善和新增項目。	2019 年 3 月 7 日
更新為「安裝 CLI」部分	AWS CLI 安裝指示和程序的主要改善和新增項目。	2019 年 3 月 7 日
更新為「設定 CLI」部分	AWS CLI 組態指示和程序的主要改進和新增項目。	2019 年 3 月 7 日